

Courte information sur le système UNIX et les réseaux téléinformatiques

version 3.7 du 1^{er} mars 2003

Marc SCHAEFER
email : schaefer@alphanet.ch

Remerciements

Mes remerciements vont à Christophe CESTONARO <ccestona@di.epfl.ch> pour son guide sur l'éditeur Emacs, à Michel SCHINZ <schinz@di.epfl.ch> pour son texte sur le projet GNU, à Paul-Jean CAGNARD <cagnard@alphanet.ch> pour sa relecture attentive ainsi qu'à Norbert EBEL <ebel@lcao.epfl.ch>, Isabelle SAINT-SAËNS <iss@deneb.inrets.fr> et Christian JOBIC <jc@inrets.fr> pour leurs nombreuses corrections et suggestions. Je remercie également Christophe WOLFHUGEL et Pascal PETIT pour la diffusion du document, ainsi que tous ceux qui m'ont envoyé des suggestions et des corrections. La version 3 a été rendue possible par Olivier RICOU <olivier.ricou@devinci.fr> et Frédéric COUCHET <fcouchet@april.org>.

Résumé

Ce guide, composé de trois parties, traite essentiellement des systèmes UNIX, de l'interface graphique X11 et des réseaux de communication sous UNIX. Après une courte introduction décrivant le contexte historique, la première partie décrit quelques concepts fondamentaux des systèmes UNIX en général et ensuite étudie les services offerts par les systèmes UNIX à disposition. La deuxième partie décrit l'interface graphique standard X11. La troisième partie traite des réseaux de communication disponibles sous environnement UNIX. Ce texte, écrit spécialement pour les membres d'ALPHANET NF désireux d'utiliser les systèmes à disposition sous UNIX et d'accéder efficacement aux ressources réseaux, ne saurait remplacer un bon livre de référence UNIX ni un livre traitant en profondeur des réseaux informatiques. Ce guide a été cependant rédigé dans l'espoir qu'il serait utile : il est rendu disponible tel quel et l'auteur décline toute responsabilité quant à son contenu.

Copyright

Copyright © Marc SCHAEFER 1993-2002

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation ; with the Invariant Sections being "Remerciements", with one Front-Cover Text "Auteur original : Marc Schaefer <schaefer@alphanet.ch>" and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Table des matières

1	Introduction	9
1.1	Buts de ce résumé	9
1.2	Conventions typographiques	9
1.3	Structure	9
1.4	Versions disponibles	9
1.5	Distributions particulières	10
1.6	Adresse postale de contact	10
1.7	Comment désigner ce document ?	10
1.8	Corrections et suggestions	10
2	Historique	11
2.1	Introduction	11
2.2	Origine du système UNIX	11
2.3	Origine de Usenet et de UUCP	11
2.4	Origine d' Internet	12
2.5	La fondation FSF et le projet GNU	12
2.5.1	Qu'est-ce que la FSF ?	12
2.5.2	Que sont la GPL et la LGPL ?	12
2.5.3	Qu'est-ce que le projet GNU ?	12
2.5.4	Les principaux logiciels	12
2.5.5	Pour en savoir plus	13
2.5.6	Comment soutenir la FSF ?	13
I	UNIX	14
3	Le système UNIX	14
3.1	Introduction	14
3.2	Particularités	14
3.2.1	Caractéristiques du système d'exploitation	14
3.2.2	Problèmes courants	14
3.2.3	Arborescence	14
3.2.4	Noms de fichiers et hiérarchie	15
3.2.5	Notions d'utilisateur et de groupe	16
4	Accès à un système UNIX	16
4.1	Appel du système par modem	16
4.2	Connexion distante au système	16
4.3	Le login	17
4.4	Fin de session	17
5	L'interface de commande (shell)	17
5.1	Introduction et familles de shells	17
5.2	Choix du shell présenté	18
5.3	Notions principales	18
5.4	Historique, complétion, variables d'environnement, édition de ligne	19
5.5	Redirections	19
5.6	Expressions régulières	20
5.6.1	Introduction	20
5.6.2	Syntaxe	20
5.7	Multi-niveau (Job control)	21
5.7.1	Concept de processus et de job	22
5.7.2	Obtenir la liste des processus	22
5.7.3	Suspendre un processus	22
5.7.4	Reprendre l'exécution d'un processus	22
5.7.5	Détruire un processus	22
5.7.6	Lancer un processus en arrière-plan	23
5.8	Syntaxe	23
5.9	Principales commandes	23
5.9.1	ls, liste d'un répertoire	23
5.9.2	a2ps, mp, enscript, mise en page de listings	24

5.9.3	alias, définition de synonymes de commandes	24
5.9.4	awk, interprète de scripts	24
5.9.5	cat, affichage ininterrompu d'un fichier	24
5.9.6	cd, changement de répertoire	25
5.9.7	chmod, changement des protections d'un fichier	25
5.9.8	crontab, opérations à heures fixes	25
5.9.9	mv, déplacer/renommer des fichiers	26
5.9.10	df, disponibilité des systèmes de fichiers	26
5.9.11	diff, affichage des différences entre fichiers	26
5.9.12	du, usage disque	26
5.9.13	emacs, éditeur pleine page multi-fonctions	26
5.9.14	find, recherche de fichiers	27
5.9.15	grep, recherche de motif dans un fichier	27
5.9.16	gzip, compression et décompression de fichiers	27
5.9.17	head, affichage des premières lignes d'un fichier	27
5.9.18	kill, destruction d'un processus ou communication de signaux	27
5.9.19	lpspr, mise en page de documents PostScript	28
5.9.20	man, manuels en ligne	28
5.9.21	mkdir, création d'un répertoire	28
5.9.22	more, affichage page par page d'un fichier	28
5.9.23	nice, changement de priorité	28
5.9.24	nroff, mise en page de manuels	28
5.9.25	passwd, changement du mot de passe	28
5.9.26	pico, éditeur pleine page simplifié	28
5.9.27	pwd, affichage du répertoire courant	29
5.9.28	quota, information sur l'utilisation disque autorisée	29
5.9.29	rm, destruction d'un fichier	29
5.9.30	rz, réception d'un fichier via ZModem	29
5.9.31	screen, gestionnaire de fenêtres virtuelles	29
5.9.32	sed, éditeur de flots	29
5.9.33	stty, changement de modes du terminal	30
5.9.34	sz, envoi d'un fichier via ZModem	30
5.9.35	tail, affichage des dernières lignes d'un fichier	30
5.9.36	tar, gestion d'archives portables	30
5.9.37	uuencode, codage de fichiers binaires pour transmission via courrier électronique	30
5.9.38	vi, éditeur pleine page	30
5.9.39	wc, comptage du nombre de lignes, de mots ou de caractères d'un fichier	32
5.9.40	file, détermination du type du contenu d'un fichier	32
6	Introduction à l'éditeur Emacs	33
6.1	La fenêtre d'Emacs	33
6.2	Touches Control, Escape et Meta	34
6.3	Fonctions générales	34
6.4	Fenêtres	34
6.5	Fichiers et buffers	34
6.6	Déplacements	35
6.7	Edition	35
6.8	Copier et coller	35
6.9	Point, Marque, Région	35
6.10	Registres	36
6.11	Recherche et remplacement	36
6.12	Macros	36
6.13	Répétitions et argument universel	36
6.14	Texte	37
6.15	Commentaires	37
6.16	Abréviations	37
6.17	Répertoire	37
6.18	Mail (mode d'écriture seulement)	38
6.19	Rmail (lecture/écriture de mails)	38
6.20	Aide	38
6.21	Impression	38
6.22	Exemple de .emacs	39
6.23	Mémento	41

7	Gestion des fichiers	42
7.1	Edition de fichiers	42
7.2	Administration des fichiers	42
7.3	Conversion de formats	42
7.3.1	Formats de disquette	42
7.3.2	Conversions de textes accentués	43
7.3.3	Conversions de sons et d'images : sox et netpbm	43
7.3.4	Conversions de fichiers multimédia	43
7.4	Compression	43
7.4.1	Comparaison de logiciels de compression	43
7.5	Transferts de fichiers	44
7.6	Chiffrement de fichiers	44
7.6.1	pgp	44
7.7	Impression de fichiers	45
7.7.1	Soumission de travaux	45
7.7.2	Etat de la queue	45
7.7.3	Destructions de travaux	45
7.7.4	En cas de problèmes	45
8	Personnaliser son environnement	45
8.1	Qu'est-ce qu'un environnement ?	46
8.1.1	Variables	46
8.1.2	Commandes de configuration	46
8.1.3	Fichiers de configuration	46
8.2	Modifier son environnement	47
8.3	Création de scripts	47
8.3.1	Buts	47
8.3.2	Sélection de l'interprète	47
8.3.3	Choix de l'interprète	48
8.3.4	Exemple d'un script sous sh	48
8.3.5	Exemple de boucle en sh	49
8.3.6	Autres exemples	49
9	Aspects de sécurité informatique	50
9.1	Piratage	50
9.1.1	En quoi suis-je concerné	50
9.2	Mot de passe	50
9.2.1	Recommandations	50
9.2.2	Changement du mot de passe	51
9.3	Protection de vos données	51
9.3.1	Pannes	51
9.3.2	Limitation d'accès à vos fichiers	51
9.3.3	Chiffrement de vos données	51
9.4	Compte personnel	52
II	Système de multifenêtrage X11	53
10	Introduction	53
11	Terminologie	53
12	Protocole	54
13	Configuration	54
13.1	Fichiers de configuration	54
13.1.1	Exemple d'un fichier .xinitrc	54
13.1.2	Exemple d'un fichier .Xmodmap	54
13.1.3	Exemple d'un fichier .twmrc	55
13.2	Commandes	57

14 Sécurité	57
14.1 Introduction	57
14.2 xhost	58
14.3 xauth	58
15 Etablissement d'une session	58
15.1 De votre côté (affichage, serveur)	58
15.2 Du côté UNIX (client)	58
15.3 Automatiser	59
15.3.1 SSH	59
15.3.2 Script rsh	59
16 Applications sous X	59
17 Logiciels TCP/IP et X11 du côté serveur (affichage)	63
17.1 Introduction	63
17.2 UNIX	63
17.3 Amiga	63
17.4 Macintosh	63
17.5 MS-DOS	63
17.6 MS-WINDOWS	64
17.7 MS-WINDOWS NT	64
17.8 OS/2	64
17.9 Atari	64
18 Problèmes courants	64
18.1 Accès au serveur refusé	64
18.2 Erreur de bibliothèques partagées	64
III Réseaux informatiques	65
19 Introduction	65
20 Principaux réseaux UNIX	65
20.1 Internet	65
20.1.1 Introduction	65
20.1.2 Description technique	66
20.1.3 Sécurité	67
20.1.4 Événements médiatiques	68
20.1.5 Comment s'y connecter	69
20.1.6 Évolution — un peu de politique des réseaux	69
20.2 UUCP	69
20.2.1 Introduction	69
20.2.2 Description technique	70
20.2.3 Sécurité	70
20.3 FidoNet	70
21 Principaux services offerts	70
21.1 Services non interactifs (store-and-forward)	70
21.2 Services interactifs	71
22 Mail (courrier électronique privé)	71
22.1 Introduction	71
22.2 Savoir-vivre	71
22.3 Interface utilisateur	72
22.3.1 mail : interface ligne	72
22.3.2 elm, mutt : interface VT100	73
22.3.3 Logiciels clients pour ordinateurs personnels	73
22.4 Tri du courrier (procmail)	73
22.5 Personnes de contact	75
22.6 Aspects techniques	75
22.6.1 Transmission	75
22.6.2 Sécurité	75

22.6.3	Spécification d'adresses	75
22.6.4	Enveloppe et entête	76
22.6.5	TOP domains	77
22.6.6	Routage mail sur des réseaux hétérogènes	77
22.7	Transfert de fichiers via mail	78
22.8	Caractères accentués français	78
23	News (messages publics)	79
23.1	Introduction	79
23.2	La modération	80
23.2.1	Introduction	80
23.2.2	Le modérateur	80
23.2.3	Modérer ou ne pas modérer ?	80
23.3	Catégories	80
23.4	Sélection de quelques forums	81
23.5	Savoir-vivre	81
23.5.1	Comment rédiger	81
23.5.2	Comment réagir	82
23.6	Que faire en cas d'abus ?	83
23.7	Interface utilisateur	83
23.7.1	tin : interface VT100	83
23.7.2	knews, interface X11	84
23.7.3	Logiciels clients pour ordinateurs personnels	84
23.7.4	Les archives	84
23.8	Aspects techniques	84
23.8.1	Transfert	84
23.8.2	Sécurité	85
24	FTP	85
24.1	Introduction	85
24.2	Logiciels clients	85
24.3	Utilisation	86
24.4	Problèmes courants	86
24.5	La commande ftp	86
24.6	Quelques sites FTP	88
25	Communication à distance	88
25.1	Introduction	88
25.2	finger	88
25.3	who	89
25.4	w	89
25.5	write	89
25.6	talk	89
26	Connexion à distance	89
26.1	Introduction	89
26.2	telnet	89
26.3	rlogin	90
26.4	rsh	90
26.5	rcp	90
26.6	ssh	90
27	Diagnostic de problèmes de réseau	90
27.1	Introduction	90
27.2	ping	90
27.3	tracert	90
27.4	netstat	91
27.5	whois	91

28 WWW (World-Wide Web)	91
28.1 Introduction	91
28.2 Terminologie	92
28.3 Protocole	92
28.4 Sécurité	92
28.5 Interfaces	92
28.5.1 lynx	92
28.5.2 w3m	93
28.5.3 Mosaic	93
28.5.4 Chimera	94
28.5.5 Netscape	94
28.5.6 Konqueror	95
28.6 Comment créer votre propre page WWW	95
28.7 Exemple d'un fichier source HTML	96
28.8 Quelques serveurs WWW	96
28.9 Performances	97
29 IRC : Internet Relay Chat	97
29.1 Introduction	97
29.2 Logiciels clients	98
29.2.1 irc	98
30 Information sur le protocole SLIP	98
30.1 Introduction	98
30.2 Information sur le protocole	98
30.3 Configuration d'un réseau SLIP	99
30.4 Problèmes courants	99
30.4.1 CSLIP	99
30.4.2 PPP	99
30.4.3 Routage mal configuré	99
30.5 Déroulement d'une session SLIP	99
30.5.1 Diagnostic	99
31 Information sur le système DNS (Domain Name Server)	100
31.1 Introduction	100
31.2 Commande associée d'interrogation : nslookup	100
32 Problèmes de télécommunication	101
32.1 Introduction	101
32.2 Les modems	101
32.2.1 Jargon technique	101
32.2.2 Vitesse et modulation	102
32.2.3 Les standards ITU-T et propriétaires	102
32.2.4 Aspects techniques	103
32.2.5 Les options avancées	104
32.2.6 Mise en œuvre d'un modem	104
32.3 Les logiciels de communication	105
32.3.1 Terminologie	105
32.3.2 Les émulations de terminal	105
32.3.3 Protocoles de transfert de fichiers	105
32.4 RNIS	106
32.4.1 Introduction	106
32.4.2 Aspects techniques	106
Glossaire	107
Plus d'informations	117
Bibliographie	117
GNU Free Documentation License	119
Index	120

Table des figures

1	Arborescence du système de fichiers	15
2	Généalogie des shells	18
3	Détail d'une sortie de <code>ls -laF</code>	24
4	Détail d'une entrée crontab	26
5	Fenêtre typique XEmacs 20.20 sous X11	33
6	Exemple d'écran X11 géré par le gestionnaire de fenêtre fvwm	53
7	Fenêtre typique xfm sous X11	60
8	Fenêtre typique ghostview sous X11	61
9	Fenêtre typique xv sous X11	62
10	Connectivité internationale	65
11	Exemple de configuration d'un réseau IP	66
12	Classes d'adresses IP originelles	67
13	Exemple d'envoi de message privé avec mail	73
14	Entête et corps d'un message mail	76
15	Grappe de connectivité mondiale Usenet	79
16	Fenêtre typique xarchie sous X11	85
17	Fenêtre xterm contenant lynx sous X11	93
18	Fenêtre typique de Mosaic sous X11	94
19	Fenêtre typique de Netscape sous X11	95
20	Exemple d'une page HTML	96

Liste des tableaux

1	Les flux standards	20
2	Les opérations de flux standards	20
3	Résumé des caractères spéciaux pour les regexp	21
4	Résumé des caractères spéciaux pour le shell	21
5	Extensions des différents formats	42
6	Comparaison d'algorithmes de compression	43
7	Quelques expressions courantes	72
8	Clients mail pour ordinateurs personnels	73
9	Adresses mail usuelles des administrateurs	75
10	Clients news pour ordinateurs personnels	84
11	Serveurs IRC EFnet par pays	98
12	Normes ITU-T et standards propriétaires	103
13	Standards de correction d'erreur et de compression de données	103
14	Protocoles de transfert courants	105

1 Introduction

1.1 Buts de ce résumé

Ce résumé traite des systèmes UNIX, ainsi que des réseaux de communication habituellement disponibles sous les environnements UNIX.

Pour des raisons pratiques, il est assez orienté vers le matériel et le logiciel à disposition. De plus, il effectue un bref survol de quelques-unes des caractéristiques essentielles du système.

Il peut être utile dans de nombreux cas de se référer à des documents comme ceux indiqués en fin de ce texte dans la section Bibliographie, ainsi qu'aux nombreux documents électroniques cités.

Ce document se veut une simple introduction à UNIX et au monde des réseaux. Il ne parle pas des logiciels non standards ou spécialisés (comme des compilateurs, des outils d'administration, etc.). Par contre, il intègre un certain nombre d'utilitaires et de logiciels importants pour l'utilisateur d'un système UNIX.

1.2 Conventions typographiques

Afin de différencier, particulièrement dans les exemples d'exécution de commandes, ce qui doit être entré tel quel au clavier de ce qui ne sert qu'à la compréhension ou qui correspond à des messages envoyés par le système, des conventions typographiques ont été utilisées. Ces conventions sont les suivantes :

Le fait qu'une partie de phrase soit imprimée en style machine à écrire *typewriter* indique que les caractères sont à taper tels quels au clavier. En général, pour confirmer l'entrée, il faut appuyer sur *RETURN*. Cette touche se nomme parfois *ENTER* ou est parfois identifiée par une flèche et se trouve en général dans la région médiane de la droite du clavier.

Lorsque le système n'affiche pas d'écho des caractères reçus (comme lors de l'entrée d'un mot de passe), la police de caractère sera plus petite, comme dans cet exemple.

Les espaces, également à taper tels quels, seront représentés par `□` lorsqu'une confusion est possible. En général on se rappellera que toute commande doit être séparée de ses arguments par des espaces.

Un mot s'écrivant **ainsi** se retrouve automatiquement dans l'index et possède probablement une entrée dans le glossaire. Les termes anglais notés comme tels dans le texte sont également listés dans l'index.

Quand un texte est affiché en réponse à une commande ou comme invitation à entrer des données, ce style est utilisé. Il peut arriver que le texte indiqué ici soit légèrement différent de ce qui est affiché (en particulier il peut être plus court).

Lorsque la construction *CTRL-X* est utilisée, elle signifie que l'on doit appuyer simultanément sur les touches *CTRL* et *X* du clavier.

Les touches à taper individuellement au clavier, sans ajouter de *RETURN* final, sont représentées *ainsi*.

La convention UNIX de faire suivre les commandes par le numéro de la section de manuel dans laquelle la commande est définie n'a pas été observée au long de ce manuel : en effet, la plupart des commandes man actuelles recherchent relativement rapidement dans les répertoires de manuel, de plus, la commande `man -k` permet de connaître ce numéro. Enfin, la plupart des commandes présentées font partie de la section 1.

Les acronymes en majuscules représentent en général des protocoles, en minuscule les programmes concernés (par exemple FTP et ftp). Lorsque des conventions particulières s'appliquent, elles sont respectées.

Exemple

```
login: nom_utilisateur
password: mot_de_passe
BIENVENUE
> date
Thu Jan 5 10:11:23 CET 1995
```

1.3 Structure

Ce document est composé de trois parties: la première offre une vue d'ensemble du système UNIX, la deuxième se veut une vue d'ensemble rapide du système de fenêtrage **X11** et la troisième montre quelques possibilités des réseaux disponibles sous UNIX, comme **Internet**, qui sont les services distribués de messagerie et les différents services interactifs. Il comprend également un glossaire, contenant les mots les plus importants et les plus utilisés dans le monde UNIX en général, suivi d'informations quant aux documents disponibles en ligne (anglais: *online*) sur notre système, et d'une bibliographie de quelques livres sur UNIX et la téléinformatique. Enfin un index reportant aux termes en gras dans le document termine celui-ci.

1.4 Versions disponibles

La version de référence est la version **PostScript**. Il existe également des versions **DVI**, texte et **HTML** (créée automatiquement par le logiciel `latex2html`). Le serveur WWW officiel est :

`http://www-internal.alphanet.ch/~schaefer/unixguide.html`

A chaque version est associé un numéro de version et une date de parution. En général, un document datant de plus de 4 mois est obsolète.

1.5 Distributions particulières

Dans la mesure du possible, j'aimerais être informé lorsque ce document est utilisé à grande échelle, même lorsque l'utilisation est conforme à la licence. Lorsque l'utilisation est non conforme, merci de me demander la permission.

Diffusions particulières qui m'ont été annoncées :

- Le Département Electricité Diffusion de SCHLUMBERGER, représenté par Damien GIBAUD.
- Diffusion papier avec un CD-ROM **Linux** par une société francophone (avec versement au projet **GNU** de 6 FF par exemplaire vendu), Michel BARTHÉLÉMY <mb@tsmotel.gatelink.fr.net>.
- Diffusion dans le CD-ROM INET96 de l'**ISOC** (voir <http://www.crim.ca/inet96/>).

1.6 Adresse postale de contact

Vous pouvez contacter l'auteur à l'adresse postale suivante si vous n'avez pas accès au courrier électronique. N'hésitez pas à me communiquer les erreurs que vous avez trouvées dans le document ou les sujets qui vous intéresseraient mais qui n'ont pas été traités. Les cartes postales de remerciement sont évidemment aussi les bienvenues¹.

Marc SCHAEFER
Tempé 8
CH-2520 La Neuveville
Suisse

1.7 Comment désigner ce document ?

La dénomination complète est *Courte information sur le système UNIX et les réseaux téléinformatiques sous environnements UNIX*. Il existe une dénomination plus courte, soit le *Guide UNIX + Réseaux*. Personnellement, je l'appelle plutôt *unixguide* et les adeptes des acronymes à trois lettres peuvent l'appeler *uxg* (à prononcer en minuscule, bien sûr).

1.8 Corrections et suggestions

Envoyez vos suggestions à l'adresse électronique guide-unix@alphanet.ch. Si vous voulez faire partie de la nouvelle équipe d'édition, envoyez un mail à guide-unix-request@alphanet.ch.

¹Je ne m'engage pas à répondre à tous les messages, bien entendu, mais je ferai mon possible.

2 Historique

2.1 Introduction

Cette section se veut informative sur les événements qui ont créé les outils qui sont actuellement disponibles. Elle a également pour but de faire mieux comprendre l'esprit des réseaux et des utilisateurs de systèmes UNIX.

On trouvera plus de renseignements dans les nombreuses **FAQs** du **forum** de **news news.announce.newusers**, la plupart écrites par Gene SPAFFORD.

2.2 Origine du système UNIX

UNIX² est un vieux système d'exploitation. Révolutionnaire pour l'époque, il a cependant bénéficié (et bénéficie toujours) de la recherche. Il est intéressant de voir qu'un projet déconsidéré au départ ait pu influencer à ce point les systèmes d'exploitation et rester le plus répandu en terme d'architectures supportées.

UNIX a été développé dès 1969 sur un vieux **PDP-11** qui traînait dans un coin par Ken THOMPSON et Dennis RITCHIE des laboratoires Bell. Le nom UNIX vient d'un jeu de mots sur **MULTICS** (Multiplexed Information and Computing System), un grand (certains disent éléphantique) système d'exploitation, qui était en développement à cette époque. On ne niera pas le fait que presque tous les mécanismes de UNIX s'en sont inspirés³.

Les premières versions ne furent pas diffusées en dehors des laboratoires Bell. La V4, contrairement aux précédentes, écrites en assembleur PDP-11, fut écrite en **C**, ce qui facilitait énormément le portage vers d'autres matériels. La V6 fut diffusée dans les universités (d'où l'apparition de la variante **BSD** de UNIX). La version V7 de 1979 comportait **UUCP**, un **shell** et un compilateur **C K&R** (une version antérieure au **C** normalisé **ANSI**, du nom de ses concepteurs, Brian KERNIGHAN et ... Dennis RITCHIE).

Pendant les années 80, de nombreuses améliorations sont apportées à UNIX, par **USL**, appartenant à **AT&T**. On passe de System III (1982), le premier UNIX commercial d'AT&T, aux nombreuses versions de System V (1983) dont System V Release 2 (1984), Release 3 (1986), et Release 4 (1988).

Les universités (principalement Berkeley) apportent de nombreuses améliorations, de même que les constructeurs (comme **NFS** par Sun Microsystems, ou le consortium **OSF** pour **DCE**). Des comités de normalisation définissent les bases de standards UNIX (**POSIX**, etc.). On intègre à UNIX les protocoles **TCP/IP** ainsi que de nombreuses améliorations de l'interface utilisateur (**X11**). Des versions librement redistribuables voient le jour (**Linux**, **NetBSD** et **FreeBSD**). En 1993, Novell rachète la marque commerciale et la cède au groupe indépendant de normalisation **X/Open**.

Comment peut-on évaluer le futur de UNIX ? Les développements récents des systèmes parallèles ont amené des versions proposant les **threads** (sorte de processus léger), ou même le **multi-threading** au sein du noyau, avec le support des architectures multiprocesseur. Des noyaux légers (micronoyaux) comme **Mach** ou **Chorus** sont aussi proposés. Enfin, la configurabilité de la machine (noyaux configurables sans recompilation⁴, grâce aux **daemons** externes et aux mécanismes de **STREAMS**) est un des buts des systèmes récents.

Les trois versions principales de UNIX sont actuellement System V Release 4.2, BSD 4.4 et OSF/1 3.2D. De très nombreuses implémentations empruntant à l'une ou plusieurs de ces versions ont vu le jour, beaucoup étant librement redistribuables.

2.3 Origine de Usenet et de UUCP

Fin 1979, juste après la diffusion de la version V7 de UNIX qui comprenait le logiciel **UUCP**, deux universitaires de l'université Duke en Caroline du Nord aux Etats-Unis, Tom TRUSCOTT et Jim ELLIS, eurent l'idée d'interconnecter deux ordinateurs afin d'échanger des informations dans la communauté UNIX. Steve BELLOVIN de l'université de Caroline du Nord implémenta la première version du système de **news** à l'aide de **scripts shell** et les installa sur les deux premiers sites *unc* et *duke*. Peu après, la première version en langage **C** fut distribuée. Des modifications furent effectuées par Tom TRUSCOTT et distribuées sous le nom **A news**.

Au début on utilisa ce système pour échanger des modifications au système d'exploitation UNIX. Ensuite, on l'utilisa pour des discussions générales.

En 1981, le volume de news devenait trop grand pour **A news**. Deux étudiants de Berkeley développèrent **B news**. Ce logiciel fut amélioré par la suite en ajoutant les **forums** modérés (anglais: *moderated*), la structure hiérarchique de nommage et de nouveaux mécanismes de contrôle, en s'inspirant des listes de distribution (anglais: *mailing-list*) d'**ARPAnet**.

Les versions actuelles du système de news sous UNIX sont **C news** et **INN**, ce dernier étant plus courant sur les sites **Internet**. Les **RFC-1036** et **son of 1036** définissent les formats des articles de news.

A l'origine, les **news** étaient transférées par des sessions **UUCP**. En 1986, la spécification du protocole **NNTP** allait permettre l'utilisation d'**Internet** comme couche de transport des news de façon standard.

La philosophie de **Usenet** est cependant très différente de celle d'**Internet** : Usenet n'a jamais été commercial et, pendant très longtemps, a été transporté aux frais des participants. Au contraire, Internet a été soutenu dès le début par les entreprises et l'Etat : il n'existait pas d'esprit de communauté comme dans Usenet. Il faut savoir que Usenet a été pendant longtemps *persona non grata*

²On écrit ce mot en majuscules, même si ce n'est pas un acronyme, car c'est l'usage, mais personne ne vous en voudra de l'écrire Unix.

³La légende prétend que UNIX fut nommé ainsi parce que **MULTICS** faisait beaucoup de choses mal : UNIX ne devait alors faire qu'une seule chose, mais bien.

⁴Le futur système d'exploitation **HURD** du projet **GNU** offre, entre autres, cette possibilité (voir section 2.5.3 page suivante). Le système d'exploitation **Linux** offre déjà la possibilité de chargement dynamique de parties du système d'exploitation sous formes de **modules** et de pilotes de périphériques (anglais: *drivers*) en mode utilisateur (ne pas confondre avec les bibliothèques partagées dynamiques (anglais: *shared libraries*) qui ne sont pas une nouveauté sous UNIX).

sur le réseau **ARPAnet** pour des raisons diverses, notamment le temps “perdu” à lire de nombreux messages “inutiles” (ou du moins sans rapport direct avec la fonction de l’employé).

2.4 Origine d’Internet

ARPAnet est, en 1969, un projet de recherche **DARPA**, du nom d’une agence de recherche du département de la défense du gouvernement des Etats-Unis. Le projet a pour but de maintenir les communications entre les ordinateurs militaires en cas de catastrophe majeure⁵ (contexte de la guerre froide). Pour ce faire, on évite les connexions point-à-point pour privilégier un protocole permettant d’adresser n’importe laquelle des machines d’un réseau, en utilisant des liens de secours avec un routage dynamique. Sous mandat du **DARPA**, des organismes de recherche gouvernementaux, privés et académiques développent diverses propositions de protocoles de transport et de haut niveau, décrits dans les **RFCs** et adoptés d’une manière relativement démocratique par les utilisateurs finaux et les développeurs.

Au départ, on utilise **ARPAnet** pour des sessions distantes (anglais: *remote login*), le courrier électronique et les **mailing-lists** techniques. Très vite, le réseau s’ouvre au monde académique et à celui de la recherche : les utilisateurs finaux sont déjà présents. On développe le protocole **TCP/IP**. **Usenet** est de plus en plus transféré à travers Internet⁶. De nouveaux protocoles sont développés. Le réseau s’ouvre aux commerciaux, des **backbones** entiers sont privatisés. En 1990, le CERN développe le système hypertexte **WWW**. Le 27 mars 1995, le projet initial du **DARPA** se termine et le réseau subventionné **NSFnet** est dissous. Il reste cependant encore de nombreux réseaux à buts non lucratifs ou de recherche, mais le **backbone** Internet est désormais en (de nombreuses, heureusement) mains privées.

2.5 La fondation FSF et le projet GNU

2.5.1 Qu’est-ce que la FSF ?

La FSF (Free Software Foundation) a été fondée au début des années 80 par Richard M. STALLMAN (aussi connu sous son nom d’utilisateur : `rms`). Le but de cette fondation est de développer des logiciels *libres*. Ce sont des logiciels que vous avez le droit de copier, utiliser, modifier et redistribuer comme bon vous semble. La seule condition est que les sources de ces logiciels doivent être disponibles gratuitement pour quiconque en fait la demande.

Il est important de comprendre que le mot *Free* dans *Free Software Foundation* ne doit pas être traduit comme *gratuit* mais bien comme *libre*. Ces logiciels peuvent tout à fait être vendus, bien que ce ne soit que rarement le cas en pratique⁷, car il existe toujours un moyen légal, en vertu de la licence, de se les procurer gratuitement.

2.5.2 Que sont la GPL et la LGPL ?

La **GPL** (General Public License) est une licence qui spécifie les conditions de distribution de tous les logiciels GNU. La **LGPL** (Library General Public License) est son équivalent pour les bibliothèques de sous-programmes.

En gros, ces licences spécifient que les logiciels GNU peuvent être copiés, modifiés et redistribués de quelque manière que ce soit, tant et aussi longtemps que les sources sont disponibles gratuitement. Toutefois, pour plus de détails, il faudra vous reporter aux textes originaux. Ces textes sont distribués avec tous les logiciels GNU ou peuvent être obtenus directement à la FSF.

Le gros avantage des logiciels distribués selon ces conditions est que si vous désirez les améliorer, vous pouvez le faire, puis distribuer votre nouvelle version. De ce fait, tout le monde en profitera. Cela conduit à d’excellents programmes écrits par des dizaines de personnes différentes.

2.5.3 Qu’est-ce que le projet GNU ?

Le projet **GNU** (GNU is Not UNIX) est un projet de la **FSF** dont le but est de développer un système d’exploitation complet, distribué selon les conditions de la **GPL**. Ce système d’exploitation reprend un certain nombre de concepts de UNIX, mais ce n’est pas UNIX (d’où le nom⁸ GNU is *not* Unix). Richard STALLMAN a commencé ce projet seul, juste après avoir créé la FSF. La première partie de ce projet consistait à écrire un éditeur avec lequel il puisse éditer ses programmes. Cet éditeur est le bien connu GNU **Emacs**. Après cela, il a écrit un compilateur **C** pour pouvoir compiler son système d’exploitation. Cela a donné le fameux **gcc**. Depuis lors, un certain nombre de personnes se sont jointes à lui pour écrire toutes sortes de programmes. Le kernel lui-même, nommé **HURD**, est en passe d’être terminé et devrait bientôt être disponible.

2.5.4 Les principaux logiciels

Voici une liste non-exhaustive et totalement subjective des principaux logiciels du projet **GNU** :

Emacs éditeur de texte très puissant et complètement programmable (*voir section 6 page 33*) ;

⁵Il est intéressant de constater que lors du tremblement de terre de 1993 en Californie, les seules liaisons épargnées (ou rétablies le plus rapidement) étaient souvent les lignes louées **Internet**, les centraux n’étant rétablis que plus tard.

⁶Un internet est un ensemble de réseaux accolés. **Internet** (ou l’Internet) est le réseau coopératif mondial reliant près de 70’000 réseaux entre eux.

⁷Certains fournisseurs de logiciels vendent cependant des adaptations de ces logiciels pour des plates-formes non supportées.

⁸Très fortement récuratif, comme le langage **Lisp** dans lequel on programme l’éditeur **Emacs**.

gcc excellent compilateur pour les langages **C**, **C++** et **Objective-C**. C'est le compilateur C le plus utilisé sous UNIX du fait de la très grande qualité du code qu'il produit. Il est de plus très utilisé dans le domaine de la recherche, ses sources étant disponibles et rapidement modifiables. Il est en outre très facile de l'adapter pour lui faire produire du code pour un nouveau processeur, ce qui en fait bien souvent le premier compilateur disponible sur un nouveau processeur ;

gdb un **debugger** interactif pour les programmes produits par **gcc**. Mêmes remarques que pour ce dernier ;

calc une calculatrice très puissante fonctionnant dans **Emacs**. Elle permet entre autres de faire du calcul matriciel, de la résolution d'équations, du calcul symbolique, etc. ;

gnuplot dessinateur de graphes bi- et tri-dimensionnels très complet. Notez que ce logiciel n'est pas GNU, mais distribué par GNU.

De plus, il existe également des versions GNU de la plupart des utilitaires UNIX. Ces versions sont souvent bien plus puissantes (et fiables !) que leurs équivalents propriétaires, et peuvent remplacer des commandes telles que `tar`, `awk`, `yacc`, `lex`, `find`, `ls`, `make`, etc. Vous trouverez une liste des logiciels GNU et d'autres logiciels libres sur :

<http://www.gnu.org/software/software.html>.

2.5.5 Pour en savoir plus

Pour informer le public de l'état du projet GNU, la FSF publie deux fois par année un bulletin d'information (le **GNU Bulletin**). Ce bulletin paraît en janvier et en juin de chaque année; il est entre autres distribué via le forum **gnu.announce**.

Vous pouvez poser des questions sur le forum **gnu.misc.discuss**, sans oublier qu'il s'agit d'un forum public et que certaines règles de savoir-vivre s'imposent (*voir section 23.5 page 81*).

En dernier recours, vous pouvez également contacter la **FSF** directement à l'adresse postale suivante :

Free Software Foundation, Inc.
59 Temple Place - Suite 330
Boston, MA 02111-1307
USA

La FSF existe également en Europe depuis quelques temps : <http://www.fsfeurope.org/>.

2.5.6 Comment soutenir la FSF ?

Si vous ou votre employeur pensez que les excellents programmes GNU que vous utilisez méritent une récompense, vous pouvez soutenir la FSF de plusieurs manières différentes :

1. En envoyant de l'argent à la FSF.
2. En donnant du matériel informatique à la FSF.
3. En prêtant des programmeurs à la FSF.
4. En achetant des cassettes de distribution de programmes à la FSF.

Dans tous les cas, consultez le bulletin GNU ou contactez la FSF pour savoir exactement comment vous y prendre.

Première partie

UNIX

3 Le système UNIX

3.1 Introduction

UNIX est un système d'exploitation multi-utilisateur et multitâche, avec mémoire virtuelle et mémoire protégée, ainsi que de nombreuses facilités (comme les services d'accès au réseau).

Dans le passé, tous les constructeurs informatiques ont offert une version de UNIX⁹ comme par exemple les divers dérivés de **BSD**, **OSF/1**, **AIX**, ou **System V**.

UNIX va de pair avec le développement de réseaux de communication à l'échelle planétaire, comme **Internet**, un grand réseau temps réel, ou, par le passé, le réseau non interactif **UUCP**.

Aujourd'hui, alors que de nombreux constructeurs ont disparu ou se sont réorientés vers les services, on retrouve UNIX à la fois sur les gros systèmes, les stations de travail et les ordinateurs personnels : **Linux**, en particulier, est probablement le système UNIX en plus grande croissance aujourd'hui.

L'avantage de UNIX par rapport aux autres systèmes d'exploitation, outre le fait qu'il est standardisé de façon ouverte, ne dépend pas d'un seul fournisseur et qu'il est très répandu, est sa puissance alliée à sa simplicité : *one program does one thing well* : un programme fait une seule chose, mais bien, et l'on peut combiner les programmes. De plus un très vaste réservoir de logiciels existe, dont la plupart sont librement redistribuables. La plupart des logiciels écrits pour UNIX sont portables, relativement facilement suivant le type de l'application, du petit système (par exemple **Linux** sur PC) jusqu'au très gros système (**UNICOS** sur Cray). La plupart du temps, une simple recompilation suffit. Parfois il faut modifier les sources. Il faut également noter que de plus en plus de versions du système UNIX lui-même sont disponibles sans aucune licence : le système **Debian GNU/Linux**, composé de milliers de programmes (ou **paquets**, anglais: *packages*) libres est très populaire aujourd'hui.

3.2 Particularités

3.2.1 Caractéristiques du système d'exploitation

Grâce au système multi-utilisateur, plusieurs utilisateurs peuvent travailler simultanément sans se gêner, et peuvent décider de la manière d'échanger leurs fichiers. Il est par exemple possible de restreindre¹⁰ l'accès à des fichiers pour un groupe de travail ou d'intérêt : par exemple sur **vulcan**, les membres du club font partie du groupe `members` et peuvent donc accéder aux fichiers présents dans le répertoire `/users/admin/admin` concernant le club, et eux seulement. Le système multitâche permet quant à lui d'exécuter en parallèle plusieurs tâches, ou de demander l'exécution différée de commandes (travail **batch**). Les réseaux de communication mondiaux permettent d'échanger des informations très rapidement.

Un des aspects frappants du système UNIX est également la généralisation du concept de fichier : les disques, les terminaux (**ttys**), les connexions réseau (**sockets**) et même parfois certaines fonctions spéciales (liste des processus, mémoire disponible) sont des fichiers¹¹.

3.2.2 Problèmes courants

UNIX est sensible à la casse (anglais: *case-sensitive*), cela signifie que si une commande, un paramètre, ou un nom d'utilisateur est écrit en minuscules, alors il doit être entré en minuscules.

Pour interrompre de façon quelque peu sauvage tout programme il suffit de taper **CTRL-C**¹².

Le jeu de caractères utilisé est **ASCII**, avec des accents selon **ISO Latin 1**¹³. La section 7.3 on page 42 donne quelques informations sur les logiciels de conversion de formats disponibles.

3.2.3 Arborecence

Le système UNIX définit les systèmes de fichiers depuis une racine unique¹⁴ appelée **root** et notée `/`. Depuis cette racine il est possible de se déplacer dans les répertoires systèmes ou utilisateurs pour autant que les permissions associées le permettent. Ces répertoires forment une réelle arborescence¹⁵, les feuilles étant les fichiers et les noeuds des **répertoires** (**répertoire**, ou **dossier**; anglais: *directory*). La figure 1 on the next page donne un exemple d'arborescence et de nommage de sentiers ou chemins (anglais: *path*).

⁹Les services standards sont normés par **X/Open**, les différences proviennent surtout des ajouts.

¹⁰`chmod` (voir section 5.9.7 page 25).

¹¹Penser simplement au répertoire `/proc` de certains systèmes modernes, comme **Linux**.

¹²On peut aussi utiliser la séquence **CTRL-Z** puis `kill %%` (voir section 5.7 page 21). Cela est redéfinissable (voir section 5.9.33 page 30).

¹³Certaines variantes de UNIX n'utilisent pas ce jeu standard par défaut, p.ex. **HPUX**.

¹⁴Oubliez dès à présent le concept bâtarde de **lecteur** offert par d'autres systèmes, qui mélange allégrement périphériques physiques, partitions et systèmes de fichiers.

¹⁵En informatique, les arbres ont la fâcheuse habitude de pousser vers le bas.

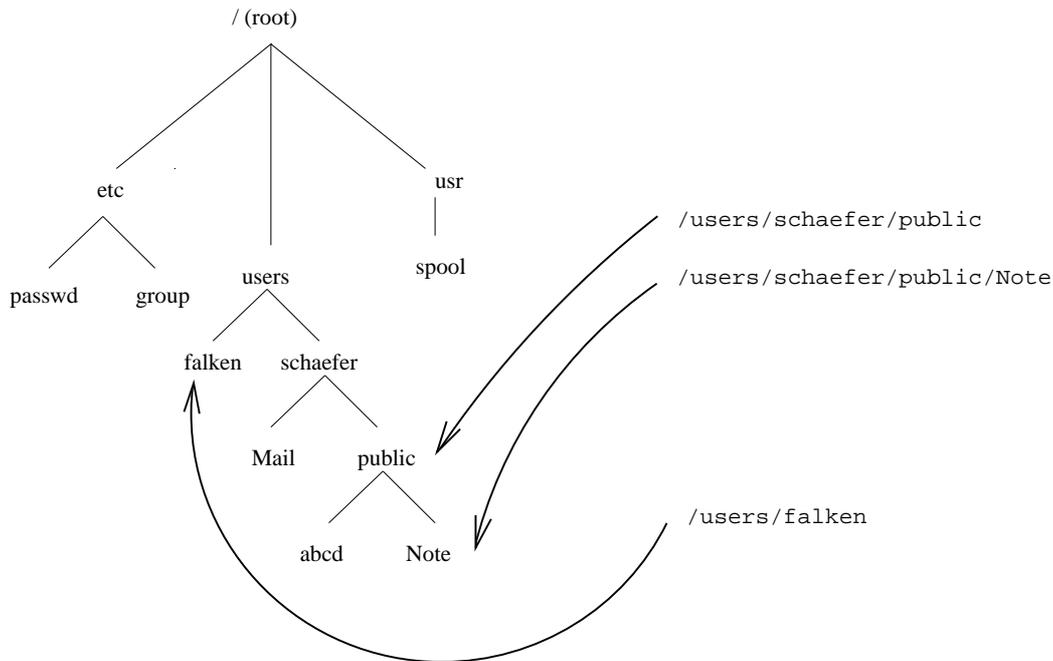


Figure 1: Arborescence du système de fichiers

La commande `cd` permet d'effectuer la navigation dans les différents répertoires. A titre d'information, les principaux répertoires système sont

- /dev périphériques système,
- /etc administration système,
- /bin commandes et utilitaires,
- /var données, fichiers de logs et **spool**, en particulier qui changent souvent.
- /usr/bin autre répertoire de commandes,
- /tmp répertoire de fichiers temporaires,
- /users ou /home répertoires utilisateurs,
- /usr/local ou /share exécutables, sources, archives, bibliothèques, documentations et manuels partagés installés localement,
- /news **spool** des **news** (souvent dans /usr/spool/news).
- /usr/spool/mail **spool** des **mails** (souvent dans /usr/mail).

3.2.4 Noms de fichiers et hiérarchie

Les noms de fichiers UNIX peuvent très bien comporter des caractères et des longueurs qui ne sont pas compatibles avec les systèmes de fichiers d'autres environnements. Par exemple, les noms de fichiers sont limités à 255 caractères (**FFS**, **UFS**, **ext2**) et sont sensibles à la casse.

Pour spécifier un nom de fichier, il est possible de donner soit un chemin relatif, soit un chemin absolu. Un **chemin absolu** commence par la **racine**, notée / et indique univoquement le chemin à parcourir depuis celle-ci jusqu'au répertoire ou fichier désiré. Par exemple /users/schaefer/Mail/un_fichier est un chemin absolu jusqu'au fichier un_fichier du répertoire Mail, lui-même contenu dans le sous-répertoire schaefer du répertoire users. Un **chemin relatif** commence par un nom de fichier ou de répertoire, ou éventuellement par . ou .. qui représentent respectivement le **répertoire courant** (anglais: *working directory*, **répertoire de travail**) et le **répertoire père** (celui qui se trouve immédiatement au-dessus du répertoire courant dans la hiérarchie). Par exemple le fichier vu précédemment de façon absolue peut également être atteint de façon relative comme suit, en supposant que le répertoire courant soit /users/schaefer :

Mail/un_fichier ou ./Mail/un_fichier ou encore ../schaefer/Mail/un_fichier. On peut ajouter autant de ./ que désiré (ne change rien), ou des combinaisons de, ../ et de spécifications de répertoire.

Attention, le souligné représente dans les exemples précédents un véritable caractère de soulignement. Il faut noter qu'inclure des espaces dans des noms de fichiers sous UNIX peut compliquer passablement leur accès, car il faudra toujours les entourer de guillemets, vu que le **shell** utilise le caractère espace comme séparateur.

Pour changer le **répertoire courant**, on utilise la commande `cd` avec comme paramètre le chemin (relatif ou absolu) du **répertoire** où l'on veut se rendre : par exemple pour aller dans le sous-répertoire public on écrira : `cd public` (où le caractère représente la touche **ESPACE**).

Pour savoir où l'on se situe dans la hiérarchie, on peut utiliser la commande `pwd`. Le répertoire courant est normalement intégré également dans le **prompt** du **shell**. Pour revenir en arrière d'un sous-répertoire dans la hiérarchie, on peut utiliser `cd` `..` (soient deux points).

Le **répertoire par défaut** (anglais: *home directory*) est le répertoire où l'on se trouve à la connexion. On peut y retourner en entrant une des commandes suivantes, si l'on suppose que l'utilisateur courant est `schaefer` :

- `cd`
- `cd ~`
- `cd $HOME`
- `cd ~schaefer`

Il faut se rappeler que UNIX est sensible à la casse également pour les noms de fichiers. Ainsi `toto` et `Toto` représentent des fichiers différents. Une convention intéressante est de mettre la première lettre des répertoires en majuscule, comme dans `Mail`, `News`, `Projets`. On peut aussi utiliser la forme typée de l'utilitaire `ls` avec `ls -F`.

Une convention qui peut être appliquée localement est de créer deux répertoires standards : `public`, avec accès libre pour les autres utilisateurs et `private` non accessible par les autres utilisateurs (*voir section 5.9.7 page 25*).

3.2.5 Notions d'utilisateur et de groupe

UNIX définit des attributs pour chaque utilisateur du système, et ceux-ci définissent également l'accès relatif des utilisateurs entre eux. Chaque utilisateur est caractérisé par les informations suivantes :

loginname nom d'utilisateur, par exemple `schaefer`

group nom(s) de groupe(s) de travail, par exemple `members`

home répertoire par défaut, par exemple `/users/schaefer`

shell nom de l'interface de commande, par exemple `/bin/tcsh`

username nom et prénom

office par exemple le lieu d'appel

On peut obtenir ces informations¹⁶ par la commande `finger` appliquée au `loginname` ou à une partie du nom d'utilisateur. Par exemple on pourrait taper `finger schaefer`, ou consulter les fichiers `/etc/passwd` ou `/etc/group` à la main¹⁷. Pour vous-même, vous pouvez utiliser `id` et `whoami`. Chaque fichier, ou répertoire, appartient à son créateur, de plus le créateur du fichier peut définir des protections concernant lui-même, son groupe, ou les autres utilisateurs. La commande `chmod` permet de changer ces protections (ou modes) (*voir section 5.9.7 page 25*).

Des systèmes de fichiers avancés permettent également d'implémenter une gestion des droits d'accès plus fine que celle basée sur les permissions UNIX, avec comme inconvénient de rendre celle-ci également plus lourde et moins synthétique. On appelle cela les **ACLs**.

4 Accès à un système UNIX

Cette section a pour but de montrer l'accès à un système UNIX, via terminal texte (**tty**, anglais: *teletype*). C'est la méthode utilisée si l'on se connecte par **modem**, terminal ou connexion réseau (p.ex. via **SSH** ou **TELNET**).

4.1 Appel du système par modem

Il faut configurer votre terminal afin d'avoir les paramètres suivants :

- 8 bits de donnée,
- 1 stop bit,
- Pas de parité,
- Emulation **VT100** (d'autres sont possibles (*voir section 8 page 45*)),

Bien souvent, un tel système qui répond par modem supporte les logins standards UNIX (interactifs (shells), non interactifs (**UUCP**) et multi-sessions/réseau (**SLIP** et **PPP**) ainsi que **FidoNet crashmail** et réception/émission de **faxes**, le tout grâce à de nombreux logiciels libres dont **mgetty**, un logiciel de réception d'appels.

4.2 Connexion distante au système

Par exemple, on utilisera **SSH** :

```
schaefer@defian: % ssh vulcan.alphanet.ch -l falken
```

¹⁶sauf le groupe

¹⁷seulement possible sans **NIS**

4.3 Le login

Le système répond et quelques instants plus tard, la connexion est établie. Le système affiche ensuite un message de bienvenue et un prompt login:¹⁸. Il faut entrer votre nom d'utilisateur, en minuscules. Ensuite, un mot de passe vous sera demandé avec Password:. Il n'y aura pas d'écho, et si vous vous trompez¹⁹, après quelques secondes la même demande reviendra. Dès que le login est réussi, le système affiche un autre message de bienvenue et vous indique si vous avez du courrier électronique privé et un éventuel dépassement du quota disque. Ensuite le système vous présente le prompt standard du **shell**. Notez qu'une erreur courante au **login** est de taper votre nom d'utilisateur en majuscules. En général, cela provoquera des effets indésirables (affichage des textes en majuscules exclusivement). On peut s'en sortir, sur une console exclusivement, avec *CTRL-D*.

Exemple: L'utilisateur *falken* tente une connexion. On remarque qu'il a entré un mauvais mot de passe au premier essai. Le mot de passe choisi est cependant peu sûr (voir section 9.2 page 50).

```
login: falken
Password: martin
Login incorrect
login: falken
Password: joshua
Welcome to NORAD, Professor Falken.
```

4.4 Fin de session

La fin de session s'obtient en quittant le programme courant (par exemple avec *x*, *q*, *.* ou un équivalent, la touche *CTRL-C* pour un arrêt sauvage), puis en quittant le ou les shells avec *exit* (ou avec *CTRL-D*) jusqu'à ce que le système affiche le message de **logout** et raccroche. Raccrocher sans avoir terminé ses sessions²⁰ peut être dangereux du point de vue de la sécurité, même si la détection du raccrochage ne pose aucun problème en général au système.

Exemple: L'utilisateur *falken* quitte le système.

```
<vulcan> falken:/d2/users/falken> exit
logout
2.3u 4.0s 1:14:07.83 0.1% 0+218k 20+0io 18pf+0w
NO CARRIER
```

On remarque le format étrange de l'avant-dernière ligne: il s'agit de statistiques²¹ d'utilisation du système. Par exemple, on apprend que l'utilisateur *falken* est resté connecté pendant 1 :14 :07, qu'il a utilisé 0.1% du temps de calcul de la machine, correspondant à 2.3 secondes de temps CPU utilisateur et 4 secondes système. Il a utilisé 218KB de mémoire non partagée, causé 20 opérations d'entrée, causé 18 fautes de page (mémoire virtuelle) et n'a jamais été remplacé complètement en mémoire par un autre programme.

Ces statistiques sont appelées par un script de déconnexion, configurable pour chaque utilisateur comme on le verra par la suite (voir section 8 page 45).

5 L'interface de commande (shell)

5.1 Introduction et familles de shells

L'interface de commande, ou **shell**, permet d'entrer des commandes au clavier et de recevoir des messages en retour. Elle permet également de définir des **variables d'environnement**, comme par exemple le type de terminal. Il existe de nombreux shells aux syntaxes différentes : il y a cependant principalement deux familles de shells. La figure 2 page suivante donne une généalogie rapide de ces différents shells interactifs.

¹⁸Celui-ci peut commencer par une majuscule également.

¹⁹Notez que le système ne précise pas le type de l'erreur (utilisateur inexistant ou mauvais mot de passe) pour des raisons de sécurité.

²⁰On verra qu'il est possible de détacher ses sessions d'un terminal pour s'y réattacher ensuite, sans que ces sessions ne soient interrompues : (voir section 5.9.31 page 29).

²¹Sortie de la commande *time* sous **shell** *tcsh* exécuté dans le fichier *.logout* (voir section 8.1.3 page 46).

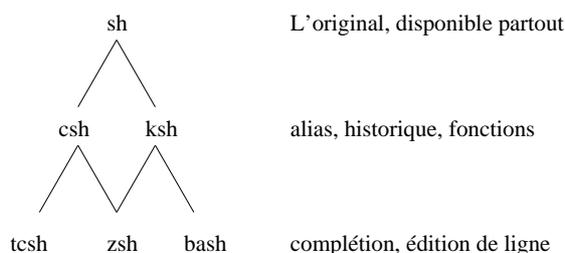


FIG. 2 – Généalogie des shells

En premier lieu, on utilisera les descendants du **Bourne shell**, soit **sh**, très spartiate, **ksh**, le **Korn shell**, normé **POSIX**, ainsi que le confortable et **GNU bash** (Bourne-again shell), disponible par défaut sous **Linux**, qui possède également un mode **POSIX**.

La deuxième famille se compose des **C-Shells**, soient par exemple **csh** et **tcsh**, qui originellement offraient une syntaxe d'écriture de script plus similaire au langage **C** et une interface plus confortable : l'arrivée de shells standards normés **POSIX** et de **bash** les a rendus moins intéressants.

Enfin, le shell le plus moderne, qui combine les deux familles est **zsh**. Citons également le shell en langage **Scheme**, **scsh**²².

Les shells dérivés de **sh** sont particulièrement conseillés pour la rédaction de courts programmes en langage shell nommés **scripts** (voir section 8.3 on page 47).

5.2 Choix du shell présenté

Le shell qui sera présenté ici est **bash**, le shell du projet **GNU**, disponible en logiciel libre sur de nombreuses plateformes et par défaut **Linux**. La plupart des concepts s'appliquent cependant à tous les shells.

5.3 Notions principales

Le shell définit les notions suivantes:

répertoire par défaut il s'agit du répertoire appartenant à l'utilisateur, accessible en tapant `cd`. Il correspond à la variable d'environnement **\$HOME** et à l'abréviation `~`. Un tel répertoire est une propriété d'un utilisateur.

répertoire courant le répertoire courant correspond à un dossier ouvert sur votre bureau. Pour connaître le répertoire courant, il faut taper `pwd`. Pour changer de répertoire courant, on utilise `cd`. Un répertoire courant est une propriété d'un processus.

variables Les **variables d'environnement** contiennent des informations comme le type de terminal et sont modifiables avec `env`, `set` et peuvent être exportées aux sous-processus du shell à l'aide de `export`. On obtient la valeur d'une variable en la précédant d'un dollar (`$`). Les variables d'environnement sont une propriété du processus.

historique Il est possible de voir les commandes tapées dans le passé avec la commande `history`. De plus les touches du curseur haut et bas permettent de se promener dans l'historique des commandes introduites. On peut également faire des recherches dans l'historique : par exemple à l'aide de `CTRL-R` puis en entrant du texte on fait une recherche inverse dans l'historique. On peut rejouer de l'historique facilement à l'aide de `!!` pour répéter la dernière commande, `!23` pour la 23^e commande ou `!mor` pour la dernière commande débutant par `mor`.

motifs du shell Il est possible d'utiliser `*` et `?`, entre autres, comme jokers remplaçant respectivement une suite de caractères quelconque ou un caractère. Enfin le caractère `~` sera remplacé par votre répertoire par défaut (typiquement : `/users/loginname` où `loginname` est votre nom d'utilisateur). De même `~user` sera remplacé par le répertoire par défaut de l'utilisateur `user`, bien souvent `/users/user`²³. Notons que ces motifs, jokers ou anglais: *wildcards* du shell ne sont pas exactement ce que l'on nomme souvent sous UNIX les expressions régulières, **regexps**. Celles-ci sont plus complexes mais également plus puissantes.

complétion La touche `TAB` permet de compléter le nom d'une commande ou d'un fichier en cours de saisie s'il n'y a pas de possibilité de confusion et que le répertoire courant est lisible.

raccourcis La commande `alias` permet de définir des équivalents de commandes. Par exemple `alias dir ls` permet de taper `dir` à la place de `ls`. Pour supprimer un alias, on utilise `unalias`. On peut aussi précéder la commande dont on veut utiliser la version originale par un `\` (ou par le sentier absolu, p.ex. `/bin/ls`).

scripts Il est possible de créer des fichiers de texte contenant une suite de commandes que l'on veut appeler (exécuter) de manière répétitive. On appelle ce genre de fichiers de commandes également des **batches**.

²²Classer **scsh** parmi les shells est un peu risqué, car ce shell n'est pas prévu pour l'usage interactif. Pour des exemples de scripts, voir <http://www.scsch.net/>.

²³Les distributions Linux préfèrent `/home/user`.

redirections Pour sauvegarder la sortie des commandes, il est possible d'utiliser les symboles de redirection `>`, `>>` et `<`, correspondant respectivement à la création, l'ajout à un fichier remplaçant la sortie standard écran et la lecture d'un fichier remplaçant l'entrée standard clavier. La notion de **pipe** (ou tuyau/tube) est aussi intéressante, permettant de connecter la sortie standard d'une commande à l'entrée d'une autre commande.

correction Certains shells peuvent corriger de façon limitée la syntaxe des commandes et éventuellement des arguments. Il présente alors une possibilité, et demande `y/n/e`²⁴

commandes multiples On peut indiquer des séquences de commandes avec le séparateur `;` ou à l'aide de **scripts**. On peut également exécuter conditionnellement des commandes avec les séparateurs `&&` et `||` : dans le premier cas, on n'exécute la deuxième commande que si la première a réussi. Dans le second, on n'exécute la deuxième commande que si la première n'a pas réussi (son code de retour est différent de zéro).

job control Il est possible de gérer plus d'un seul processus ou job dans le même **shell**.

expansion L'expansion des arguments des commandes, particulièrement en présence d'expressions régulières (**regex**) est effectuée par le **shell**. Il faut donc précéder d'un `\` les éventuels caractères **jokers** à interpréter par la commande (p.ex. **grep**). On peut aussi entourer l'argument par des guillemets, voir plus bas.

code de résultat Chaque commande offre un code de résultat qui peut être traité par l'appelant. Le code de retour est stocké dans la variable d'environnement spéciale `$_`. Lorsque cette valeur vaut zéro, la commande précédente a réussi.

guillemets Le rôle des guillemets est multiple: les guillemets doubles `''` empêchent l'expansion des jokers et la séparation d'arguments par des espaces, les guillemets simples `'` empêchent en plus les expansions de variables. Enfin, les guillemets inverses (```, anglais: *backtick*) exécutent la commande dans les guillemets et retournent la sortie standard comme expansion.

tuyaux Les tuyaux (anglais: *pipes*) permettent de connecter la sortie d'un programme à l'entrée d'un autre, afin d'effectuer une opération de flot complexe.

5.4 Historique, complétion, variables d'environnement, édition de ligne

Le shell permet d'entrer des commandes, mais aussi de rappeler des anciennes commandes avec la fonction historique, voire de les modifier. Il est possible par exemple d'afficher les anciennes commandes, avec la commande `history`, ou de remonter visuellement dans l'historique des commandes avec la touche de curseur pointant vers le haut en émulation **VT100**.

Il est également possible d'utiliser une option très intéressante, soit la complétion des noms de fichiers. En effet, si dans votre répertoire courant se trouve un fichier nommé `abcdef`, il suffit de taper `a` puis la touche `TAB` et `abcdef` sera affiché (sous certaines conditions, notamment d'unicité). Si le terminal émet un signal sonore, il peut être indiqué de taper `CTRL-D` pour obtenir les différentes possibilités que le **shell** a trouvées. On peut même configurer **tcsh** pour ne proposer que les types de fichiers (différenciés par leur extension) compatibles avec le programme concerné :

```
complete xdvi n/*/f:*.dvi/
```

On peut extraire la valeur des **variables d'environnement** en les précédant de `$`. On appelle ceci une **expansion**. On peut les utiliser pour stocker des valeurs intermédiaires dans des **scripts**, ou pour communiquer des paramètres (comme le type du terminal) aux programmes exécutés. Les commandes `env`, `set` et `setenv` permettent de modifier les variables d'environnement (voir section 8 page 45).

Le **shell** offre de plus des possibilités étendues d'édition de ligne à l'aide des touches du curseur présentes avec la plupart des émulations de terminal.

5.5 Redirections

Faisons quelques exemples :

`date` affiche la date courante sur le terminal.

`date |wc -c` compte le nombre de caractères dans la sortie de la commande `date` et affiche cette information sur le terminal.

`date --error` affiche une erreur sur le terminal.

`date --error |wc -c` affiche une erreur sur le terminal, et compte zéro caractères.

`date --error 2>&1 |wc -c` donne le nombre de caractères du message d'erreur.

Il y a trois fichiers (flux, flots) ouverts de façon standard lorsqu'une commande est lancée par le shell, en règle générale :

²⁴Avec **tcsh**, pour l'activer, si ce n'est pas déjà fait, utiliser `set correct=cmd` (voir section 8 page 45). Pour accepter, il faut taper `y`, sans appuyer sur `RETURN`.

Numéro	description	usuellement
0	entrée standard	le clavier
1	sortie standard	l'écran
2	erreur standard	l'écran

TAB. 1 – Les flux standards

Les opérations possibles sont :

Opération	implémentation	notes
ouverture	0< f, 1> f, 1>> f	entrée, sortie, sortie avec ajout sur f
fermeture	0<- , 1>-	
redirection	2>&1	la sortie d'erreur passe dans la sortie standard
HERE document	<<EOF...EOF	simule une entrée standard dans un script

TAB. 2 – Les opérations de flux standards

Séparer la sortie d'erreur de la sortie standard permet de pouvoir traiter les erreurs spécialement. Chacun de ces fichiers peut être fermé, ouvert sur un fichier différent, ou connecté à un autre de ces fichiers. On peut également définir des numéros de fichiers supplémentaires et les utiliser dans des programmes en C, Perl ou dans des scripts shell. Par exemple, le fichier 3 pourrait être défini comme un tube de communication à un script agissant comme point de rendez-vous.

Notez que l'ordre sur la ligne de commande a une importance lorsqu'on redirige des flux sur d'autres flux. Comparer par exemple :

```
date --error > /dev/null 2>&1
```

et

```
date --error 2>&1 > /dev/null
```

5.6 Expressions régulières

5.6.1 Introduction

Les expressions régulières, aussi appelées expressions remplaçantes, ou encore **regexp**, servent à extraire parmi des chaînes de caractères celles qui correspondent à certains critères ou **motifs** (anglais: *pattern*). Par exemple, on peut effectuer une recherche dans un répertoire, dans un éditeur, etc.

Une chose à comprendre lors de l'utilisation d'expressions régulières dans le **shell** est que l'expansion des arguments (la recherche) est faite *avant* d'appeler la commande concernée.

Il faut éviter dès le départ de confondre les expressions régulières, utilisables dans des commandes comme `grep`, `sed` ou comme motifs de recherche dans un éditeur, des caractères utilisés comme **jokers** dans le **shell** : Ces **jokers**, que l'on retrouve sous beaucoup de systèmes d'exploitation, comme `*` ou encore `?`, forment un sous ensemble des possibilités des expressions régulières d'UNIX.

5.6.2 Syntaxe

Les caractères représentés dans un motif d'expression remplaçante devront être existants et identiquement placés par rapport aux autres caractères présents dans le motif pour que la partie de phrase soit considérée comme adéquate. Par exemple, en supposant que la phrase est `Le lièvre saute par-dessus la tortue`²⁵, les motifs suivants provoqueront un succès :

- sau
- par-dess
- e

Au contraire, les motifs suivants ne donneront pas de succès :

- saz
- pardess
- 34

²⁵Adaptation de *The Quick Brown Fox Jumps Over The Lazy Dog*, phrase test typique comprenant toutes les lettres de l'alphabet.

Certains caractères spéciaux permettent d'exprimer des requêtes plus complexes, comme par exemple : *donnez-moi toutes les phrases contenant abc puis n'importe quel caractère, puis def*. Pour ce faire, le **joker** ? est à employer. Notez que cela n'est valable que pour le **shell**. En général, pour une expression régulière, le caractère . joue ce rôle.

Des requêtes telles que *donnez-moi toutes les phrases contenant abcd, puis une suite de caractères qui me sont indifférents et dont le nombre m'est égal, puis def* sont exprimables à l'aide du caractère *. De nouveau, lors d'expressions régulières générales (non pas au niveau du shell), il faut faire précéder l'étoile du caractère que l'on désire voir multiplié un nombre quelconque de fois. Par exemple . * remplace n'importe quelle suite de caractères.

Ces deux caractères spéciaux sont nommés des **jokers** et sont connus dans beaucoup de systèmes d'exploitation. Il est intéressant d'en connaître quelques autres :

- \ Permet d'annuler l'effet spécial du caractère qui le suit. Cela peut être utilisé, par exemple, pour demander un point (alors qu'un point en général peut remplacer n'importe quel caractère.)
- ^ Permet de demander une comparaison forcée au début de la ligne. Notez que pour que ce caractère ait cette signification spéciale il doit débiter la ligne. Notez également qu'au début d'un ensemble, il inverse logiquement celui-ci (le "non" logique).
- [] Permet de donner un ensemble de valeurs possibles. Par exemple, pour la requête *donnez-moi une phrase commençant par abc, puis un seul chiffre (obligatoire), puis def et n'importe quoi*, on dirait ^abc[0-9]def.
- () Les parenthèses permettent de grouper logiquement une expression.

Les commandes **grep** et **egrep** permettent de rechercher des motifs dans un ou plusieurs fichiers. Voici un exemple (relativement complexe) d'utilisation des expressions régulières. Le but est d'afficher toutes les lignes d'un fichier indiqué en paramètre, sauf celles qui commencent par d2/local, d3/divers ou var/adm/tmp.

```
egrep -v "\Ad2/local|\Ad3/divers|\Avar/adm/tmp" fichier
```

Le symbole barre verticale, ou encore | permet dans ce cas de spécifier l'alternative, et l'argument -v de la commande **egrep** permet d'inverser la proposition.

Caractère(s)	Signification
.	caractère quelconque (sauf fin de ligne \n)
\$	fin de ligne (\n)
^	début de ligne sauf dans [] où il signifie "complément" (négation logique)
[]	un des caractères contenus dans les crochets
-	dans [] signifie de ... à ...
?	l'expression régulière précédente est optionnelle
*	répétition ≥ 0 du caractère précédent (éventuellement vide).
+	répétition > 0 (non vide)
	alternative entre deux expressions
\c	supprime la signification spéciale du caractère 'c'
()	groupe des expressions

TAB. 3 – Résumé des caractères spéciaux pour les **regexp**

Caractère(s)	Signification
[]	un des caractères contenus dans les crochets
-	dans [] signifie de ... à ...
^	dans [] signifie "complément" (NON)
?	remplace n'importe quel caractère
*	remplace n'importe quelle suite (éventuellement vide)
\c	supprime la signification spéciale du caractère 'c'

TAB. 4 – Résumé des caractères spéciaux pour le **shell**

5.7 Multi-niveau (Job control)

Les facilités de gestion des processus et des jobs du **shell** tcsh sont assez avancées, c'est pour cela que les utilisateurs novices ne devraient pas les utiliser, du moins pas dans un premier temps. De plus, l'interface screen est souvent bien mieux adaptée vu qu'elle maintient une fenêtre virtuelle par processus et permet de réattacher un ensemble de fenêtres préalablement détachées par commande explicite ou par une fin brutale de connexion (voir section 5.9.31 page 29).

Le système de **job control**, en français contrôle de travaux multiples, permet de lancer plusieurs travaux en parallèle, de les reprendre tour à tour comme travail de premier plan et de mettre un travail de premier plan en second plan. Il est également possible d'interrompre momentanément un processus, puis de le poursuivre ou de le détruire. Il est même possible de quitter le **shell** initial en laissant les processus tourner sans présence de l'utilisateur. Dans ce cas cependant il sera impossible de reprendre les sorties de ces processus ailleurs que dans des fichiers (il sera impossible de réattacher ces processus à un autre **shell**).

5.7.1 Concept de processus et de job

Un processus est un environnement d'exécution où, usuellement, un programme (commande, application, **shell**, ...) s'exécute. Cet environnement comprend des variables d'état et de choix. Du point de vue de l'utilisateur, plusieurs processus peuvent s'exécuter à un moment donné de façon concurrente. Ils peuvent aussi échanger des données. Par exemple,

```
ls -laR | wc -l
```

crée deux processus, un premier listant tous les fichiers de tous les sous-répertoires du répertoire courant, et un second recevant la liste du premier (via un **pipe**, une connexion entre la sortie d'une commande et l'entrée d'une autre (en remplacement du clavier), connexion représentée par une barre verticale) qui compte le nombre de lignes. Dans ce cas, même si pour l'utilisateur les processus monopolisent le terminal (il ne peut rien faire d'autre), il s'agit réellement de processus indépendants du **shell** initial. D'ailleurs on verra dans la section 5.7.3 comment rendre la main au **shell** initial en stoppant puis détachant en arrière-plan les processus.

Un job est un processus un peu particulier, en ce sens qu'il a des relations privilégiées avec son processus père, souvent un **shell**. De plus, un processus est identifié par un numéro unique dans tout le système, alors qu'un job se voit donner en plus un numéro de job unique pour le **shell** courant. On se référera en général dans les commandes à un processus par son simple numéro, et à un job par un pourcent suivi de son numéro, par exemple `kill %1` détruit le job numéro 1. Des valeurs spéciales sont définies, par exemple `kill %` tue le processus manipulé ou créé en dernier.

5.7.2 Obtenir la liste des processus

Il existe deux commandes principales, l'une montrant les processus, `ps`, et l'autre les jobs, `jobs`. Le concept de processus est plus général : un processus est identifié de façon unique dans le système par son numéro. Un job, au contraire, est identifié par un numéro qui est unique seulement dans le **shell** correspondant. Cela explique qu'on ne puisse employer **jobs** que pour les processus du **shell** courant. Les formats d'affichage de `ps` sont assez compliqués, et on ne retiendra que le numéro du processus, généralement indiqué comme PID, et le nom de la commande courante afin de retrouver des processus.

`ps` liste des processus du **shell** courant.

`ps -uuser` liste des processus appartenant à l'utilisateur `user`. Sous **BSD** ou **Linux** la commande `ps -x` est à utiliser.

`ps -ef` liste de tous les processus du système sous **POSIX** (p.ex. **Solaris**²⁶, ou **Linux** avec le tiret).

`ps -aux` liste de tous les processus du système sous **BSD**. Sous **Linux**, on peut également utiliser cette forme *sans le tiret*.

`jobs -l` liste des jobs avec leur numéro de job ainsi que leur numéro de processus système.

On notera aussi la commande `top`, qui permet de connaître les processus gourmands en puissance de calcul.

5.7.3 Suspendre un processus

Si le processus à suspendre est celui qui monopolise le clavier, il suffit de taper `CTRL-Z`. Sinon, on entre `stop %1` si le job à stopper porte le numéro 1. On peut aussi entrer `stop` suivi du numéro (PID) du processus à stopper. On peut également stopper des processus n'appartenant pas au **shell** courant.

5.7.4 Reprendre l'exécution d'un processus

Avec `fg` et `bg` on peut reprendre l'exécution, respectivement en **premier plan** (anglais: *foreground*), donc monopolisant le clavier, ou en **arrière-plan** (anglais: *background*), d'un job en indiquant son numéro de job. Certains jobs interactifs (éditeurs par exemples) ne peuvent pas être mis en arrière-plan sans être suspendus automatiquement à nouveau. On ne peut pas reprendre l'exécution d'un job non rattaché au **shell** courant.

Pour reprendre l'exécution du dernier processus suspendu ou pour le reprendre au premier plan s'il est en arrière-plan, il suffit d'entrer au **shell** `%`.

5.7.5 Détruire un processus

Pour ce faire, on utilise la commande `kill` (voir section 5.9.18 page 27) comme dans les exemples suivants (on suppose que le processus 5643 nous appartient, et qu'un job numéro 3 existe). Pour connaître ces numéros, il suffit de consulter la section 5.7.2.

`kill 5643` Tentative de destruction du processus 5643, celui-ci peut refuser (par exemple il peut ignorer la demande)

²⁶Certains systèmes comme **Solaris** ou **OSF/1** disposent de deux commandes `ps` différentes (`/usr/ucb/ps` et `/usr/bin/ps`).

`kill -9 5643` Destruction du processus 5643 sans lui demander son avis

`kill %3` Tentative de destruction du job 3

`kill -9 %3` Destruction du job 3

`killall -TERM apache` Envoi du signal de terminaison à tous les processus Apache.

5.7.6 Lancer un processus en arrière-plan

On peut mettre en arrière-plan un processus à son lancement, on appelle cela également **détacher** un processus, en terminant la commande par un `&`. Par exemple, la commande suivante affiche la date en parallèle du **shell** courant. Noter que cela n'est pas vraiment utile dans cet exemple.

```
date &
```

5.8 Syntaxe

La syntaxe générale des commandes est la suivante :

```
commande -options paramètre1 ... paramètren
```

Il est important de remarquer qu'un espace sépare chacun des paramètres et les paramètres de la commande elle-même. On remarquera également que certains paramètres peuvent commencer par un tiret, noté `-`. En général, ces paramètres spéciaux (options) sont à spécifier comme premier paramètre.

Originellement, seule des options à un caractère étaient spécifiés, ce qui rendait possible leur combinaison : p.ex. `ls -l -a -R` est aussi exprimable comme `ls -laR`. Le projet **GNU** définit également des options longues, que l'on précède de deux tirets : comme par exemple `ls --inode`. Ces options ne sont bien entendu pas combinables en une seule.

Malheureusement, certains logiciels ne respectent pas ces conventions et proposent des options à un tiret mais plusieurs caractères.

Une forme permettant de spécifier un argument à un paramètre est par exemple : `--option=valeur` ou encore `--option valeur`.

5.9 Principales commandes

Il faut différencier les **commandes internes** du shell des commandes externes (UNIX) : la commande `type` permet de déterminer si une commande est interne ou non. Parmi les commandes internes, citons `cd` (qui ne pourrait fonctionner en commande externe : les commandes externes sont exécutées dans un sous-processus, qui est détruit en fin d'exécution, y compris le changement de répertoire éventuel).

Quelques-unes des commandes disponibles seront expliquées, dans leurs aspects les plus simples. Les commandes de messagerie et de communication sont décrites en deuxième partie de ce guide en page 65. Pour une aide plus complète, utilisez la commande `man`, comme expliqué ci-après (*voir section 5.9.20 page 28*).

5.9.1 `ls`, liste d'un répertoire

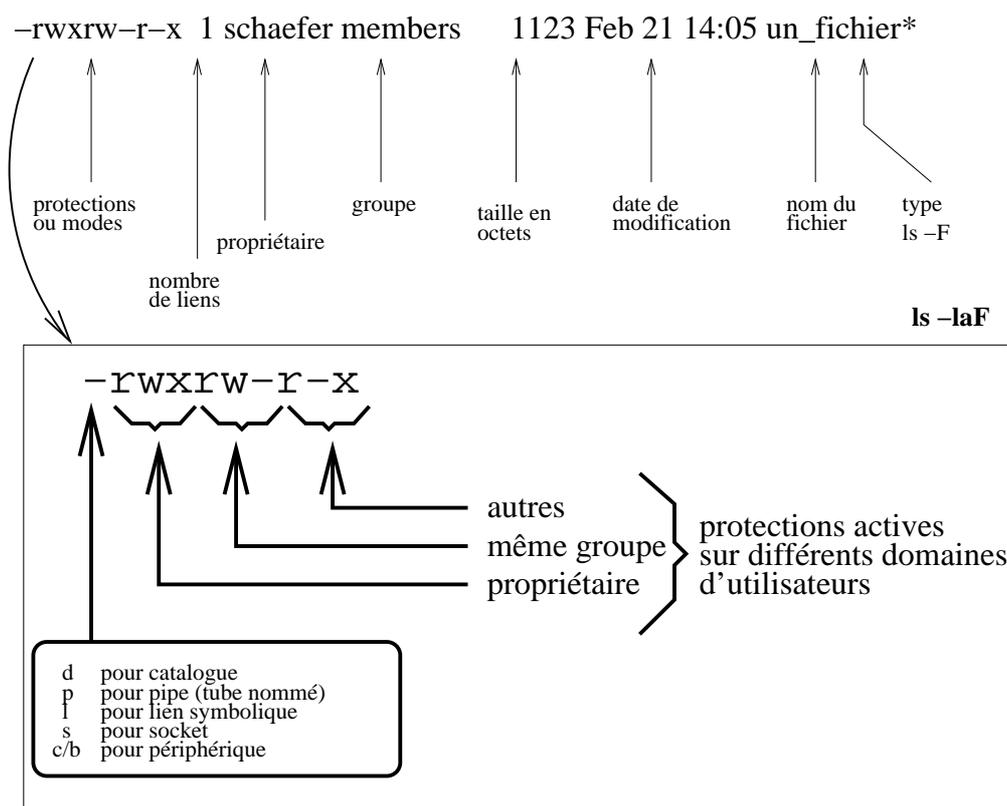
Elle permet de lister le contenu d'un répertoire. Beaucoup d'options sont disponibles, parmi elles, certaines sont intéressantes :

`ls -lag | more` lister tous les fichiers de façon complète avec les modes, les groupes, les créateurs et les dates, ainsi que les fichiers spéciaux commençant par `.` et faire afficher par la commande `more`. Il s'agit d'un exemple d'utilisation des pipes.

`ls -F` lister les fichiers avec des indications de type de fichier sous la forme d'un caractère final (fichier, répertoire/, lien@, exécutable*).

La figure 3 page suivante détaille la sortie de la commande `ls -laF`²⁷.

²⁷`ls -aF` est équivalent, mais est plus amusant, d'après cagnard :-).

Figure 3: Détail d'une sortie de `ls -laF`

5.9.2 a2ps, mp, enscript, mise en page de listings

Ces commandes permettent de transformer un texte, usuellement un listing, en un document **PostScript** pour une meilleure mise en page. Il existe de nombreuses options, les plus classiques transforment un fichier texte en un fichier **PostScript** comportant deux colonnes, en mode paysage (anglais: *landscape*), soit tourné de 90 degrés. La sortie est ensuite stockée dans le fichier `listing.ps` (ou directement à l'imprimante avec `a2ps` si on n'utilise pas l'option `-nP`).

```
enscript -plisting.ps -2rh programme.c
a2ps -nP programme.c > plisting.ps
```

5.9.3 alias, définition de synonymes de commandes

Elle permet la création de raccourcis ou de synonymes de commandes qui existent par exemple sur d'autres systèmes.

`alias dir ls -lagF` la commande `dir` est un synonyme pour `ls -lagF`

5.9.4 awk, interprète de scripts

Cet interprète (on dit aussi abusivement interpréteur) est surtout utilisé pour la conversion de formats textuels (tables, tableaux) et pour l'extraction de données. Par exemple, on peut extraire la troisième colonne d'un tableau et sommer les deux dernières colonnes:

```
awk '{print $3 " " " ($NF-1) + $NF}'
```

Ce script utilise le concept de séparateurs de `awk`. `NF` représente le nombre de champs (anglais: *field number*). Le caractère `$` précède toute expansion de variable (comme dans un **shell**). Notez la présence d'apostrophes afin d'éviter l'expansion des variables directement par le shell.

5.9.5 cat, affichage ininterrompu d'un fichier

Permet d'afficher sans interruption possible un fichier dont le nom est communiqué en paramètre. La commande `more` (voir section 5.9.22 page 28) permet quant à elle un affichage page par page. Une utilisation intéressante de cette commande est pour concaténer des fichiers. Par exemple, `cat a b c d > tous` crée un fichier `tous` contenant la concaténation (mise bout à bout) des fichiers `a` à `d`, grâce à une **redirection**.

5.9.6 cd, changement de répertoire

La commande `cd` permet de se déplacer dans l'arborescence des répertoires. Il y a trois cas intéressants :

`cd ..` revenir au répertoire précédent dans l'arborescence.

`cd /` revenir à la racine.

`cd` revenir au répertoire par défaut de l'utilisateur, `~` ou `$HOME`.

5.9.7 chmod, changement des protections d'un fichier

Elle permet de modifier les attributs, ou protections, d'un fichier ou d'un répertoire, seulement si vous en êtes le créateur. Elle permet également de protéger vos données (voir section 9.3.2 page 51).

Ses différentes formes, en particulier avec les versions GNU, sont :

`chmod qui±permission objet ... objet` Ajouter ou retirer des permissions, pour le propriétaire, groupe ou autres suivant la valeur de `qui`.

`chmod qui=permission objet ... objet` Décider de permissions, appliquées au propriétaire, groupe ou autres suivant la valeur de `qui`.

`chmod permission_octale objet ... objet` Décider de toutes les permissions simultanément, en mode octal.

Par exemple :

`chmod u+r .` autorise vous-même à lister le répertoire courant (`.`)

`chmod g+w bin` autorise les gens de votre groupe à écrire dans `bin`. Si `bin` est un répertoire, l'accès en création et effacement de fichiers²⁸ est autorisé.

`chmod o+x bin` autorise n'importe qui à entrer dans le répertoire `bin`, ou à exécuter `bin` si `bin` est un fichier.

`chmod a+rx .` ajoute les permissions lecture et accès au répertoire courant pour le propriétaire, le groupe et les autres.

`chmod 664 fichier` autorise lecture et écriture pour le propriétaire et le groupe, et lecture seulement pour les autres.

Notez que la création de répertoires ou de fichiers se fait avec un mode par défaut qui dépend de la commande `umask` (voir section 8.1.2 page 46). Le paramètre de mode de la commande `chmod` peut être également exprimé sous forme absolue en numérique, en exprimant les protections utilisateur, groupe et autres de gauche à droite par trois²⁹ valeurs de 0 à 7. Sachant que l'exécution³⁰ l'écriture et la lecture sont symbolisées respectivement par les valeurs 1, 2 et 4, on a, par exemple :

`chmod 664 fichier` comme $6 = 2 + 4$, on autorise l'accès en écriture et en lecture pour nous et notre groupe, tandis que seule la lecture est possible pour les autres utilisateurs qui ne sont pas membres du groupe.

Une représentation graphique des trois groupes de permission est à la figure 3 page précédente.

5.9.8 crontab, opérations à heures fixes

La commande `crontab` permet d'effectuer des opérations à heures fixes, comme par exemple compresser des fichiers, mettre à jour des informations, envoyer des messages. Son utilisation est très complexe et son utilité très minime pour des utilisateurs non expérimentés³¹.

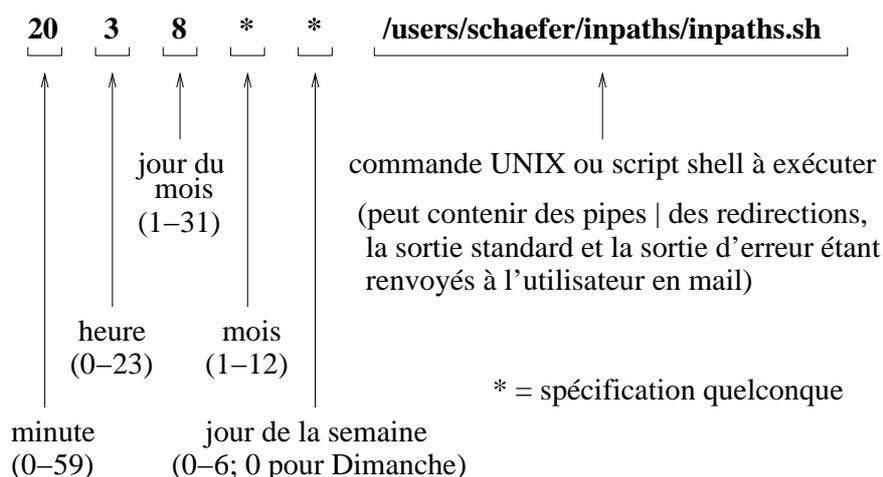
On configure cette commande en lui donnant en entrée standard un fichier de configuration contenant les commandes à exécuter ainsi que les spécifications temporelles (quand exécuter). Il est conseillé d'éditer un fichier que l'on nomme par exemple `crontab`, afin d'éviter de perdre sa configuration en cas de mauvaise manipulation. Ensuite, il suffit de taper la commande `crontab < crontab` pour changer la configuration. Une alternative est de lancer l'éditeur par défaut (`$EDITOR`) avec `crontab -e`. On peut entrer `crontab -l` pour connaître la configuration courante. Le format de chacune des lignes peut sembler compliqué et est détaillé à la figure 4 page suivante.

²⁸Y compris ceux qui ne vous appartiennent pas (les protections des fichiers eux-mêmes n'entrent pas en ligne de compte). Pour autoriser la création sans l'effacement des fichiers des autres, utilisez le **sticky bit**, `chmod +t bin`.

²⁹En fait, il existe un quatrième chiffre initial sous-entendu qui vaut en général zéro. Autrement, il est possible de spécifier des extensions de mode permettant des opérations spéciales. Parmi ces valeurs, citons le **sticky bit** (permettant des répertoires partagés comme `/tmp`) et les bits de changement d'utilisateur et de groupe (anglais: *set user/group id on execution*), utilisés par exemple par la commande `passwd` pour accéder au fichier des mots de passe. Le manuel UNIX de la commande `chmod` donne plus de détails à ce sujet (voir section 5.9.20 page 28).

³⁰La recherche de fichiers, pour un répertoire. Le listage d'un répertoire est possible avec le bit de lecture.

³¹Certains anciens systèmes UNIX ne supportent d'ailleurs pas des configurations par utilisateur. Dans ce cas seul l'administrateur peut installer des travaux à heure fixe. On peut parfois utiliser `at` à la place.

FIG. 4 – Détail d'une entrée **crontab**

La configuration décrite lance chaque 8 du mois à 3 heures 20 minutes du matin un **script**.

Il faut savoir que souvent le **shell** utilisé pour lancer les commandes spécifiées est **sh**. On peut spécifier `#!/bin/csh` par exemple au début d'un script dont le nom est donné en paramètre pour changer d'interprète. Il ne faut pas oublier de spécifier les sentiers (chemins de recherche, **PATH**) complets : l'environnement n'est probablement pas initialisé parfaitement. Les versions récentes de **cron** supportent la définition de variables au début de la crontab.

Une commande similaire, beaucoup plus simple à utiliser, et qui permet cette fois de retarder l'exécution de travaux (par exemple pendant la nuit ou le week-end) est la commande **at**.

```
at now + 5 min < fichier_commande
```

5.9.9 **mv**, déplacer/renommer des fichiers

Cette commande permet de renommer ou de déplacer un fichier sur le même système de fichiers. Les versions récentes permettent également de déplacer (en les copiant) des fichiers entre deux systèmes de fichiers. Dans ce dernier cas l'opération n'est pas **atomique**.

5.9.10 **df**, disponibilité des systèmes de fichiers

Celle-ci permet de connaître les systèmes de fichiers accessibles, et, pour chacun : son point de montage dans l'arborescence, sa capacité et son taux d'occupation. On peut spécifier un répertoire en argument, par exemple `df ~` permet de connaître l'espace disponible sur le disque où se trouve le compte de l'utilisateur. En général, un pourcentage (de 5 à 10 %, configurable) de chaque système de fichier est réservé à **root**. Il est donc possible que le total ne corresponde pas.

L'option `-i` permet de demander non pas l'espace disque mais le nombre de fichiers préalloués et disponibles dans le système de fichiers : en général, les systèmes de fichiers **POSIX** sont basés sur le concept d'**inode** préalloués : une fois la valeur maximum atteinte, le système de fichiers ne peut plus créer de nouveaux fichiers.

5.9.11 **diff**, affichage des différences entre fichiers

Cette commande affiche les différences entre deux fichiers. Cela est parfois utilisé par des utilitaires stockant des modifications incrémentales dans des sources ou textes sous leur contrôle, permettant de retrouver des versions antérieures ou le travail à plusieurs, comme avec **RCS** ou **SCCS**. La commande **patch** permet de retrouver le fichier modifié à l'aide du résultat de **diff** et du texte original.

5.9.12 **du**, usage disque

Cette commande permet de connaître l'espace disque utilisé par un sous-répertoire. L'unité dépend du système (parfois 512 octets, parfois 1 Koctet, ou d'autres valeurs). Un paramètre très souvent utilisé est `-s *`, qui permet d'obtenir la taille de tous les objets du répertoire courant (fichiers et répertoires).

5.9.13 **emacs**, éditeur pleine page multi-fonctions

Emacs est un éditeur **VT100** offrant des possibilités de macros et de scripts inégalées. Une description sommaire de cet éditeur se trouve dans la section 6.

uemacs, éditeur pleine page simplifié Micro-emacs est un sous-ensemble de l'éditeur **Emacs**.

ne, éditeur pleine page avec menus déroulants Nice-Editor est un éditeur ressemblant à **Emacs**, à part une caractéristique intéressante : il permet l'affichage de menus déroulants sur un terminal **VT100** ou similaire par simple pression de la touche **PF1** ou équivalente.

5.9.14 find, recherche de fichiers

Cette commande permet de trouver des fichiers depuis une racine spécifiée. Par exemple, la commande suivante affichera tous les fichiers ayant un **t** dans leur nom, et ceci dans tous les sous-répertoires du répertoire courant, y compris ce dernier :

```
find . -name '*t*' -print
```

Notez les guillemets qui empêchent une interprétation de la **regex** par le **shell**³².

Cette commande est très puissante. On peut lui faire rechercher des fichiers en fonction d'un utilisateur ou d'un groupe particulier, en connaissant une partie du nom, en fonction de sa taille, de sa date de création ou de dernière modification ou de son **mode** (les protections UNIX, voir **chmod**). On peut également exécuter une commande (ce qui sera souvent peu rapide mais fort utile) sur les fichiers trouvés.

Voici un exemple un peu plus complexe qui recherche depuis la racine tous les fichiers **core** appartenant à l'utilisateur **schaefer** et qui sont plus gros que 500 kilo-octets ou modifiés il y a plus de 7 jours, sauf s'ils appartiennent au groupe **keep**. Les fichiers trouvés sont automatiquement effacés. Notez l'usage des parenthèses (précédées d'un **backslash** pour éviter leur interprétation par le **shell**). Remarquez également la notation utilisée pour le *ou* et le fait que la conditionnelle *et* s'exprime par simple juxtaposition. Enfin admirez la syntaxe très particulière de l'exécution d'une commande.

```
find / -name 'core' -user schaefer \( -size +500k -o -mtime +7 \) \! -group keep -exec rm {} \;
```

5.9.15 grep, recherche de motif dans un fichier

grep est une commande permettant de trouver rapidement et facilement des motifs (expressions régulières) dans un ou plusieurs fichiers. Typiquement, pour trouver la chaîne **Bonjour** (différente de la chaîne **bonjour** si l'argument **-i** n'est pas spécifié) dans les fichiers du répertoire courant, on écrira

```
grep Bonjour *
```

Il existe plusieurs variantes à cette commande, dont **egrep**, qui permet de spécifier des conditions plus générales. L'option **-i** permet d'ignorer la casse, ce qui est parfois bien pratique.

Le nom **grep** vient de *global regular expression print* ou *g/re/p* une commande de l'éditeur ligne **ed** qui a été transformée en un programme externe.

5.9.16 gzip, compression et décompression de fichiers

Cette commande permet la compression de fichiers textuels ou binaires, ainsi que la décompression d'archives **gzip** et **compress**. On reconnaît les archives **gzip** par leur extension **.gz**, et les archives **compress** par leur extension **.Z**.

```
gzip fichier compresse fichier en fichier.gz
```

```
gzip -d fichier.gz décompresse fichier.gz en fichier
```

Notez que l'on peut aussi spécifier **gzip -d fichier** pour décompresser un fichier avec extension **.gz** pour autant que **fichier** n'existe pas déjà. La forme **gzip -d fichier.gz** est cependant plus indiquée. Certains systèmes offrent une commande **gunzip** à la place.

5.9.17 head, affichage des premières lignes d'un fichier

Avec cette commande on affiche les premières lignes d'un fichier.

5.9.18 kill, destruction d'un processus ou communication de signaux

Cette commande permet de détruire un processus ou de lui transmettre des signaux (conditions exceptionnelles, comme par exemple le raccrochage du modem, un délai dépassé, etc.). Si le processus cible est le numéro 5346,

```
kill 5346 Tente de détruire le processus 5346
```

```
kill -9 5346 Force la destruction du processus 5346
```

Un aspect intéressant de cette commande est la possibilité de détruire un groupe de processus. Par exemple, pour détruire tous les processus vous appartenant, sauf le **shell** courant, il suffit de donner les commandes suivantes :

```
kill -15 -1 Tente de détruire tous vos processus
```

```
kill -9 -1 Force la destruction de tous vos processus
```

Pour obtenir la liste des signaux, utiliser **kill -l**.

³²Rappelez-vous: en général l'expansion des expressions remplaçantes se fait au niveau du shell, *avant* d'appeler le programme correspondant.

5.9.19 `lpspr`, mise en page de documents PostScript

Certains systèmes UNIX (principalement **OSF/1** et **Solaris** avec les imprimantes **DEC**) offrent une commande permettant d'imprimer des documents en format **PostScript** recto-verso (option `-K2`) ou plusieurs pages par côté (par exemple `-N2` pour deux pages par côté). Il suffit ensuite de transmettre via un **pipe** à la commande d'impression proprement dite, soit

```
lpspr -K2 -N2 | lpr
```

Des variantes libres de cette commande existent, parmi elles citons `pstops` et `psnup`.

5.9.20 `man`, manuels en ligne

La commande `man` permet d'obtenir des informations sur des commandes. Dans une première approche, il suffit de taper `man foo` si `foo` est la commande à expliciter. Les manuels systèmes sont organisés en sections et utilisent un format standard³³ (commandes utilisateurs 1, commandes administrateur 8, appels systèmes 2, bibliothèques de fonctions (p.ex. C standard) 3, fichiers spéciaux (p.ex. dans `/dev`) 4, formats de fichiers 5, jeux 6, macros et conventions 7. Par exemple, pour obtenir des informations sur la commande `ls`³⁴, on peut écrire `man 1 ls`³⁵ ou encore `man ls` ce qui est moins précis (toutes les pages disponibles seront présentées, une par une). Une option intéressante est `-k` qui permet d'obtenir les noms des différents manuels contenant le mot-clé passé en paramètre. Ainsi si l'on désire connaître la liste des pages de manuel qui parlent de `tcp`, on entrera `man -k tcp`.

5.9.21 `mkdir`, création d'un répertoire

Crée des répertoires. L'option `-p` permet de créer automatiquement toute une hiérarchie. On peut donner comme argument un chemin complet à créer. Les répertoires déjà existants dans ce chemin seront ignorés.

5.9.22 `more`, affichage page par page d'un fichier

Permet d'afficher un fichier avec arrêt à chaque page, ainsi que d'autres fonctions spéciales. On passe d'une page à l'autre avec les touches `ESPACE` (page suivante) et `b` (page précédente). On quitte avec `q` et on demande de l'aide avec `h`. Notez que certaines versions de cette commande (notamment la variante `less`) ont besoin d'un deuxième `q` pour sortir du programme.

5.9.23 `nice`, changement de priorité

La commande `nice` permet de diminuer la priorité d'un processus à son lancement afin de ne pas gêner les autres processus (it's nice, isn't it?).

```
nice commande
```

Pour voir le niveau de priorité d'un processus ainsi que son niveau de "gentillesse", on peut utiliser la commande `ps -l`. Plus la valeur est importante, moins la priorité est forte.

La commande `renice` permet de diminuer la priorité d'un processus actif (vous appartenant). Augmenter la priorité n'est en général possible que pour `root`.

5.9.24 `nroff`, mise en page de manuels

Il arrive souvent que les documentations soient disponibles sous une forme non facilement visualisable : on le remarque à l'extension numérique du fichier (p.ex. `foo.1`). On peut les transformer en un texte formaté à l'aide de cette commande. Pour les manuels, on utilise un **macro** spécifique : `nroff -man foo.1 > foo.txt`. Cette commande produit un fichier `foo.txt` qui peut ensuite être affiché page par page avec `more`. On pourrait aussi utiliser un **pipe** à la place de la redirection.

5.9.25 `passwd`, changement du mot de passe

Cette commande permet de changer de mot de passe (voir section 9.2 page 50). Certaines restrictions seront imposées par le système quant à la longueur du mot de passe et à son contenu pour diminuer les risques de piratage. L'ancien mot de passe est toujours demandé. Il n'y a pas d'écho pendant les demandes de mot de passe.

5.9.26 `pico`, éditeur pleine page simplifié

Cet éditeur est très simple à utiliser et malgré cela relativement puissant. Il est disponible en standard sous environnement **PINE** (cet environnement permet de lire le **mail** et les **news** de façon similaire et intégrée sous interface **VT100**), ou séparément.

³³Certains systèmes classent également les manuels par variante de système d'exploitation. On précise alors la recherche par une option du style `-systype os_type`. Le défaut est de montrer tous les manuels.

³⁴Les documentations UNIX représentent souvent une commande suivie de sa section de manuel entre parenthèses comme dans `ls(1)`.

³⁵Solaris introduit l'argument `-s` avant le numéro de section.

5.9.27 `pwd`, affichage du répertoire courant

Elle permet d'afficher le répertoire courant.

5.9.28 `quota`, information sur l'utilisation disque autorisée

Lorsque l'espace disque devient limité, il se peut que l'administrateur décide d'attribuer un quota à tous les utilisateurs, n'autorisant qu'un certain nombre de fichiers et un volume maximum. Pour connaître votre quota actuel, il suffit de taper `quota -v`. Les valeurs sont en kilooctets, donc par exemple 10000 signifie 10 megaoctets. Si au **login** (à votre prochaine connexion) le système vous indique quelque chose ressemblant à `QUOTA EXCEEDED`, vous devez supprimer des fichiers sinon vous risquez des pertes de données car toute tentative de création de nouveaux fichiers ou d'extension de fichiers existants se soldera par un échec. Ce message peut également s'afficher au cours d'une session.

5.9.29 `rm`, destruction d'un fichier

Permet de détruire un fichier. `rmdir` permet quant à elle de détruire un répertoire préalablement vidé. Certaines options de ces commandes sont pratiques mais dangereuses, et ne sont donc pas décrites ici, à part les options suivante :

`rm -i fichier1 ... fichiern` permet de détruire tous les fichiers indiqués avec confirmation pour chacun. On appelle cela le mode interactif de **rm**. Certains ajoutent un **alias** dans leur `.cshrc` afin de remplacer `rm` par `rm -i`. Cela peut parfois s'avérer plus dangereux qu'utile : le risque est grand alors de prendre l'habitude de confirmer sans vraiment lire. Une erreur à ne pas faire est de mettre l'option `-i` ailleurs qu'au début de la commande : le système n'en tiendrait pas compte et détruirait sans confirmation.

`rm -r repertoire1 ... repertoiren` permet de supprimer récursivement une arborescence (répertoires, sous-répertoires et fichiers). Inutile d'insister sur le danger d'une telle commande.

Il faut donc noter que par défaut, la commande `rm` ne demande aucune confirmation. Les fichiers sont donc irrémédiablement perdus, à moins que le fichier ait été créé le jour précédent, au moins, dans ce cas, il existe sur les copies automatiques journalières de sécurité (vérifiez cependant avec votre administrateur système si des copies de vos données sont effectuées régulièrement). Il faut noter que vouloir récupérer un fichier préalablement effacé à l'aide des structures de répertoires est une opération la plupart du temps vouée à l'échec, en particulier sous UNIX en raison du caractère multi-utilisateur et de la création fréquente de fichiers. Selon les caractères présents dans le nom de fichier, il peut être nécessaire d'utiliser le `\` ou la complétion de la ligne de commande. Si le premier caractère du nom de fichier est `-`, utilisez `rm ./-nom`.

5.9.30 `rz`, réception d'un fichier via ZModem

On tape simplement `rz` et on configure son programme de terminal pour l'émission d'un fichier.

5.9.31 `screen`, gestionnaire de fenêtres virtuelles

Cet utilitaire permet d'ouvrir plusieurs fenêtres virtuelles sur un seul terminal. On passe d'un écran à l'autre avec des séquences de contrôle. Il est possible, avec un simple terminal **VT100** d'ouvrir jusqu'à dix fenêtres indépendantes. Des fonctions spéciales, comme la sauvegarde des entrées sorties, sont possibles. On commence une session **screen** en tapant `screen` ou `screen -r`. La deuxième forme est employée quand on désire continuer (anglais: *to resume*) une session détachée préalablement.

Le mode de contrôle s'obtient avec `CTRL-A`, puis un caractère de commande. Taper un chiffre dénote la fenêtre (de 0 à 9). Par exemple (en supposant que `CTRL-A` a déjà été pressée). Attention, les commandes sont sensibles à la casse.

? Aide

c créer une nouvelle fenêtre.

k détruire de façon sauvage la fenêtre courante. Il est mieux de fermer le shell avec `exit` ou `CTRL-D`.

0 aller sur la fenêtre 0.

H sauver/ne pas sauver la session en cours dans cette fenêtre dans un fichier `screenlog.x`, où x est le numéro de la fenêtre.

h copie d'écran dans fichier `hardcopy.x`.

w liste des fenêtres.

m contrôle d'activité allumé/éteint.

5.9.32 `sed`, éditeur de flots

Cet interprète permet de gérer des flots de données en exécutant des opérations de substitution. Par exemple, pour remplacer les noms de **forums** de la notation pointée (**rec.humor.funny**) en la notation UNIX de stockage interne (`rec/humor/funny`), on pourrait utiliser cette commande :

```
sed 'y%.%/%' < entree > sortie
```

Ce script demande de remplacer les occurrences de points par des barres obliques (anglais: *slash*). Les pourcents sont des caractères délimiteurs qui peuvent être quelconques tant qu'ils ne figurent pas ni dans le motif ni dans la chaîne remplaçante. Une description extensive de ce logiciel peut être trouvée dans [27].

5.9.33 stty, changement de modes du terminal

Cette commande est très complexe mais également très puissante. Elle permet de redéfinir des paramètres du terminal, comme par exemple la touche d'interruption `intr` (défaut : `CTRL-C`), ou la touche d'effacement `erase` (défaut : `BACKSPACE`) (voir section 8 page 45). Dans le cas où la touche `BACKSPACE` n'a pas le comportement voulu il suffit donc de taper `stty erase BACKSPACE`.

5.9.34 sz, envoi d'un fichier via ZModem

Elle permet d'envoyer des fichiers via le protocole **ZModem**, en général à travers un **modem**. On indique simplement le(s) nom(s) du (des) fichier(s) à envoyer sur la ligne de commande. Si votre programme de terminal ne supporte pas ZModem, vous pouvez essayer **XModem** ou un protocole similaire. En dernier recours, la commande **kermit** permet d'utiliser le protocole kermit. Elle existe sur la plupart des systèmes UNIX. Dans certains cas, il faut renommer le fichier avant l'envoi avec **mv**, pour que le nom de fichier soit compatible avec le système de fichiers distant.

5.9.35 tail, affichage des dernières lignes d'un fichier

On peut afficher les dernières lignes d'un fichier ou afficher en continu les ajouts à un fichier, ce qui peut être très utile pour surveiller un fichier de **log** (anglais: *logfile*), respectivement :

```
tail nom_fichier
tail -f nom_fichier
```

Dans ce dernier cas, on interrompt la commande avec `CTRL-C`.

5.9.36 tar, gestion d'archives portables

Cette commande permet de rassembler plusieurs fichiers en une seule archive. Il n'y a cependant aucune compression. On peut ensuite compresser le fichier résultant avec **gzip** par exemple.

```
tar cvf - . | gzip -9 > truc.tar.gz
```

archive tous les fichiers du répertoire courant et de ses sous-répertoires, compresse cette archive avec **gzip** (l'argument -9 sélectionne la meilleure (mais plus lente) compression) et stocke le résultat dans `truc.tar.gz`. La version **GNU** de cette commande permet de simplifier en `tar cvfz truc.tar.gz .`, sans toutefois pouvoir spécifier la force de la compression.

```
tar tvf truc.tar
```

liste le contenu d'un fichier **tar** `truc.tar`.

```
gzip -d < truc.tar.gz | tar xvf -
```

extrait le contenu d'un fichier **tar** compressé avec **gzip**. De nouveau, la version **GNU** est plus puissante et incorpore ce mécanisme: `tar xvfz truc.tar.gz`.

Note : le `-` dans la commande `tar` désigne la sortie ou l'entrée standard au lieu d'un fichier. Il faut aussi signaler qu'une commande similaire, parfois plus puissante pour certaines applications, est **cpio**. Notez toutefois que son utilisation est légèrement plus complexe.

5.9.37 uuencode, codage de fichiers binaires pour transmission via courrier électronique

Grâce à cette commande, on peut envoyer des fichiers binaires par courrier électronique (voir section 22.7 page 78). Ils sont reconnaissables très facilement par leur en-tête `begin 644 nom_de_fichier`. On les décode à l'aide de `uudecode`.

```
uuencode < fichier fichier > fichier.uu
```

crée le fichier uuencodé `fichier.uu` qui pourra être transmis par **mail**.

```
uudecode fichier.uu
```

décode le fichier uuencodé après réception et sauvegarde le résultat dans un nouveau fichier dont le nom est pris de la ligne `begin` du fichier `fichier.uu`.

5.9.38 vi, éditeur pleine page

Cette commande permet d'éditer des fichiers de façon plein-écran. Cet éditeur peut sembler compliqué à utiliser, mais il est très puissant, et de plus a l'avantage de fonctionner sur tous les systèmes UNIX quelle que soit leur taille et leur niveau d'installation. Il possède trois modes :

déplacement C'est le mode par défaut. Il permet le déplacement à l'aide des touches du curseur sur n'importe quel caractère de l'écran. On le quitte généralement en tapant `i` (insertion) ou `a` (ajout) pour le mode d'insertion, et en tapant `:` ou `/` pour le mode de commande.

commande Il permet d'entrer des commandes similaires à celles de **ed**, l'éditeur par ligne. Il est par exemple possible de substituer des caractères dans tout le fichier. On quitte ce mode avec `RETURN`.

insertion Ce mode permet d'entrer des caractères. Pour le quitter, il faut taper `ESCAPE`

Mode de déplacement Il est possible de se déplacer avec les touches du curseur, la touche espace, la touche *RETURN* et la touche *BACKSPACE*. D'autres fonctions sont disponibles :

CTRL-G information sur le fichier courant et la position dans ce fichier

i passage en mode insertion (avant le curseur)

a passage mode ajout/insertion (après le curseur)

d puis d destruction d'une ligne, mise dans le tampon

J concaténation de deux lignes

x destruction du caractère courant

\$ se positionner à la fin de la ligne

0 se positionner au début de la ligne

& répéter la substitution précédente

p ajouter à l'endroit du curseur le contenu du tampon, (*P* insère avant le curseur)

D détruire jusqu'à la fin de la ligne, mise dans le tampon

Y mettre la ligne courante dans le tampon

CTRL-U Une demi-page plus haut

CTRL-D Une demi-page plus bas

CTRL-B Une page plus haut

CTRL-F Une page plus bas

CTRL-L Rafraîchir l'écran

. Répéter la commande précédente

On notera que l'on peut précéder ces commandes par un nombre (par exemple *32Y* mettra dans le tampon les 32 lignes suivantes, que l'on pourra insérer avec *p*).

Mode de commande Les commandes, qui doivent toutes être terminées par un *RETURN* et commencer par un *:*, sont très nombreuses. La plupart acceptent un argument numérique qui permet de répéter *n* fois l'opération (par exemple *5d* détruit cinq lignes depuis la position courante). En voici un échantillon :

j concaténation de lignes

d destruction de lignes

x sauver et quitter. Attention, cette commande est dangereuse car très similaire à une commande de chiffrement (celle-ci en majuscule). De plus, elle est parfois spécialement implémentée si on force la sortie avec le point d'exclamation (certaines versions ne sauvegardent pas dans ce cas). Il est recommandé d'utiliser plutôt la combinaison *wq*.

w sauver

q quitter

n passer au fichier suivant si plusieurs fichiers

\$ se déplacer à la fin du fichier

set ic ignorer la casse en recherche. Annuler avec *set noic*

set nu numérotter les lignes. Annuler avec *set nonu*

N'oubliez pas que dans le mode de commande, toute commande commence par : et finit par un RETURN.

Si l'éditeur refuse une opération (par exemple il refusera la commande *q* si le texte courant n'est pas sauvé), on peut forcer l'exécution par un point d'exclamation ajouté (par exemple *q!*). De plus, en tapant directement le numéro d'une ligne, on peut s'y déplacer. Cela est également possible d'ailleurs en lançant l'éditeur avec un argument commençant par *+* suivi d'un numéro de ligne.

Voici quelques exemples de commandes complexes (il faut toujours ajouter un *:* avant et un *RETURN* après). On notera que *\$* représente la dernière ligne et *.* la ligne courante dans les spécifications d'adresse de ligne.

1, \$s/foo/bar/g Remplace toutes les occurrences de *foo* par *bar*.

%s/foo/bar/g Equivalent.

., \$s/foo/bar/g Remplace toutes les occurrences de *foo* par *bar*, depuis la ligne courante.

3, 7s%foo/bar%bar/foo%g Remplace aux lignes 3 à 7 les occurrences de la sous-chaîne *foo/bar* par *bar/foo*. Cela montre que l'on peut utiliser un autre caractère séparateur que */*.

1, \$s/foo/bar/c Remplace, après confirmation, chaque occurrence de *foo* par *bar*: répondre par *y* pour oui et *n* pour non.

1, \$s/CTRL-VCTRL-M//g Supprimer les retours de ligne indésirables: conversion MS-DOS vers UNIX.

Notez que le remplacement est séparé par un caractère, ici /, mais qui peut être quelconque. Il faut également faire attention car certains caractères, notamment ., *, [et] ont des significations particulières, il faut les précéder du caractère d'échappement \ (voir section 5.6 page 20). Ce dernier doit être doublé s'il doit être utilisé tel quel. Voyez également la signification de CTRL-V qui permet d'entrer un caractère de contrôle.

Certaines opérations se font sans : ni RETURN, celles-ci sont donc exécutées en mode déplacement.

Noter que pour rechercher des chaînes, on doit utiliser la construction /chaîne puis RETURN, ceci sans : initial. Voyez l'état ic décrit plus haut.

Mode d'insertion Le mode d'insertion n'autorise ni déplacement ni commande, et doit être terminé par un ESCAPE. C'est dans ce mode que l'on ajoute effectivement du texte.

5.9.39 wc, comptage du nombre de lignes, de mots ou de caractères d'un fichier

On peut compter les lignes d'un fichier :

```
wc -l mon_fichier compte les lignes de mon_fichier
```

5.9.40 file, détermination du type du contenu d'un fichier

UNIX n'attache pas vraiment d'importance aux **extensions** pour déterminer le type d'un fichier : au contraire, on utilise en général une table *magique* permettant de déterminer ce type. La commande file utilise cette table :

```
schaefer@defian:~% file cat_join ANIMATE IMAGES/marc.jpeg bin/print_dilog
cat_join:          symbolic link to work/DIALOG/urbanet_backup/FLOPPY/server/cat_join
ANIMATE:          directory
IMAGES/marc.jpeg: JPEG image data, JFIF standard 1.01, aspect ratio X1:Y1
bin/print_it:     Bourne shell script text
```

6 Introduction à l'éditeur Emacs

Emacs est bien plus qu'un éditeur textuel. Nous allons décrire cependant deux aspects particuliers d'Emacs en tant qu'éditeur : la rédaction de texte et celle de programmes informatiques. Deux aspects supplémentaires fort utiles seront en outre évoqués : l'utilisation comme client mail (anglais: *mailer*) et comme éditeur de répertoire (anglais: *directory browser*). Plusieurs concepts sont propres à Emacs et ils seront brièvement décrits ici.

Le but de ce mode d'emploi est de familiariser le lecteur avec la puissance d'Emacs et d'ainsi faciliter son travail. Un conseil important : utiliser autant que nécessaire les commandes d'aide d'Emacs qui sont très complètes et suivre son **tutorial**. La version d'Emacs qui sera décrite est la 18.58, à part quelques passages sur la version 19. On trouvera une introduction plus complète dans [26].

```
emacs: /usr/local/bin/xemacs [20.2 XEmacs Lucid] adiff.hxx
File Edit Apps Options Functions Buffers Tools Rectangle/Register Spell C++ Mode Commands Help

#ifndef _adiff_h_
#define _adiff_h_

#include <cmath>

#ifdef STL_VECTOR
#include <vector>
#else
#include "aarray.h"
#endif

#include "error.h"

class md {
protected:

    static int nbvar; // nbre de variables du probleme

    int n;           // numero de variable, = 0 si ce n'est pas une variable
    double val;     // valeur
}

-----XEmacs: adiff.hxx 14:47 (C++ Compiling Font)-----L1--Top-----
cd /users/buse/dicesare/these/programs/Optimisation/lanad/exemple1/
make fem
g++ -O2 -felide-constructors -I/usr/X11R6/include/X11 -c aarray.cc
g++ -O2 -felide-constructors -I/usr/X11R6/include/X11 -c error.cc
g++ -O2 -felide-constructors -I/usr/X11R6/include/X11 -c adiff.cc
g++ -O2 -felide-constructors -I/usr/X11R6/include/X11 -c fmatrix.cc
g++ -O2 -felide-constructors -I/usr/X11R6/include/X11 -c graph.cc
g++ -O2 -felide-constructors -I/usr/X11R6/include/X11 -c geom.cc
g++ -O2 -felide-constructors -I/usr/X11R6/include/X11 -c gnuplot.cc

-----**XEmacs: *compilation* 14:47 (Compilation Compiling Font:run)-----L10--
```

FIG. 5 – Fenêtre typique XEmacs 20.20 sous X11

Un des problèmes courants lors de l'utilisation d'Emacs sur un terminal **VT100** est que certaines touches sont mal adaptées. Par exemple, la touche *Meta* doit être simulée par la touche **ESC**³⁶, en appuyant séparément, contrairement à **CTRL** ou *Meta*, et la séquence **CTRL-S**, bien souvent nécessaire, est inutilisable. Il arrive souvent que la touche **BACKSPACE** appelle le mode d'aide, il faut utiliser **DEL** pour effacer. Cependant, avec des **bindings** appropriés on peut venir à bout de ce genre de problèmes. Certains raccourcis ont été personnalisés. Pour obtenir les mêmes copier le fichier **.emacs** depuis `~ccestona`/³⁷. Le fichier **.emacs** est le fichier d'initialisation d'Emacs.

6.1 La fenêtre d'Emacs

Emacs se présente sous la forme d'une fenêtre principale. Elle est limitée vers le bas par une ligne d'information. Celle-ci vous apprend dans l'ordre : si le texte a été modifié (**) ou s'il est en mode lecture seulement (&&), ensuite le nom du document en mémoire (buffer) et finalement le mode d'édition (**Fundamental**, **Text**, etc.) et la proportion de texte sous la fenêtre. La ligne inférieure blanche est un mini-buffer qui permet de visualiser les commandes effectuées et de dialoguer avec Emacs.

³⁶Sur une station, la touche *Meta* existe en général, ou elle est remplacée par la touche **ALT**.

³⁷ne marche qu'à l'EPFL. En cours d'installation sur le **mips**.

6.2 Touches Control, Escape et Meta

Comme Emacs est uniquement textuel, l'appel de commandes se fait par l'intermédiaire du clavier. Deux touches permettent ces appels : la touche Control (*CTRL*) et la touche Meta (*Meta*). En général, les caractères à taper avec *CTRL* ou *Meta* sont en minuscules.

Exemple : On veut annuler une modification au texte.

UNDO appel court : *CTRL-X u*
 appel long : *Meta-X undo*

Cet exemple illustre les deux manières d'appeler une fonction soit : *CTRL-touche* suivi d'un caractère (optionnel) et *Meta-X nom_de_la_commande*.

Certains claviers sont dépourvus de touche *Meta*. On peut pallier à cet inconvénient en utilisant la touche Escape (*ESC*) avant la deuxième touche (normalement pressée en même temps). Toutes les commandes ont un nom. Dans notre exemple UNDO peut être appelée par *Meta-X undo* ou *CTRL-X u*. Dans les deux cas on tape d'abord la touche d'appel de commande, puis le nom de la commande. *CTRL-X u* est le raccourci clavier de la commande UNDO. La philosophie d'Emacs est d'économiser du temps en réalisant une foule d'opérations grâce à ses commandes. Elles ne valent souvent la peine que si on connaît leur équivalent clavier.

Reprenons l'exemple de UNDO. *Meta-X undo* est l'appel de la commande par son nom long. *CTRL-X u* est le raccourci clavier. Seules les commandes les plus souvent employées ont un équivalent clavier : les autres doivent être appelées par leur nom long.

6.3 Fonctions générales

Voici quelques fonctions très utiles pour utiliser Emacs :

CTRL-X extension de caractère: suivi d'un car. (v. plus haut)

Meta-X exécute une commande: suivi d'un nom long. (v. plus haut)

CTRL-G arrête un calcul ou une opération en cours. Cette commande est très utile lorsque l'on a invoqué une fonction involontairement ou lorsqu'Emacs attend une réponse et que l'on ne veut pas lui en donner.

CTRL-L refresh et recentrage. Réécrit l'écran actuel et place le curseur sur la ligne centrale.

CTRL-X u undo, pour annuler une manipulation erronée.

CTRL-X CTRL-C quitte Emacs. Attention, cette commande ne demande pas de confirmer la sauvegarde de nouveaux buffers.

TAB ou *Espace* complétion dans le minibuffer. La complétion n'est pas possible dans le cas où le mini-buffer précise **regexp**. Dans le cas où plusieurs possibilités sont offertes, ouvre une fenêtre les affichant.

6.4 Fenêtres

Dans Emacs, il est possible d'avoir plusieurs fenêtres et de les manipuler. Par exemple les fonctions d'aide invoquées ouvrent en général une fenêtre annexe que l'on identifie grâce à sa ligne d'information. On peut la supprimer ensuite grâce à *CTRL-X 1*.

CTRL-x 1 tue toutes les fenêtres sauf celle dans laquelle se trouve le curseur.

CTRL-x 0 tue la fenêtre où se trouve le curseur (si ce n'est pas la seule fenêtre).

CTRL-x 2 sépare la fenêtre en deux avec une ligne horizontale.

CTRL-x 5 idem avec une ligne verticale.

CTRL-x o place le curseur dans une autre fenêtre.

CTRL-Meta-w déroule l'autre fenêtre vers le bas.

Dans Emacs 19.30, *CTRL-x 3* remplace *CTRL-x 5* (ce dernier, suivi de 2 permet de créer une nouvelle fenêtre X11).

6.5 Fichiers et buffers

Les fichiers sont bien connus en informatique. Emacs ne modifie bien entendu pas les fichiers directement : il doit d'abord les charger en mémoire.

Emacs contient les textes en mémoire, sous forme de buffer. On peut dès lors avoir plusieurs textes en mémoire, chacun ayant un nom. Ce nom se situe dans la bande d'information en bas de la fenêtre. Mieux, on peut les visualiser simultanément, chacun dans une fenêtre différente. On peut en voir la liste et sauter de l'un à l'autre. Le buffer de défaut, est **scratch**.

CTRL-x CTRL-f charge un fichier dans le buffer actuel.

CTRL-x CTRL-s sauve le fichier contenu dans le buffer actuel.

CTRL-x CTRL-w idem mais demande le nom sous lequel doit être sauvé le buffer.

CTRL-x i insère un fichier dans le buffer actuel.

CTRL-x b permet de choisir le buffer dans la fenêtre.

CTRL-x CTRL-b liste les buffers utilisés dans une fenêtre. Pour changer de fenêtre, voir la section 6.4. *f* permet une sélection avec le curseur (dans cette fenêtre).

CTRL-x s sauve tous les buffers ayant été modifiés.

CTRL-x k tue un buffer (effacement de la mémoire et de la fenêtre).

Meta-x print-buffer imprime le buffer à l'imprimante.

CTRL-x 4 f charge un fichier dans une autre fenêtre, ouverte à cet effet.

CTRL-x 4 b change de buffer dans l'autre fenêtre (si elle est ouverte).

CTRL-x CTRL-q inverse le mode lecture-seulement (&&) du buffer actuel.

6.6 Déplacements

A part les touches du curseur on peut utiliser respectivement :

CTRL-v et *Meta-v* en avant, en arrière d'un écran.

Meta-< et *Meta->* au début, à la fin du buffer.

CTRL-a et *CTRL-e* au début, à la fin d'une ligne.

Meta-e et *Meta-a* en avant, en arrière d'une phrase.

Meta-f et *Meta-b* en avant, en arrière d'un mot.

6.7 Edition

Voici les fonctions utiles pour éditer textes et programmes :

CTRL-o insère une ligne vierge à la hauteur du curseur.

Meta-DEL efface le mot immédiatement avant.

Meta-d efface le mot immédiatement après.

Meta- efface les espaces au milieu desquels se trouve le curseur.

CTRL-d efface le caractère sous le curseur.

CTRL-t inverse le caractère actuel avec le précédent.

CTRL-x CTRL-t inverse la ligne du curseur avec la précédente.

Meta-u met le mot suivant le curseur en majuscules.

Meta-l met le mot suivant le curseur en minuscules.

6.8 Copier et coller

Voici deux opérations fort utiles lors de l'édition. Plusieurs possibilités sont offertes par Emacs pour les effectuer. La plus simple est d'utiliser *CTRL-k* pour effacer des lignes. Quand une ligne est effacée, elle est stockée en mémoire dans un tampon. On affiche ce tampon (**buffer**) grâce à *CTRL-y*. En répétant plusieurs *CTRL-k* à la suite, les lignes s'ajoutent les unes aux autres dans le tampon. On peut ainsi couper puis coller un morceau de texte important. Mais ces fonctions s'empilent réellement : on peut couper un morceau A puis un second B, réécrire ce morceau B, avec *CTRL-y* puis réécrire celui coupé en premier soit A, en utilisant *CTRL-y* puis *Meta-y*. On peut ainsi empiler autant de couper coller que l'on veut et retrouver un morceau effacé au début de l'édition. Plus pratique encore est l'utilisation de régions. On peut ainsi copier une région sans l'effacer.

CTRL-k efface jusqu'à la fin de la ligne et ajoute ce qui a été effacé au tampon cumulatif.

CTRL-y réécrit (ailleurs) ce qui a été effacé, c.-à-d. le tampon.

Meta-y après *CTRL-y*, remplace l'insertion créée par *CTRL-y* avec le tampon précédent (dans la pile de tampons).

6.9 Point, Marque, Région

Emacs permet de manipuler des blocs de texte, appelés ici *régions*. On définit la région entre le point et la marque. Le point est synonyme de curseur : c'est le carré noir où se fait l'insertion de texte. La marque est quant à elle invisible. On peut la fixer où on veut, mais attention à bien se rappeler où elle se trouve. Sinon, on peut échanger le point et la marque pour identifier cette dernière. La manipulation de régions est très pratique pour copier une partie de texte sans l'effacer, par exemple.

Définitions :

Point pointeur symbolisé par le curseur.

Marque pointeur invisible, déplaçable.

Région zone délimitée par le point et la marque.

Commandes :

CTRL-@ place la marque à l'endroit du curseur. C'est l'équivalent de *CTRL-Espace*.

CTRL-x CTRL-x échange le point et la marque.

Meta-@ place la marque après le prochain mot (région=mot).

Meta-h définit la région comme le paragraphe actuel.

CTRL-x CTRL-p idem avec la page actuelle.

CTRL-x h idem avec tout le buffer actuel.

CTRL-w coupe (effacement) la région (à réécrire avec *CTRL-y*).

Meta-w copie la région (sans l'effacer) (à réécrire avec *CTRL-y*).

CTRL-x TAB indente la région d'un espace.

CTRL-x CTRL-l met la région en minuscule.

CTRL-x CTRL-n met la région en majuscule.

Meta=- compte les lignes de la région.

CTRL-x n restreint le buffer à la région, pour édition temporaire.

CTRL-x w rend le buffer entier éditable à nouveau.

Meta-x write-region sauve la région sous un nom donné.

Meta-x print-region envoie la région à l'imprimante.

6.10 Registres

Les registres permettent de mémoriser des morceaux de texte ou des emplacements du curseur. On choisit le nom d'une lettre pour chaque registre qu'on souhaite utiliser. Mais il ne faut pas confondre buffers et registres. On peut éditer un buffer, le sauver mais ces opérations ne s'appliquent pas à un registre. Ceux-ci servent temporairement à contenir une partie de texte durant l'édition.

CTRL-x x copie la région dans un registre NOM.

CTRL-x r idem avec le rectangle incluant la région.

CTRL-x g insère le registre NOM après le curseur.

CTRL-x / sauve la position du curseur dans un registre NOM.

CTRL-x j place le curseur à l'endroit sauvé dans le registre NOM.

Meta-x view-register affiche le contenu du registre NOM.

Meta-x append-to-register appond la région à un registre.

Meta-x prepend-to-register idem mais au début du registre.

6.11 Recherche et remplacement

Ces fonctions se comprennent d'elles-mêmes !

Meta-% remplace un mot par un autre.

CTRL-s recherche une chaîne de caractères vers l'avant.

CTRL-r recherche une chaîne de caractères vers l'arrière.

6.12 Macros

Il est possible d'économiser la frappe de parties de textes (ou de commandes) répétitives grâce à une macro dans Emacs. Il faut d'abord définir la macro et ensuite on peut l'invoquer à tout moment.

CTRL-x (enregistre une frappe au clavier définissant la macro.

CTRL-x) termine cet enregistrement.

CTRL-x e appelle la dernière macro définie.

6.13 Répétitions et argument universel

La plupart des commandes d'Emacs prennent un argument. C'est-à-dire que ces fonctions effectuent une tâche d'une certaine manière, qui peut être précisée grâce à un argument numérique. Par exemple, la commande new-line (*RETURN*) génère un saut à la ligne. Mais on peut lui passer un nombre comme argument, 4 par exemple, ce qui aura pour effet de générer 4 sauts de ligne. Il faut entrer l'argument avant d'invoquer la fonction. La plupart des commandes répètent leur action le nombre de fois précisé par l'argument. L'argument peut aussi servir une seule fois à la commande, comme par exemple pour *set-left-margin*.

CTRL-u nbre argument universel. Le nombre sera passé comme paramètre à la prochaine commande appelée.

CTRL-x ESC répète la dernière fonction complexe.

6.14 Texte

Ces fonctions se comprennent d'elles-mêmes !

Meta-x text-mode passe en mode texte.

Meta-s centre une ligne.

Meta-S centre un paragraphe.

CTRL-x f fixe la marge à droite.

Meta-q reformate le paragraphe sans dépassement de la marge.

Meta-g idem avec la région.

6.15 Commentaires

Emacs permet de commenter un programme informatique.

Meta-; insère un commentaire (sauf en début de ligne) ou l'aligne s'il y en a déjà un sur la ligne.

CTRL-x ; fixe la colonne des commentaires.

Variables comment-start contient la chaîne de début de commentaire.

comment-end idem avec fin de commentaire.

comment-start-skip chaîne contenant un commentaire vide.

6.16 Abréviations

Une abréviation est un ensemble de caractères qui est utilisé à la place d'un mot plus long. Imaginons que l'utilisateur doive taper le mot `programme` plusieurs fois. Il lui sera possible de définir une abréviation `prg` qu'il tapera à la place du nom complet. Les abréviations peuvent être étendues pour donner au lecteur le nom original qu'elles représentent.

CTRL-x + génère une abréviation du mot précédant le curseur.

CTRL-x ' étend l'abréviation précédant le curseur.

Meta-x list-abbrev donne une liste des abréviations définies.

Meta-x abbrev-mode passe au sous-mode d'abréviations.

Meta-x edit-abbrevs permet d'éditer les abréviations.

CTRL-c CTRL-c met à jour les abréviations éditées.

Meta-x write-abbrev-file écrit un fichier contenant les abréviations.

Meta-x read-abbrev-file lit le fichier contenant les abréviations.

NOTE : Par défaut le fichier contenant les abréviations est nommé `~/abbrev.defs`.

6.17 Répertoire

Emacs permet d'éditer des répertoires sur le disque utilisé.

CTRL-c CTRL-d affiche simplement le contenu d'un répertoire.

CTRL-x d édite un répertoire, dont le nom est demandé.

g relit le répertoire (update).

m marque un fichier en vue de commandes ultérieures.

R copie les fichiers marqués.

d marque un fichier comme à effacer.

c copie un fichier.

f édite le fichier.

M `chmod` (argument=nombre à 3 chiffres, convention **octale** UNIX).

r renomme un fichier.

u enlève une marque.

v permet de voir un fichier.

x exécute les effacements.

marquer les copies de sécurité débutant par *#* pour effacement.

C compresse un fichier.

U décompresse un fichier.

Note: on peut se déplacer dans l'arborescence grâce à *v* et *CTRL-x k*.

6.18 Mail (mode d'écriture seulement)

Ce mode permet de composer des messages de **mail**.

CTRL-x m passe au mode Mail.

CTRL-x 4m ouvre une autre fenêtre et passe au mode Mail.

CTRL-c CTRL-s envoie un message.

CTRL-c CTRL-c envoie un message et tue le buffer.

CTRL-c CTRL-w insère le fichier **.signature** à la fin du mail.

6.19 Rmail (lecture/écriture de mails)

Attention : Il est dangereux d'utiliser ce mode si l'on ne connaît pas bien Emacs, car on risque de perdre son courrier privé. Ce mode permet de lire, de répondre à des messages et d'en composer de nouveaux. En fait il permet d'éditer le fichier **RMAIL**³⁸ où viennent s'ajouter les mails reçus.

Meta-x rmail-mode passage au mode Rmail.

Meta-x rmail-summary ouvre une fenêtre avec un menu des mails. La sélection se fait avec *j*.

. va au début du message.

ESPACE affiche le prochain écran du message.

DEL affiche l'écran précédent du message.

n va au prochain message.

p va au message précédent.

> va au dernier message.

NB j va au message numéro NB (où NB est un nombre).

d efface le message et va au prochain.

u réécrit le message effacé.

q quitte Rmail et sauvegarde les effacements.

CTRL-x CTRL-s idem sans sauver.

m édite un message (mode mail).

r répond à un message.

f envoie le message courant à un autre utilisateur (**forward**).

w édite le message courant.

CTRL-o sauve le message.

6.20 Aide

Emacs propose des fonctions d'aide très complètes.

CTRL-h a **apropos**, liste les commandes contenant l'argument.

CTRL-h f décrit une fonction.

CTRL-h k décrit une touche.

CTRL-h b affiche tous les raccourcis clavier.

CTRL-h s décrit la syntaxe d'une fonction.

CTRL-h w donne l'équivalent clavier d'une fonction.

CTRL-h m décrit un mode de travail, ses fonctions spécifiques.

CTRL-h i **GNU Info**. Voyage dans une arborescence d'informations très vaste.

6.21 Impression

Emacs permet d'imprimer un buffer directement sans l'avoir sauvé (voir section 6.5). On peut choisir l'imprimante à utiliser avec la commande

```
set-variable lp-switches -Pnom_imprimante
```

Note: rappelle la syntaxe de `lp` ou `lpr`. Sous UNIX, on peut soit indiquer l'imprimante avec le paramètre `-P` ou à l'aide d'une variable d'environnement, par exemple, avant de lancer Emacs :

```
setenv PRINTER nom_imprimante
```

Vous trouverez plus d'informations à ce sujet dans la section 8.

³⁸ne pas confondre avec la commande **rmail**, décrite à la section 20.2.2.

6.22 Exemple de .emacs

```
;; Exemple de fichier de configuration pour Emacs (fichier .emacs)

;; Exemple de définition de fonction:
;; Cette fonction permet de dérouler l'autre fenêtre vers le
;; bas d'un certain nombre de lignes.

(defun scroll-other-window-down (n)
  "Scroll next window downward ARG lines;
A near full screen is 'next-screen-context-lines' less than a full screen.
When calling from a program, supply a number as argument or nil."
  (interactive "P")

  (scroll-other-window (if n
    (- (prefix-numeric-value n)
      '-)))
;; Cette fonction sera utilisable par un raccourci-clavier à condition
;; d'ajouter la ligne suivante (meta-control-w):

(define-key esc-map "\C-w" 'scroll-other-window-down)

;; Voilà comment changer la valeur par défaut de différentes variables:

(setq default-left-margin 1)           ;; marge à gauche
(setq default-major-mode 'text-mode)   ;; mode de text
(setq text-mode-hook 'turn-on-auto-fill) ;; retour à la ligne automatique
(setq mail-signature t)                ;; insertion automatique du fichier
                                       ;; .signature en fin de mail

;; Voilà encore un appel de fonction fort utile puisqu'il permet
;; de rendre la région visible (écran couleur).

(transient-mark-mode 1)

;; Ces deux appels de fonction permettent d'avoir l'heure dans
;; la ligne d'information ainsi que le numéro de la ligne dans
;; laquelle se trouve le curseur.

(line-number-mode 't)
(display-time)

;; Pour utiliser les abréviations:

(read-abbrev-file "~/abbrev_defs")    ;; lit le fichier
(setq default-abbrev-mode t)          ;; expend par défaut

;; Encore une petite fonction personnalisée: idem à rmail-summary mais
;; économisant de la place dans la fenêtre de mail.

(defun rmail-small-sum ()
  "Same as rmail-summary but smaller window"
  (interactive)
  (rmail-summary) (shrink-window 5 ))

;; Voilà comment personnaliser les raccourcis-clavier:

(global-set-key "\C-r" 'rmail)        ;; C-r appelle le mode rmail
(global-set-key "\M-r" 'rmail-small-sum) ;; M-r affiche la liste des mail

;; NOTE: les équivalents précédemment définis pour C-r et M-r sont perdus.
```

```
;; Pour ceux qui ont la chance d'utiliser EMACS 19, voici des COULEURS !!!
;; Note: les identificateurs utilisés pour les couleurs dépendent du
;; système utilisé.
```

```
(cond (window-system
      (setq hilit-mode-enable-list '(not text-mode)
            hilit-background-mode  'light
            hilit-inhibit-hooks    nil
            hilit-inhibit-rebinding nil)

      (require 'hilit19)
    ))
```

```
(hilit-translate string 'cyan
                 keyword 'green
                 )
```

```
;; Les lignes qui suivent sont facultatives et peuvent nécessiter
;; des modifications (couleurs) sur votre machine.
```

```
(set-background-color "grey20")
(set-foreground-color "yellow")
(set-cursor-color "green")
(set-face-background 'region "grey1" )
```

```
;; Pour avoir les couleurs actualisées avec C-l
```

```
(setq hilit-auto-rehighlight 't)
```

```
;; Pour fonctionner en ISO Latin 1 (accents)
```

```
;; Affichage des caractères 8 bits.
(standard-display-european t)
```

```
;; Les nouveaux caractères sont des lettres.
(require 'iso-syntax )
```

```
;; Incantation magique, pas très nécessaire
(set-input-mode (car (current-input-mode))
                (nth 1 (current-input-mode)) 0)
```

6.23 Mémento

Général

CTRL-x extension de caractère: suivie d'un caractère.

Meta-x exécute une commande avec un nom long.

CTRL-g arrête un calcul ou une opération en cours.

CTRL-l rafraîchissement (**refresh**) et recentrage.

CTRL-x u annulation (**undo**).

CTRL-x CTRL-c quitte Emacs.

TAB ou *Espace* complétion dans le minibuffer, si possible.

? aide ou propositions dans le minibuffer.

Fichiers

CTRL-x CTRL-f charge un fichier (buffer actuel).

CTRL-x CTRL-s sauve le fichier (buffer actuel).

CTRL-x CTRL-w idem mais sous un autre nom.

CTRL-x i insère un fichier dans le buffer actuel.

Buffers

CTRL-x b permet de choisir le buffer dans la fenêtre.

CTRL-x CTRL-b liste les buffers utilisés.

f permet une sélection avec le curseur.

CTRL-x s sauve les buffers modifiés.

CTRL-x k tue un buffer.

Meta-x print-buffer imprime le buffer.

CTRL-x 4 f charge un fichier dans une autre fenêtre.

CTRL-x 4 b change de buffer dans l'autre fenêtre.

CTRL-x CTRL-q inverse le mode lecture-seulement (&&).

Déplacement

CTRL-v ou *Meta-v* en avant, en arrière d'un écran.

Meta-< ou *Meta->* au début, à la fin du buffer.

CTRL-a ou *CTRL-e* au début, à la fin d'une ligne.

Meta-e ou *Meta-a* en avant, en arrière d'une phrase.

Meta-f ou *Meta-b* en avant, en arrière d'un mot.

Edition

CTRL-o insère une ligne vierge à la hauteur du curseur.

CTRL-d efface le caractère sous le curseur.

CTRL-k efface jusqu'à la fin de la ligne (tampon cumulatif).

CTRL-y réécrit (ailleurs) ce qui a été effacé (=tampon).

Meta-y après *CTRL-y*, remplace l'insertion par le tampon précédent.

Meta-DEL efface le mot immédiatement avant.

Meta-d efface le mot immédiatement après.

Meta- efface les espaces au milieu desquels se trouve le curseur.

CTRL-t inverse deux caractères (actuel échangé avec précédent).

CTRL-x CTRL-t inverse la ligne du curseur avec la précédente.

Meta-u met le mot suivant le curseur en majuscules.

Meta-l idem en minuscules.

Fenêtres

CTRL-x l tue toutes les autres fenêtres.

CTRL-x 0 tue sa propre fenêtre (si elle n'est pas unique).

CTRL-x 2 sépare fenêtre en deux, ligne horizontale.

CTRL-x 5 idem avec ligne verticale (3 avec Emacs 19.30).

CTRL-x o place le curseur dans une autre fenêtre.

CTRL-x b permet de choisir le buffer dans la fenêtre.

CTRL-Meta-v déroule l'autre fenêtre vers le bas.

Point, marque, région

CTRL-@ place la marque.

CTRL-x CTRL-x échange le point et la marque.

Meta-@ place la marque après le prochain mot (région=mot).

Meta-h définit la région comme le paragraphe actuel.

CTRL-x CTRL-p idem avec la page actuelle.

CTRL-x h idem avec tout le buffer actuel.

CTRL-w tue la région (à réécrire avec *CTRL-y*).

Meta-w copie la région (à réécrire avec *CTRL-y*).

CTRL-x TAB indente la région (1 espace).

CTRL-x CTRL-l met la région en minuscules.

CTRL-x CTRL-n met la région en majuscules.

Meta= compte les lignes de la région.

CTRL-x n restreint le buffer à la région, pour édition temporaire.

CTRL-x w rend le buffer entier éditable à nouveau.

Meta-x write-region sauve la région sous un nom donné.

Meta-x print-region envoie la région à l'imprimante.

Registres

CTRL-x x copie la région dans un registre NOM.

CTRL-x r idem avec le rectangle incluant la région.

CTRL-x g insère le registre NOM après le curseur.

CTRL-x / sauve la position du curseur dans un registre NOM.

CTRL-x j place le curseur à l'endroit sauvé dans le registre NOM.

Meta-x view-register affiche le contenu du registre NOM.

Meta-x append-to-register ajoute la région à un registre.

Recherche et Remplacement

Meta-% remplace un mot par un autre.

CTRL-s recherche une chaîne de caractères vers l'avant.

CTRL-r recherche une chaîne de caractères vers l'arrière.

Macros

CTRL-x (enregistre une frappe au clavier définissant la macro.

CTRL-x) termine cet enregistrement.

CTRL-x e appelle la dernière macro définie.

Répétition

CTRL-u nbre argument universel. Effectue une fonction nbre fois.

CTRL-x ESC répète une fonction complexe.

Ecriture de texte

Meta-x text-mode passe au mode de texte.

Meta-s centre une ligne.

Meta-S centre un paragraphe.

CTRL-x f fixe la marge à droite.

Meta-q reformate le paragraphe sans dépassement de la marge.

Meta-g idem avec la région.

Répertoire

CTRL-c CTRL-d affiche simplement le contenu d'un répertoire.

CTRL-x d édite un répertoire, dont le nom est demandé.

Dans le mode d'édition de répertoire

g relit le répertoire (update).

m marque un fichier

R copie les fichiers marqués.

d marque fichier comme étant à effacer.

c copie un fichier.

f édite le fichier.

M chmod.

r renomme un fichier.

u enlève une marque.

v permet de voir un fichier.

x exécuter les effacements.

marquer les fichiers débutant par *#* pour effacement.

C compresse un fichier.

U décompresse un fichier.

Mail

CTRL-x m passe au mode Mail.

CTRL-x 4m ouvre une autre fenêtre et passe au mode Mail.

CTRL-c CTRL-s envoie un message.

CTRL-c CTRL-c envoie un message et tue le buffer.

CTRL-c CTRL-w insère le fichier **signature** à la fin du mail.

Rmail

Meta-x rmail-mode passage au mode Rmail.

Meta-x rmail-summary ouvre une fenêtre avec un menu des commandes mail. La sélection se fait avec *j*.

. va au début du message.

ESPACE affiche le prochain écran du message.

DEL affiche l'écran précédent.

n va au prochain message.

p va au message précédent.

> va au dernier message.

NB j va au message numéro NB (où NB est un nombre).

d efface le message et va au prochain.

u réécrit le message effacé.

q quitte Rmail et sauvegarde les effacements.

CTRL-x CTRL-s idem sans sauver.

m édite un message (mode mail).

r répond à un message.

f envoie le message actuel à un autre utilisateur.

w édite le message actuel.

CTRL-o sauve le message.

Aide

CTRL-h a **apropos**, liste les commandes contenant l'argument.

CTRL-h f décrit une fonction.

CTRL-h k décrit une touche.

CTRL-h b affiche tous les raccourcis clavier.

CTRL-h s décrit la syntaxe.

CTRL-h w donne l'équivalent clavier d'une fonction.

CTRL-h m décrit un mode de travail.

CTRL-h i GNU Info.

7 Gestion des fichiers

Cette section résume les principales commandes pour les principales fonctions de gestion des fichiers sous UNIX.

7.1 Edition de fichiers

Les commandes `vi`, `emacs`, `uemacs`, `ne` et `ed` permettent d'éditer (modifier ou créer) des fichiers.

7.2 Administration des fichiers

Les commandes `cp`, `mv`, `mkdir`, `rm`, `rmdir`, `ls`, `tail`, `head`, `wc`, `grep`, `du` et `find` permettent de gérer vos fichiers, comme par exemple copier, déplacer, supprimer, rechercher, etc.

7.3 Conversion de formats

Par convention, on fait souvent suivre un nom de fichier d'une **extension** formée d'un point ainsi que d'une abréviation du type, souvent sur trois lettres, en particulier sous environnements propriétaires **Microsoft**. La table 5 liste quelques-unes de ces extensions et indique les programmes de conversion ou de visualisation appropriés.

Très important : ce qui compte en règle générale pour UNIX n'est pas l'extension, mais bien le contenu des fichiers. La commande `file` (voir section 5.9.40 page 32) permet de déterminer le type du contenu d'un fichier. Les interfaces graphiques utilisent bien souvent le type **MIME**, voire parfois également les extensions pour déterminer le type des fichiers.

Extension	type	création	extraction
.jpg .jpeg	Image compressée en général avec perte	cjpeg	djpeg
.gif	Image	netpbm	netpbm, xv
.xbm	Image (X bitmap)	netpbm	netpbm, xv
.lha .lzh	Archive compressée	lha	lha, lharc
.zip	Archive compressée	zip	unzip
.gz .z	fichier compressé	gzip	gzip -d, gunzip
.bz2	fichier compressé	bzip2	bzip2 -d
.Z	fichier compressé	compress	uncompress, gzip -d
.shar	archive shell	shar	unshar
.tar	archive	tar cvf	tar xvf
.au	Son	sox	sox
.iff	Image ILBM/Amiga	netpbm	netpbm
.zoo	Archive compressée	zoo	zoo
.arj	Archive compressée	-	unarj
.arc	Archive compressée	arc	arc
.dvi	T _E X	tex	xdvi, dvips
.ps	Postscript	divers	ghostview

TAB. 5 – Extensions des différents formats

Les extensions peuvent être combinées : p.ex. `.tar.gz` ou `.tgz`. Dans ce cas, il faut d'abord décompresser avec `gzip` puis désarchiver avec `tar` (voir section 5.9.36 page 30).

Notez que l'on extrait les fichiers d'une archive **shar** soit au moyen du **shell** `sh` en lisant les indications se trouvant au début du fichier, ou de la commande `unshar`.

7.3.1 Formats de disquette

Beaucoup de systèmes UNIX sont livrés avec un lecteur de disquettes bas-de-gamme compatible avec d'autres systèmes au niveau physique. Suivant les systèmes d'exploitation et la variante d'UNIX, il est possible d'accéder à des systèmes de fichiers différents (par exemple **MS-DOS** ou **OS/2**). Si ce support ne se trouve pas en standard, il est souvent possible de porter la suite de logiciels **mttools** qui permettent d'accéder d'une façon peu transparente (pas d'accès direct aux fichiers, des commandes spécifiques de copie doivent être utilisées). Parmi ces commandes, citons : `mdir`, `mread`, `mwrite`. Certains systèmes offrent une commande pour l'éjection de disquettes : `eject`. Il est possible d'utiliser certaines variantes de la commande `tar` pour séparer des archives en petits fichiers (on peut aussi utiliser les commandes `split` ou `sploin`, on utilisera alors un simple `cat` pour réunir les différentes parties du fichier). On peut aussi soit utiliser `tar` directement sur le périphérique "physique" (ou même `cp` sous certaines réserves, notamment la détection de fin de fichier), une disquette étant considérée comme un fichier sous UNIX,

soit monter la disquette (mount) et alors la disquette est considérée comme étant une arborescence. Il ne faut alors plus utiliser le périphérique physique (p.ex. /dev/fd0) mais bien le point de montage spécifié lors du mount).

7.3.2 Conversions de textes accentués

Le programme **accents** permet la conversion de formats divers (**Macintosh**, **MS-DOS**, etc.) en **ISO Latin 1**. Par exemple, pour convertir un texte MS-DOS en texte UNIX, on utilisera la commande suivante.

```
accents < msfile > unixfile -from pc -to iso
```

Notez qu'il faudra peut-être encore convertir les fins de lignes (à l'aide d'un éditeur comme **vi** ou **emacs**, on peut le faire facilement) (voir section 5.9.38 page 32). Il existe un programme plus complexe permettant de faire des conversions de formats de textes, l'utilitaire **GNU recode**. **tr** peut aussi être utilisé.

7.3.3 Conversions de sons et d'images : sox et netpbm

Pour la conversion de sons ou d'images, les commandes **sox** et le sous-ensemble de commandes des logiciels **cjpeg**, **djpeg**, et **netpbm** sont disponibles.

7.3.4 Conversions de fichiers multimédia

Le protocole de codage **MIME** permet de transmettre n'importe quel type de données de façon encapsulée via courrier électronique. Les commandes **mpack** et **munpack** permettent de convertir le format encapsulé et encodé en un format utilisable. On peut aussi utiliser **uuencode** et **uudecode** (voir section 22.7 page 78).

7.4 Compression

Les commandes **compress**, **gzip** et quelques autres permettent la compression de données. En général on préférera utiliser **gzip** (éventuellement couplé à **tar**) ou éventuellement **lha** ou **zip**, voire **bzip2** pour de gros fichiers comprenant du texte pour lesquels le gain en place est non négligeable.

7.4.1 Comparaison de logiciels de compression

La comparaison de logiciels de compression de données est assez difficile : toutes les plates-formes ne disposent pas des mêmes logiciels et surtout toutes les implémentations d'algorithmes identiques ne sont pas équivalentes. Parfois, des fonctionnalités spéciales sont possibles : conversion de jeux de caractères, mode **pipe**, etc. La plupart du temps ce n'est pas le cas. Il y a parfois des problèmes de brevets (particulièrement sur les algorithmes de compression dits arithmétiques).

Les critères de comparaison sont les suivants :

- Les licences associées doivent être larges ;
- le logiciel doit fonctionner sous UNIX ;
- les sources doivent être disponibles et portables ;
- l'algorithme est sans pertes ;
- le jeu de test doit être publié.

Les informations de la table 6 ont été extraites de la **FAQ de comp.compression** et ont été obtenues à l'aide d'essais effectués sur un jeu standard de fichiers documentés dans cette même FAQ. Ce jeu standard comporte 18 fichiers de différents types (texte, exécutable, image, etc.) totalisant de l'ordre de 3 MB. Le temps doit être évidemment considéré uniquement de façon relative.

Logiciel	version	famille	taille [octets]	temps compression
compress	4.0	LZW	1259141	1 mn
lharc	1.02	LZ77	1200580	5 mn
yabba	1.0	LZ78	1159340	3 mn
freeze	2.3.5	LZ77	1109290	6 mn
zoo	2.10	LZ77H	1096166	4 mn
lha	1.0	LZ77H	1096138	6 mn
zip/gzip	1.9	LZ77H	1021043	2 mn
hpack	0.75a	PPMC	1005367	85 mn
comp	2	arithm.	890976	27 mn

TAB. 6 – Comparaison d'algorithmes de compression

Notons que cette comparaison est assez ancienne. J'ai effectué un test similaire pour comparer, dans ce cas précis, les deux compresseurs les plus populaires sous UNIX aujourd'hui (2002), soient **gzip** et **bzip2**, avec la compression maximum (-9), sur

les mêmes données standards, via une commande `tar`. On obtient, sur un Mobile Pentium II/366 avec Debian GNU/Linux 2.2 et kernel Linux 2.2.18 :

Logiciel	version	famille	taille [octets]	temps compression
gzip	1.2.4	LZ77H	1063917	7 secondes
bzip2	0.9.5d	block-sorting	892778	7 secondes

Ces valeurs sont trompeuses : `bzip2` prend normalement bien plus de temps et de mémoire pour compresser que `gzip`, et parfois (rarement) est moins efficace. Par contre, sur du texte répétitif comme par exemple des articles de news réunis en une archive (format `rnews`), `bzip2` montre toute son efficacité.

7.5 Transferts de fichiers

Les commandes `sz` et `rz` permettent de transférer des fichiers jusqu'à votre terminal ou ordinateur personnel, respectivement d'envoyer des fichiers sur votre compte. (voir section 32.3.3 page 105).

La commande `ftp` permet de transférer des fichiers entre systèmes distants sur **Internet** (voir section 24 page 85). La commande `rnp`, ou mieux, `scp`, permet quant à elle de faire de la copie de fichiers de la même façon que la commande `cp`, mais sur des machines distantes où vous devez nécessairement avoir un accès. Notez que le transfert de fichiers peut être transparent grâce à **NFS**, qui permet le partage de disques sur un réseau. On peut aussi transférer des fichiers via **mail** (voir section 22.7 page 78).

7.6 Chiffrement de fichiers

Pour le chiffrement (anglais: *encryption*) de fichiers, on utilisera **crypt** ou **PGP**, avec une préférence pour PGP en raison de ses possibilités étendues et de la grande robustesse de l'algorithme. Plus de détails sur le chiffrement en général se trouvent à la section 9.3.3 page 51.

7.6.1 pgp

Introduction **PGP** (anglais: *Pretty Good Privacy*) est un logiciel implémentant une cryptographie à clé révélée (**public-key cryptography**, asymétrique) développé par Philip Zimmermann, un consultant en sécurité aux Etats-Unis. Ce logiciel permet l'envoi de messages authentifiés (dont on peut s'assurer que l'expéditeur est bien celui qui a signé le message) et/ou cryptés (dont le déchiffrement ne peut se faire qu'à l'aide d'une deuxième clé, nommée la clé privée et qui n'a pas besoin d'être diffusée, contrairement aux méthodes de chiffrement classiques). Il implémente également d'autres fonctionnalités très pratiques, par exemple, pour le test d'un logiciel par un groupe fermé d'utilisateurs. Il est basé sur deux algorithmes principaux, réputés cryptographiquement sûrs : le premier est **RSA**, un algorithme à **clé révélée**³⁹, reposant sur la difficulté de la factorisation de nombres entiers. Le deuxième algorithme est **IDEA**, développée par ASCOM TECH. Il s'agit d'un "simple" algorithme à clé secrète (il faut connaître la clé qui a été utilisé pour le chiffrement pour décoder), mais avec des caractéristiques de résistance à l'analyse cryptographique très fortes. Cet algorithme est utilisé en conjonction avec RSA pour augmenter la vitesse de l'ensemble (tout faire en RSA aurait été trop lent).

Disponibilité Les deux algorithmes (surtout celui d'ASCOM) sont libres de licences⁴⁰ en Suisse et dans la plupart des pays occidentaux, à part aux Etats-Unis, où une compagnie, RSA DSI, a breveté l'algorithme RSA, interdisant l'utilisation sans paiement de licence, et l'exportation à cause de la loi sur les munitions (**ITAR**) et dans certains pays restrictifs sur les moyens de chiffrement (en particulier la France, où tout procédé de chiffrement ou d'authentification doit faire l'objet d'une autorisation, et la Russie, où des lois restrictives ont été récemment adoptées).

Par sa nature, permettant d'assurer une certaine protection de la sphère privée et simultanément autorisant l'authentification de personnes, ce logiciel a créé beaucoup de controverses.

Le logiciel lui même est disponible sans licence sous forme source pour UNIX ainsi que d'autres plateformes (comme MacOS, MS-DOS, AmigaOS, etc.).

Notons que plusieurs versions existent, notamment la 2.3a et la 2.6ui. Cette dernière est la version recommandée (i pour international, donc le code RSA n'a pas été illégalement exporté des Etats-Unis, mais écrit sur territoire étranger, afin de contourner les lois ridicules d'exportation US – lois qui ont été assouplies depuis).

Les versions de PGP antérieures à la 2.6 enfreignent donc les lois US protégeant les droits d'auteurs de la firme RSA DATA SECURITY INTERNATIONAL, l'exploitant des brevets **RSA**. Pour cette raison de droits d'auteurs, les fichiers produits par la version 2.6 ne sont plus compatibles avec la version 2.3. Cela rend cette dernière moins intéressante et pousse les utilisateurs à mettre à jour leurs logiciels et donc à renoncer à des versions exploitant peut-être illégalement certains algorithmes [31].

Quant aux rumeurs ayant prétendu que les dernières versions ne sont pas aussi sûres que les précédentes, elles ont été démenties par l'auteur. Les sources sont là pour le vérifier. De plus, l'archive PGP est en général signée.

³⁹La clé pour crypter est publiée largement, elle est différente de la clé pour décrypter et jouit de propriétés intéressantes : on peut par exemple signer électroniquement avec la clé privée, et à l'autre bout vérifier l'identité de l'auteur. La différence principale avec un algorithme à clé secrète (aussi appelé symétrique car la même clé est utilisée pour crypter et décrypter) étant que dans ce dernier cas, on a besoin d'un canal sûr par où faire circuler une clé.

⁴⁰En cas de vente d'un produit contenant **IDEA**, une licence est à payer à ASCOM TECH.

Fonctionnalités Les principales possibilités de ce logiciel sont l'authentification, le chiffrement et la compression (même si cette dernière est plutôt un effet de bord). Il vient avec une documentation très complète (en anglais), ainsi qu'une courte aide en ligne en français.

Exemples Seules les plus importantes sont décrites ici. Référez-vous à la version en ligne de l'aide avec `pgp -h`, au manuel UNIX avec `man pgp` ou encore à la documentation venant avec ce logiciel.

`pgp -kg` Créer votre propre paire de clés publique et privée.

`pgp -kv` Visualiser votre trousseau (anglais: *keyring*) de clés publiques (anglais: *keyring*). On peut aussi ajouter un nom de fichier de trousseau de clés pour visualiser, par exemples, vos clés privées.

`pgp -ka fichier` Ajouter les clés (en général publiques) de futurs destinataires à votre trousseau de clés.

`pgp -kx nom fichier` Extraire (copier) votre clé publique, celle associée à `nom` pour la communiquer, en la stockant dans `fichier`.

`pgp -e fichier destinataire` Crypter le fichier `fichier` pour le destinataire `destinataire` dont la clé publique se trouve dans votre trousseau de clés (anglais: *keyring*). Le fichier résultant se nommera `fichier.pgp`.

`pgp fichier.pgp` Décrypter le fichier `fichier.pgp` à l'aide de votre clé secrète.

`pgp -s fichier` Signer un fichier avec votre clé secrète pour authentification. On peut aussi détacher la signature. L'extraction se fait avec `pgp fichier`.

Notez qu'en ajoutant a aux paramètres, on peut générer des fichiers transportables par courrier électronique.

Comment échanger des clés Le but d'un serveur de clé (anglais: *keyserver*) est de rendre possible l'échange de clés et de servir comme clavothèque⁴¹ (registre, dépôt, anglais: *repository*) de clés publiques à des fins d'authentification surtout. Il en existe plusieurs⁴² comme, par e-mail `pgp-public-keys@mit.edu`, ou via WWW à l'URL `http://wwwkeys.ch.pgp.net/` ou `http://www.keyserver.net/`.

Notons que toute clé, avant d'être utilisée, et en particulier avant d'être signée, devrait être vérifié (p.ex. par téléphone avec l'empreinte (anglais: *fingerprint*), ou mieux à l'aide d'une carte d'identité).

Plus d'informations Le forum de news `alt.security.pgp` traite de ce logiciel. Il traite également des problèmes de l'auteur avec la Justice des Etats-Unis.

7.7 Impression de fichiers

7.7.1 Soumission de travaux

La commande pour soumettre un fichier à l'imprimante est `lp`⁴³.

7.7.2 Etat de la queue

Pour voir l'état de la queue d'impression, on peut utiliser la commande `lpstat -t` ou `lpq`.

7.7.3 Destructions de travaux

La destruction de travaux (**jobs**) est possible avec la commande `cancel`⁴⁴, en spécifiant le numéro de job d'impression indiqué lors de la soumission de la commande ou obtenu grâce à la commande `lpstat`.

7.7.4 En cas de problèmes

L'utilisateur `lp` vous enverra un **mail** contenant le numéro de job et des informations sur celui-ci ainsi que sur la raison de la panne.

8 Personnaliser son environnement

Cette section traitera des moyens de changer la configuration utilisateur. Un utilisateur non expérimenté ne devrait même pas la lire, mais y revenir lorsqu'il estimera avoir atteint une pratique suffisante pour éviter les erreurs les plus courantes.

⁴¹Mot suggéré par un relecteur qui veut rester anonyme.

⁴²Notre serveur dispose d'un serveur de clé, `keyserv@alphanet.ch`, accessible uniquement depuis l'intérieur de notre réseau.

⁴³parfois `lpr`.

⁴⁴parfois `lprm`. Dans ce cas, on peut tuer tous les jobs en donnant `-` comme paramètre. Notez qu'il faut en général être connecté sur la machine depuis lequel le job a été lancé.

8.1 Qu'est-ce qu'un environnement ?

Un environnement est un ensemble de variables et de fichiers de configuration contenant des états et des options du système. Par exemple, une variable d'environnement peut indiquer que votre terminal est un terminal VT100, une autre que votre **shell** préféré est **tcsh**, et une dernière que vous maintenez un système de manuel personnel qu'il faut consulter lorsque la commande **man** est invoquée.

8.1.1 Variables

Le système utilise des variables comme

TERM type de terminal courant

PATH chemin de recherche des commandes

LD_LIBRARY_PATH chemin de recherche des bibliothèques partagées

MANPATH chemin de recherche des manuels

EDITOR éditeur de texte par défaut

Certaines applications nécessitent également des variables d'environnement pour fonctionner. On peut les afficher indépendamment avec `echo $VAR`, si la variable s'appelle `VAR`.

On peut voir une courte liste des variables avec `env`, ou `set`. On modifie les variables comme suit :

`export TERM=ansi` mettre le terminal à ANSI et exporter pour les programmes appelés

`set correct=all` En shell **tcsh** : mode correction de tout ce qui est entré (commandes et arguments, non recommandé)

`PS1="hello>"` changer le prompt en `hello>` (set est optionnel)

`PATH=$PATH:/usr/local/bin/netpbm` ajouter au sentier de recherche des exécutables un répertoire, pour le shell seulement.

Notez bien que ces modifications sont temporaires (jusqu'à la fin du **shell** courant). Pour les rendre permanentes, il faut les ajouter aux fichiers de démarrage du shell (*voir section 8.1.3*).

8.1.2 Commandes de configuration

La plupart des commandes de configuration sont exécutées au **login**. D'autres commandes sont exécutées à chaque lancement d'un **shell**. Parmi celles-ci, citons les commandes suivantes :

stty pour initialiser les paramètres du terminal (*voir section 5.9.33 page 30*),

mail pour vérifier si vous avez du courrier privé,

quota pour contrôler si vous avez des dépassements de quota,

umask pour définir le mode de création de fichiers ou de répertoires utilisé par défaut. Celui-ci se définit en inversant la valeur désirée (il s'agit bien d'un masque au sens informatique). Par exemple, pour que les fichiers créés le soient au mode 644 et les répertoires au mode 755, on utilisera `umask 022`. En cas de paranoïa, on peut utiliser `umask 077` (*voir section 5.9.7 page 25*). Cette commande doit être insérée dans votre `.cshrc` (*voir section 8.1.3*).

8.1.3 Fichiers de configuration

Les fichiers habituels du système sont cités dans les paragraphes suivants. Notez que certaines applications ont besoin de fichiers de configuration ou de données. La plupart du temps, ceux-ci sont créés automatiquement.

La plupart de ces fichiers commencent par `.` et ils ne sont donc pas visibles avec `ls`, mais avec `ls -a`.

`.bash_profile` Ce fichier contient des séquences d'instructions exécutées à chaque lancement d'un shell de connexion (**login shell**) **bash**.

`.bashrc` Ce fichier contient des séquences d'instructions exécutées à chaque nouveau sous-shell **bash** ouvert (excluant les shells de connexion).

`.bash_logout` Ce fichier contient des commandes exécutées en fin de connexion pour un shell de connexion **bash**.

`.plan` Ce fichier contient des indications personnelles comme par exemple le projet sur lequel vous travaillez et est affiché par la commande `finger` (*voir section 25.2 page 88*). Pour que d'autres utilisateurs puissent y accéder, il faut que son mode (sa protection) soit `rw-r--r--`, soit avec une notation abrégée `chmod 644 .plan`. Cependant, cela ne fonctionnera que si votre répertoire est accessible (mode `x`). Cela se fait également avec une commande comme `chmod +x $HOME` (*voir section 5.9.7 page 25*).

`.forward` Ce fichier, s'il existe, indique au système de **mail** où il faut envoyer les mails qui sont reçus par vous. Cela peut être utile si vous avez plusieurs accès. On peut aussi l'utiliser pour envoyer à plusieurs utilisateurs des copies de mails, ou pour sauvegarder tous les mails entrant dans un fichier, ou encore pour un message de non-présence. Par exemple, pour copier tous les messages privés entrants à la fois dans votre boîte-aux-lettres et dans un fichier, utiliser un fichier contenant :

```
\schaefer, /users/schaefer/Mail/tous_les_mails
```

en supposant bien sûr que le nom de l'utilisateur est bien `schaefer`. Le `\` figurant avant le nom d'utilisateur est là pour éviter une boucle infinie de résolution des forwards provoquant (après un nombre fixé de boucles, dépendant de la configuration du système de mail, typiquement 25) le renvoi du message à l'expéditeur. Faites attention à ne pas créer de boucles de messages en redirigeant sur un système qui, lui aussi, redirige sur le système initial. De plus, vérifiez avec votre administrateur système que le système de **mail** gère correctement les accès concurrents à des fichiers spécifiés dans un `.forward`.

`.rhosts` Ce fichier, s'il existe, autorise l'accès *sans mot de passe* à votre compte pour des utilisateurs locaux ou distants listés dans ce fichier. Il est évident que cette option ne devrait être utilisée que par des utilisateurs expérimentés, en raison des problèmes de sécurité causés. Évitez de mettre dans ce fichier des autorisations pour des sites en dehors de votre organisation sans consulter votre administrateur système.

`.netrc` Ce fichier contient des séquences de connexions automatiques à des sites distants pour transfert de fichiers (voir commande **ftp**). Vu que ce fichier peut contenir des mots de passe, la protection doit être assurée en conséquence par la commande `chmod 600 .netrc`.

Exemple: Voici un exemple de fichier `.netrc` (une ligne vide finale est nécessaire)

```
machine ftp.alphanet.ch
login ftp
password schaefer@alphanet.ch
```

`.signature` Ce fichier peut contenir un court texte qui sera ajouté à tout message sortant du système en **mail** ou **news**.

8.2 Modifier son environnement

Modifier son environnement de façon permanente se fait toujours dans un fichier de configuration exécuté à l'entrée dans le système ou à la création du **shell** (`.bash_profile` ou `.bashrc`). Pour changer une variable, il faut utiliser l'assignation de variables, avec ou sans `export`. Une bonne idée est de regarder les exemples et de tester ses modifications avec un `su - utilisateur`. On pourra toujours quitter en cas de problèmes et reprendre la copie que vous n'aurez pas manqué de faire, vu que si vous lisez cette section, c'est que vous êtes un utilisateur avancé d'UNIX. Changer son **shell** par défaut peut se faire avec la commande **chsh**, mais cela n'est pas recommandé.

Voici un exemple de changement du chemin de recherche des exécutable. On ajoute le répertoire `/usr/local/newbin` au chemin de recherche (anglais: *path*) et on exporte ce changement aux sous-shells et commandes lancés ensuite :

```
export PATH="$PATH:/usr/local/newbin" change le sentier de recherche
```

8.3 Création de scripts

La rédaction de **scripts** est une opération assimilable à de la programmation. Il est donc conseillé de consulter un livre de référence sur le **shell** choisi et de s'inspirer des nombreux exemples disponibles.

8.3.1 Buts

Les scripts (ou *batches*) permettent de regrouper des fonctions complexes dans des fonctions plus simples à appeler, ou d'automatiser des manipulations compliquées ou à heures fixes (cf. **crontab**). Un script tourne toujours dans un **shell** et il est important d'indiquer au système dans quel shell un script donné a été réalisé.

8.3.2 Sélection de l'interprète

La sélection de l'interprète peut se faire de plusieurs façons. La première consiste à faire précéder du nom de l'interprète (la commande) et à donner le nom du script en paramètre. Cette méthode est conseillée lors de la configuration de la commande **crontab** (voir section 5.9.8 page 25).

```
/bin/sh /users/schaefer/scripts/mon_script.sh
```

Il est à noter que l'extension `.sh` n'a aucun rôle dans la sélection de l'interprète.

La sélection de l'interprète peut également se faire en mettant une ligne spéciale au tout début du script (première ligne). Par exemple :

```
#! /bin/tcsh
```

ou

```
#! /usr/local/bin/perl
```

ou encore

```
#! /bin/sh
```

On peut aussi donner le script à exécuter via un **pipe**, comme dans

```
cat script | /bin/sh
```

La dernière méthode consiste à demander à l'interprète courant d'exécuter un script à l'aide d'une commande spécifique, comme par exemple `source nom_de_script` sous **tcsh** ou **csch**, ou l'opérateur `.` sous **sh**.

8.3.3 Choix de l'interprète

Le choix de l'interprète approprié pour une tâche donnée n'est évidemment pas universel. Cependant, les programmeurs **shell** expérimentés font les constatations suivantes :

sh Très adapté pour les petits scripts, qui sont très portables car **sh** existe sur quasiment toutes les plates-formes UNIX, il est extrêmement déconseillé pour les grands scripts ou la gestion d'arithmétique (elle n'est pas incluse dans le langage, il faut utiliser des programmes externes comme `bc`, ou `expr` pour les expressions simples). De plus, certains le considèrent comme peu fiable. En session interactive, ce shell est inutilisable.

bash Ce shell, dérivé de **sh**, possède des fonctions interactives comparables à **tcsh**. Il est très répandu sous environnements **Linux**, notamment en raison de sa compatibilité avec la norme **POSIX**.

csch A cause des nombreuses contradictions dans sa grammaire et les trop nombreuses mises à jour, ce shell ne devrait être utilisé que pour la session interactive, mais même pour cela, il est avantageusement remplacé par **tcsh**.

tcsh Originellement, un excellent shell interactif, il est aujourd'hui avantageusement remplacé par **bash**.

ksh Cette extension normalement compatible avec **sh** est suffisamment normalisée (**POSIX**), et, fait nouveau par rapport à **sh**, supporte les tableaux, et les valeurs numériques dans le langage. Pour la session interactive, relativement déconseillé.

perl Ce langage (non interactif) a l'avantage d'offrir beaucoup de fonctions qui augmentent la vitesse de l'ensemble. On pourrait le comparer à une sorte de **C** interprété. Il offre aussi des possibilités étendues de sécurité, particulièrement en ce qui concerne les fameux **suid scripts**. Malheureusement, beaucoup le considèrent comme étant *write-only* : en effet sa syntaxe est quelque peu primitive.

awk Ce langage (non interactif) sert surtout au traitement de données représentées sous forme de champs textuels.

Il faut noter que même si la création de scripts est souvent plus simple que la programmation en un langage tel que **C**, et parfois plus portable, les scripts sont exécutés de façon très lente en raison surtout des nombreuses commandes externes exécutées. Un compromis entre le temps de développement et le coût à l'exécution doit être trouvé.

8.3.4 Exemple d'un script sous sh

Editez ce fichier et sauvez-le sous `rename.sh`. Entrez la commande `chmod 755 rename.sh`. Ensuite, il suffira de le tester avec, par exemple, `./rename.sh tgz tar.gz` pour renommer tous les fichiers du répertoire courant qui finissent en `.tgz` en `.tar.gz`.

```
#! /bin/sh
# Ce script renomme tous les fichiers dont l'extension est donnée,
# sans le point initial, en premier paramètre du script, en remplaçant
# l'extension par celle donnée, toujours sans point initial, en deuxième
# paramètre. Pour que ce script marche, il faut faire un
# chmod 755 dessus (ou chmod u+x).

if [ $# != 2 ]; then
    echo "$0 fromext toext"
    echo "$0 : bad args."
    exit 2
fi

echo "Je vais renommer les fichiers en .$1 en .$2"
for i in *.$1
do
    NEWFILENAME=`basename $i $1`.$2
    echo "Ancien nom : $i Nouveau nom : $NEWFILENAME"
    mv $i $NEWFILENAME
done
```

Explication Les variables du shell **sh** commencent toutes par le symbole `$`. Ces variables permettent de stocker un état dans un script ou d'améliorer la lisibilité ou la configurabilité de celui-ci. Certaines variables spéciales sont définies implicitement (et ne sont pas directement modifiables par une affectation). Citons, parmi elles :

`$#` Contient le *nombre d'arguments* passés au script. Notez que contrairement au langage C, ce nombre ne contient pas le nom du script lui-même mais seulement les arguments effectifs.

`$*` Permet de représenter tous les arguments, séparés par le séparateur standard (usuellement l'espace).

`$i` Où *i* peut valoir de 0 (nom de la commande courante) à `$#` (dernier argument).

On utilise la commande `basename` : celle-ci permet d'extraire un nom de fichier d'un chemin complet (en enlevant éventuellement une extension spécifiée en premier paramètre au script `$1`). On utilise un **backquote** afin d'assigner à la variable `NEWFILENAME` la sortie de l'exécution de `basename` et non pas la commande elle-même. Enfin la commande `mv` est utilisée afin de renommer le fichier.

Il est d'usage de tester les arguments et s'ils ne correspondent pas à une utilisation normale du script considéré, d'afficher une courte aide ou un message d'erreur.

Notre script affichera donc, en cas de mauvaise manipulation, un message d'erreur indiquant sur la première ligne le format attendu et le message `bad args` (mauvais arguments) sur la seconde.

8.3.5 Exemple de boucle en sh

Première version figée et simple à comprendre :

```
#!/bin/sh
# Ce script exécute 5 fois la commande passée en paramètre.

echo "Je vais exécuter 5 fois la commande $*"
for i in premier 2 3 4 dernier
do
    $*
done
```

Deuxième version un peu plus dynamique :

```
#!/bin/sh
# Ce script prend en paramètre un nombre d'itération ainsi qu'une
# commande et ses arguments. Un argument négatif n'est pas conseillé.

if [ $# -lt 2 ]; then
    echo "$0 number command param ... param"
    echo "$0 : bad args."
    exit 2
fi

NUMBER=$1
# Oublier le 1er argument
shift

while [ $NUMBER -gt 0 ]
do
    $*
    NUMBER=`expr $NUMBER - 1`
done
```

8.3.6 Autres exemples

Cette section illustre quelques exemples de commandes avancées sous **sh**.

Supposez que vous vouliez éditer tous les fichiers qui contiennent une certaine chaîne. Il ne s'agit pas d'éditer des fichiers dont le *nom* serait composé d'une certaine chaîne (ce problème se résoudrait facilement avec une expression régulière (*voir section 5.6 page 20*)). On utilisera pour ce faire l'exécution de commande avec prise de la sortie comme arguments d'une commande (on utilise les **backquotes**, ou accent grave). Notez l'option `-l` de `grep` qui demande de ne lister que les noms des fichiers.

```
vi `grep -l regexp *`
```

Des commande n'utilisant pas d'entrée standard peuvent être utilisées avec la commande `xargs`. Supposons que vous désiriez effacer tous les fichiers contenant une certaine chaîne. Les deux possibilités suivantes sont équivalentes. Une dernière méthode (en spécifiant l'option `-l` à `xargs`) aurait l'avantage de ne pas souffrir de limitation de nombres d'arguments (au prix d'une exécution plus lente, en effet la présence de l'option `-l` force une exécution de `rm` par argument).

```
rm `grep -l regexp *`
grep -l regexp * | xargs rm
```

Mais que se passe-t-il si les noms de fichiers contiennent des espaces ou d'autres séparateurs (tabulateurs, fins de lignes, etc) ? La solution est, du moins avec les outils **GNU**, d'utiliser un argument spécial (souvent `-0`), qui permet d'indiquer aux commandes qui le supportent de travailler en mode chaîne terminée par zéro (le **NUL** du langage **C**). Comme les noms de fichiers ne peuvent contenir ce caractère, l'interprétation n'est plus ambiguë :

```
grep -lZ regexp * | xargs -0 rm
```

9 Aspects de sécurité informatique

9.1 Piratage

On entend par ce terme une des actions suivantes :

1. s'introduire dans un système sans y être autorisé ou en ne respectant pas les limitations d'accès,
2. échanger des logiciels sous une licence ne l'autorisant pas explicitement (en particuliers les logiciels commerciaux).

Cette section ne s'intéressera qu'à la première possibilité. La seconde est évidemment à vos risques et périls (ainsi que bien souvent à ceux de l'organisation qui héberge sans le savoir vos agissements, malheureusement).

9.1.1 En quoi suis-je concerné

Que peut-il se passer si votre compte est piraté ou utilisé à des fins illicites par des tiers ?

Si votre compte est piraté Les risques ne sont pas limités au *monde électronique*. Vous risquez :

- la perte de votre compte,
- des pertes de données,
- la perte de confiance de vos correspondants (faux mails d'injures, etc.),
- le piratage d'autres systèmes ou de comptes à l'aide de votre compte, ce qui amènera le discrédit sur votre organisation et vous-même, sans compter les problèmes judiciaires : votre responsabilité sera probablement reconnue comme engagée.

Pour éviter des problèmes Il faut se rendre compte que vous, en tant qu'utilisateur final, avez intérêt à assurer un minimum de sécurité concernant votre accès, par exemple :

- ne pas partager votre compte avec plusieurs utilisateurs,
- choisir un mot de passe respectant les règles élémentaires (*voir section 9.2*),
- suivre les consignes de votre administration.

Considérations judiciaires Dans la plupart des pays, le piratage de systèmes informatiques ou de logiciels est sévèrement réprimé par l'amende et/ou la prison⁴⁵.

9.2 Mot de passe

Un mot de passe est l'unique moyen d'assurer votre identité. Il est important de le choisir avec soin grâce aux recommandations qui suivent.

9.2.1 Recommandations

Un mot de passe est composé d'au maximum 8 caractères et d'au minimum 6. De plus, il doit contenir des lettres en majuscules, minuscules et des chiffres. Attention, une lettre majuscule n'est pas à entrer comme une lettre minuscule. Il faut éviter d'utiliser des symboles spéciaux, les caractères doivent être pris dans la liste suivante :

- lettres minuscules de a à z,
- lettres majuscules de A à Z,
- chiffres 0123456789,
- éventuellement d'autres symboles tels que `+ * ! - , _ souligné, / slash, \ backslash` et espace.

⁴⁵En Suisse : Code pénal articles 143 et 143bis. Réclusion si but d'enrichissement illégitime. Sinon, puni de l'emprisonnement ou amende. Poursuivi sur plainte.

Il faut éviter de rendre les mots de passe faciles à deviner. Particulièrement, les recommandations suivantes s'appliquent :

- pas de mot figurant dans un dictionnaire quelconque, ni même avec des variations en remplaçant certaines lettres par des chiffres,
- au moins une majuscule et un chiffre,
- le mot de passe ne doit pas comporter de numéros de téléphone, d'adresses ni de données personnelles.

Le mot de passe est *personnel*. En cas de perte, il est en général impossible de le retrouver. Par contre l'administrateur du système peut vous le remplacer.

N'utilisez jamais le même mot de passe sur plusieurs systèmes d'un niveau de sécurité non équivalent ou qui ne sont pas sous la même administration, surtout sur les BBS où le mot de passe est visible pour l'opérateur. Sous UNIX, même l'opérateur ne peut pas connaître les mots de passe (*voir section 9.4 page suivante*).

De plus, sachez que beaucoup de réseaux offrent la possibilité à certains d'intercepter les mots de passe. Dans certains cas, l'accès à votre compte à travers des réseaux publics (p.ex. Internet) est déconseillé. Il est cependant possible d'utiliser des méthodes de chiffrement des sessions (*voir section 20.1.3 page 67*).

Il faut également noter que les caractères # et @ ont une signification spéciale en cas de **login** via **telnet** et ne sont donc pas recommandés dans le mot de passe.

9.2.2 Changement du mot de passe

La commande `passwd`⁴⁶ permet de changer de mot de passe. Elle demandera d'abord l'ancien mot de passe, puis deux fois le nouveau, afin de contrôler que celui-ci n'est pas entré de manière erronée. Il n'y aura pas d'écho. Notez qu'il est possible que cette commande restreigne les mots de passe possibles (en demandant une longueur minimale et un *bon* mot de passe selon des critères divers). Par exemple, la commande peut exiger un mélange de majuscules et de minuscules (qui ne sont pas équivalentes) ainsi que des chiffres.

9.3 Protection de vos données

9.3.1 Pannes

Aucun système informatique n'est fiable de façon absolue. Les pannes peuvent être logicielles ou matérielles. La meilleure façon de se prémunir contre les pannes est d'effectuer des copies à intervalles réguliers sur des supports conservés dans des endroits sûrs (ou au moins distincts). Renseignez vous auprès de votre administrateur système pour connaître la politique de sauvegarde appliquée chez vous.

9.3.2 Limitation d'accès à vos fichiers

L'accès aux fichiers d'un compte dépend des autorisations accordées par son propriétaire. Par défaut UNIX est permissif et autorise la lecture des fichiers dont on n'est pas propriétaire, même chez les autres utilisateurs. Cela peut varier d'un site à un autre suivant la politique locale, mais autoriser la lecture de ses fichiers aux personnes partageant les mêmes machines, des collègues le plus souvent, n'est pas absurde. Cela n'empêche en rien de se créer un répertoire personnel et en interdire l'accès via la commande `chmod`. Le mode de protection des fichiers et répertoires que l'on crée est lié à son `umask` (*voir section 8.1.2 page 46*). Si ce niveau de sécurité ne vous convient pas, les pas suivants peuvent être effectués pour assurer une meilleure protection de vos données :

`cd` revenir au **répertoire par défaut**

`chmod 711 .` n'importe qui peut pénétrer et lire vos fichiers non protégés spécifiquement dans votre répertoire, mais vous seul pourrez en lister le contenu⁴⁷

`chmod 700 * .???*` Tous vos fichiers seront protégés, vous seul pourrez les lire/modifier/exécuter. De même, les sous-répertoires de votre répertoire principal seront protégés.

`chmod +r .plan` Votre activité est donc accessible en lecture par tous, par exemple pour la commande `finger` (*voir section 25.2 page 88*).

`chmod +rx public` Vous autorisez l'accès au répertoire public. D'autres commandes de ce genre peuvent être utiles, par exemple, pour votre **homepage** (*voir section 28.6 on page 95*).

9.3.3 Chiffrement de vos données

La meilleure façon de protéger des données est de ne pas les stocker sur des supports que vous n'avez pas dans votre strict et personnel contrôle. Évidemment, il n'est pas toujours nécessaire d'en arriver là. On peut soit jouer sur la confiance, la sécurité du système informatique, ou rendre inutilisables (sans mot de passe supplémentaire) certaines données. Plusieurs programmes de chiffrement sont disponibles, parmi eux **crypt** et **PGP**, qui offre un niveau de sécurité supérieur et plus de possibilités grâce au système de clés publiques et privées. Notez cependant que l'utilisation de **PGP** nécessite une autorisation dans certains pays (*voir section 7.6.1 page 44*).

⁴⁶Parfois `yppasswd`, si le système utilise **NIS**.

⁴⁷Dans certains cas, ceci peut avoir plus d'inconvénients que d'avantages.

9.4 Compte personnel

Votre compte est *PERSONNEL*. Vous êtes prié de ne jamais laisser quiconque d'autre que vous-même y accéder. Cela est dans votre intérêt. Évitez de même d'entrer le mot de passe en présence d'autres personnes. Ne laissez jamais un terminal non bloqué (utilisez l'option `LOCK SCREEN` de `screen` (voir section 5.9.31 page 29) ou la commande `xlock` sur terminal X11 (voir section II page suivante)). N'écrivez pas votre mot de passe sur une feuille de papier. N'utilisez surtout pas votre mot de passe sur d'autres systèmes, particulièrement sur des **BBS** ou des jeux. Il va sans dire que le respect des consignes indiquées à la section 9.2 ainsi que celles de votre administrateur système est essentiel.

Sous UNIX, il est impossible à l'opérateur de connaître, à partir de la version codée de votre mot de passe, le mot de passe en clair : le système d'exploitation stocke les mots de passe dans un fichier nommé en général `/etc/passwd`. Parfois, et c'est de plus en plus le cas sur les systèmes modernes, même les mots de passe cryptés ne sont pas visibles aux utilisateurs normaux (**shadow passwords**). Même en possédant les mots de passe cryptés, le décodage peut prendre un certain temps pour des mots de passe bien choisis et si ceux-ci sont changés souvent, le risque est très faible. Par contre si les mots de passe cryptés sont visibles pour tous et que les mots de passe sont des mots dérivés du dictionnaire, le décryptage se fera en quelques heures !

La fonction système qui crypte les mots de passe est basée sur une version modifiée de **DES**. L'utilisateur entre son mot de passe qui est utilisé comme clé pour crypter un bloc de 64 bits de zéros. Les sept premiers bits de chaque caractère sont extraits pour former 56 bits (d'où la limitation à 8 caractères du mot de passe). Les autres 8 bits sont utilisés comme parité. La table de chiffrement de **DES** est ensuite perturbée par une graine aléatoire qui sera stockée dans les deux premiers caractères du mot de passe crypté (son but est de générer deux chaînes différentes même si deux utilisateurs ont le même mot de passe, afin d'augmenter l'espace de recherche). Ensuite, on appelle 25 fois **DES**. Le résultat est ensuite converti en une suite de caractères. Cette fonction est similaire à une fonction de hachage et n'a pas d'inverse. On peut évidemment parcourir tout l'espace de recherche, ce qui est très long, ou se borner à des dictionnaires et à des règles, comme le logiciel `crack`, ce qui ne sert à rien si les mots de passe sont bien choisis. L'idéal est bien sûr de cacher également les mots de passe cryptés.

Ceci n'empêche nullement un programme simulant une connexion erronée ou un **trojan horse** de vous faire entrer le mot de passe en clair, ou plus simplement que vous entriez votre mot de passe sur un système où le mot de passe est visible de l'opérateur, ce qui est le cas pour presque tous les **BBS**. N'oubliez pas, de plus, que la transmission de votre mot de passe sur un réseau peut le faire intercepter (sauf si celui-ci utilise **Kerberos** ou **Secure-RPC**). C'est le cas pour **Internet** en général (voir section 20.1.3 page 67). De plus, l'utilisation d'un système de fenêtrage comme **X11** sans sécurité élémentaire peut poser certains risques, en particulier depuis les terminaux X (voir section 14 page 57).

Deuxième partie

Systeme de multifenêtrage X11

10 Introduction

Le système de multifenêtrage X11, aussi appelé X Window System, a été développé au MIT et est aujourd'hui l'interface graphique de la quasi totalité des ordinateurs sous UNIX.

Il permet de rendre totalement indépendants plusieurs aspects d'un système de fenêtrage et ainsi de développer des applications clientes des services de l'interface, indépendamment du matériel, du système d'exploitation, de la localisation de l'interface et de l'aspect de celle-ci (**look-and-feel**).

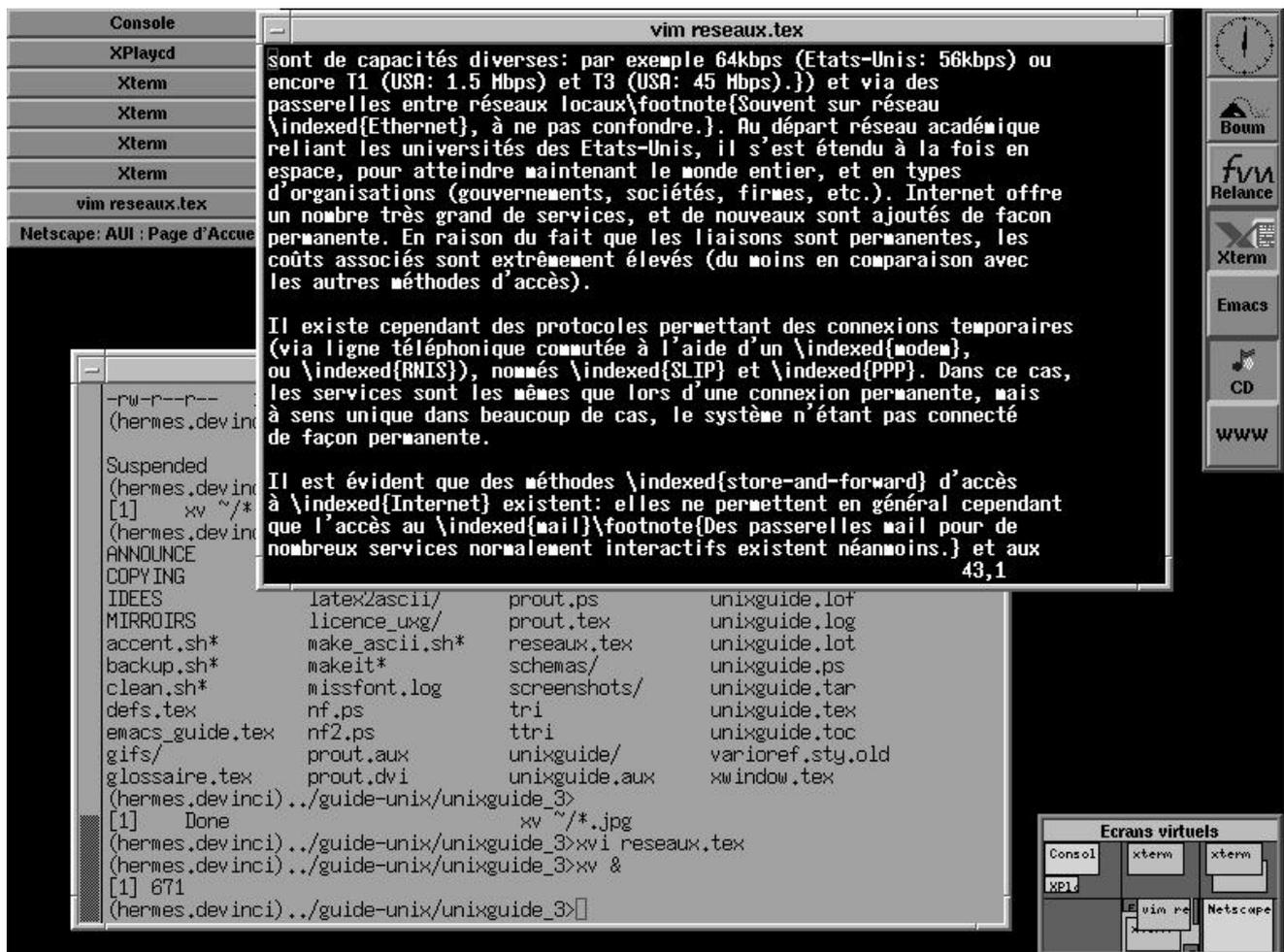


FIG. 6 – Exemple d'écran X11 géré par le gestionnaire de fenêtre fwm

Ce système rend donc également possible l'affichage de clients en provenance de plusieurs machines sur un écran unique (que l'on appelle le serveur). Par-dessus X11 viennent ensuite se greffer des **window-managers** ou gestionnaires d'interface qui déterminent l'aspect et le mode d'interaction (type de fenêtres, de gadgets, de boutons, couleurs, ...). Ces gestionnaires sont des clients spécifiques. Il existe plusieurs gestionnaires d'interface graphique, comme par exemple Motif, OpenWindows, etc.

Un autre intérêt du protocole X11, mis à part le fait qu'il est supporté sur un nombre imposant de plates-formes, est aussi que ses sources sont disponibles sous une licence peu limitative. Cependant, il est évident que certains clients (particulièrement les gestionnaires d'interface, comme OSF/Motif) ne sont pas disponibles de cette façon.

Une bonne introduction à X11 se trouve dans [16].

La plupart des systèmes implémentent au moins X11 Release 4, certains X11 Release 5, et depuis mai 1994, la version X11 Release 6 est disponible.

11 Terminologie

La terminologie X11 peut sembler contradictoire ou du moins insensée au profane. En effet, un gestionnaire d'écran, comme une application, sont considérés comme des **clients**, et l'écran *physique* est lui considéré comme un **serveur**. En fait, il est assez normal que la couche offrant des services graphiques utilisés par les clients distants se nomme serveur.

Le serveur est donc la machine sur laquelle l'affichage proprement dit s'effectue. Tout le reste n'est que clients, y compris un gestionnaire d'interface (**window-manager**). Votre ordinateur personnel (où s'affichent les différents clients distants) est donc appelé un serveur X. Les deux parties (clients, serveur) peuvent très bien se trouver sur un même système (station de travail par exemple).

12 Protocole

X11 fonctionne normalement à l'aide du service **TCP** de **TCP/IP** dans la région du port 6000⁴⁸. Les clients se connectent donc sur le serveur grâce à **TCP/IP** ou un médium local : sous UNIX il est très courant que la connexion locale se fasse non pas par **TCP/IP**, mais par **socket UNIX**, ce qui est bien souvent plus efficace et permet de supprimer l'écoute externe.

Le protocole **XDMCP** permet à un serveur de connaître tous les systèmes sur lequel il peut se connecter via un **login** graphique UNIX.

La plupart des terminaux (serveurs) X, comme des stations de travail qui démarrent sous X sans mode console, tournent à l'aide de xdm. Ils supportent en général l'authentification par clé (voir section 14 page 57).

13 Configuration

La configuration d'une session se fait à travers divers fichiers de configuration et commandes UNIX. Comme avec **SLIP**, cette configuration est très dépendante de votre réseau local.

13.1 Fichiers de configuration

La plupart des applications sous **X11** utilisent des entrées dans un fichier de configuration standard `.Xresources` ou encore `.Xdefaults`. On redéfinit le clavier, si cela est nécessaire, dans `.Xmodmap`. Le démarrage de la session s'effectue en général automatiquement depuis `.xsession` via xdm ou encore manuellement depuis `.xinitrc` à l'aide de la commande `xinit`. De plus, la configuration du **window-manager** se trouve dans un fichier du type `.twmrc`, `.mwmrc`, `.tvtwmrc` ou encore `.fvwmrc` suivant le type de gestionnaire (`twm`, `mwm`, `tvtwm` ou `fvwm` qui sont respectivement le gestionnaire X standard, le gestionnaire **Motif** et le gestionnaire d'écrans virtuels.)

La variable d'environnement `XAPPLRESDIR` peut être positionnée pour pointer sur un répertoire contenant, pour chaque application (ou objet X11), sous forme d'un fichier, les définitions de ressources par défaut.

Au niveau du système, bien souvent des configurations et ressources par défaut sont stockées dans `/etc/X11/Xmodmap` et dans les répertoires `/etc/X11/Xresources/` et `/usr/X11R6/lib/X11/app-defaults/`. Des environnements comme **KDE** influencent les paramètres par défaut (anglais: *resources*) via le protocole X standard, mais en utilisant leurs propres bases de configuration.

13.1.1 Exemple d'un fichier `.xinitrc`

```
#! /bin/sh

# Ajout des définitions utilisateurs
xrdb -merge $HOME/.Xresources
# Changement de clavier
xmodmap $HOME/.Xmodmap
# Lancement de quelques clients en spécifiant leur géométrie,
# que l'on peut obtenir entre autres avec xlsclients
xclock -geometry 95x82+926+0 &
xload -geometry 109x84+813+0 &
xterm -geometry 79x9+0+0 &
xterm -geometry 79x25+0+172 &
# Chargement d'une image de fond (il y a aussi xloadimage ou xv)
xsetroot -bitmap /users/schaefer/bitmaps/darkgrey.xbm
# Client spécial: window-manager virtuel
tvtwm
```

13.1.2 Exemple d'un fichier `.Xmodmap`

```
! Définition de quelques touches pour le clavier suisse romand
! sous XFree 86 Linux 3.1.1 sous clavier PC.
keycode 10 = 1 plus bar
keycode 11 = 2 quotedbl at
keycode 12 = 3 asterisk numbersign
```

⁴⁸Le port 6000 est le port de base de l'écran 0. Les écrans suivants se trouvent à 6000 + x. Voir l'analogie avec `DISPLAY=vulcan:0.0`.

```

keycode 13 = 4 ccedilla
keycode 16 = 7 slash bar
keycode 17 = 8 parenleft cent
keycode 20 = apostrophe question apostrophe
keycode 21 = asciicircum grave asciitilde
keycode 22 = BackSpace
keycode 34 = egrave udiaeresis bracketleft
keycode 35 = diaeresis exclam bracketright
keycode 47 = eacute odiaeresis
keycode 48 = agrave adiaeresis braceleft
! Alternative commentée
!keycode 49 = paragraph degree
keycode 49 = grave asciitilde
keycode 51 = dollar sterling braceright
keycode 94 = less greater backslash
keycode 113 = Mode_switch
! Définition des touches ALT: ALT droite est ALT ...
clear mod1
add mod1 = Mode_switch
! mais ALT gauche est META ...
keycode 64 = Meta_L Alt_R
clear mod2
add mod2 = Meta_L
! Control gauche devient Compose (pour les caractères accentués)
keycode 109 = Multi_key
clear control
add control = Control_L

```

13.1.3 Exemple d'un fichier .twmrc

On peut modéliser exactement comme on le souhaite son gestionnaire de fenêtre, indiquer ce que fait chaque bouton de la souris dans chaque situation, choisir les fontes, les couleurs, l'action des touches fonctions etc... Chaque gestionnaire a un petit quelque chose dans la présentation qui lui est propre. Voici un exemple de fichier .twmrc mais les gestionnaires plus d'actualité sont plutôt les variations sur fvwm dont les fichiers de lancement .fvwmrc sont très bien documentés.

```

#
# fichier de lancement .twmrc
#

#WarpCursor
IconFont "vg-20"
TitleFont "vg-20"
MenuFont "spc08x16v"
IconRegion "100x1500+0+100" North West 10 10
BorderWidth 4
# ne déplace que le cadre de la fenêtre si l'option suivante est commentée
OpaqueMove
# place aléatoirement les nouvelles fenêtres si l'option suivante est activée
#RandomPlacement
Color
{
    TitleForeground "gray73"
    TitleBackground "black"
    BorderColor "brown"
    BorderTileBackground "grey"
    MenuForeground "darkorchid"
    MenuBackground "light blue"
    MenuTitleBackground "honeydew"
    MenuTitleForeground "MediumBlue"
    IconForeground "gray73"
    IconBackground "black"
}

```

```
#Button = KEYS : CONTEXT : FUNCTION
#-----
Button1 =      : root      : f.menu "button1" # clic de gauche
Button2 =      : root      : f.menu "button2"
Button3 =      : root      : f.menu "button3"
Button1 = s    : window    : f.menu "button1" # clic avec SHIFT
Button2 = s    : window    : f.menu "button2" # mais dans un fenetre
Button3 = s    : window    : f.menu "button3"
button1 = m    : window    : f.raise          # avec la touche META
button2 =      : icon      : f.move
button3 =      : icon      : f.move
Button1 =      : title     : f.raise
Button2 =      : title     : f.lower
Button3 =      : title     : f.move
"F1" =        : window    : f.iconify    # la touche F1 sur une fenetre
"F1" =        : icon      : f.iconify    # l'iconifie et vice-versa
"F2" =        : window    : f.raise
"F2" = c      : window    : f.lower
"F3" =        : window          : f.lower
"F3" = c      : window    : f.destroy
"F3" = c      : icon      : f.destroy
"F4" =        : window    : f.zoom
"F7" =        : window    : f.zoom
"F8" =        : window    : f.fullzoom
```

```
IconDirectory ".icons" # quels dessins choisir pour les icones
```

```
Icons
```

```
{
  "xterm"      "vt100.bit"    # obtained from IconDirectory
  "working"    "vt100.bit"
  "console"    "xterm.icon"
```

```
}
```

```
NoTitle      # pas de barre titre à ces fenêtrés svp
```

```
{
  "xbiff"      # indique s'il y a du courrier
  "xpostit"
  "TWM Icon Manager"
  "oclock"    # horloge
  "xlogo"
```

```
}
```

```
menu "button1"
```

```
{
  "Machine"      f.title
  "Xterm"        !"xterm +vb -sb -sl 4000 -T Xterm -n Xterm -bg grey73 &"
  "-rlogin-"      !"
  "lion"         !"xterm -sb -sl 4000 -name lion -T lion -e llogin lion &"
  "lynx"         !"xterm -sb -sl 4000 -name lynx -T lynx -e llogin lynx &"
}
```

```
menu "button2"
```

```
{
  "Window Ops" f.title
  "(De)Iconify" f.iconify
  "Move Window" f.move
  "Resize Window" f.resize
  "Raise Window" f.raise
  "Lower Window" f.lower
  "Destroy Window" f.destroy
```

```

"Refresh" f.refresh
"Zoom"          f.zoom
"Full zoom"     f.fullzoom
"Circle up"     f.circleup
"Circle down"  f.circledown
"Focus"        f.focus
"Focus on Root" f.unfocus
"Forced move"   f.forcemove
"More window ops." f.menu "More ops."
}
menu "More ops."
{
"Reread .twmrc" f.twmrc
"Kill twm" f.quit
"twm Version" f.version
"Hide Icon manager" f.hideiconmgr
"Show Icon manager" f.showiconmgr
"Show display:"      !"echo $DISPLAY > /dev/console"
}

```

```

menu "button3"
{
"Outils" f.title
"Courrier" !"xterm +sb +t +cb -name Elm -T Elm -e elm &"
"Nouvelles" !"xterm -e slrn -bg tan &"
"Xftp"      !"xftp -background tan &"
"Xarchie"   !"xarchie -host archie.univ-rennes1.fr &"
"Xmosaic"   !"xmosaic &"
"Calculatrice" !"xcalc -stipple -geometry +900+150&"
"Fond : Lune"  !"xphoon &"
"Fond : Meteo"  !"temps &" # ceci est un script qui affiche la carte météo
"Fond normal"  !"xv -clear; xstdcmap -delete default"
"Xlock"       !"xlock -mode blank &"
}

```

13.2 Commandes

À part les commandes de sécurité (*voir section 14*), on a les commandes suivantes :

xmodmap Cette commande permet de reconfigurer le clavier.

xrdb Cette commande permet de modifier ou de lire l'état de la configuration courante (les ressources, anglais: *resources*).

xlsclients Cette commande permet de connaître les clients actifs et leur positionnement à l'écran.

xinit Cette commande permet en général de lancer le système X11 manuellement en cas de connexion en mode console ⁴⁹.

xev Cette commande est très utile pour déterminer les codes de touches afin de reconfigurer un clavier.

14 Sécurité

14.1 Introduction

Le système X11 étant distribué, les clients peuvent donc venir de n'importe quelle machine du réseau, et sur ces machines de n'importe quel utilisateur. La couche de sécurité de X11 a pour but de n'autoriser l'accès qu'à des machines ou des utilisateurs spécifiques. Pour ce faire, on a deux possibilités, même si parfois les constructeurs n'en implémentent qu'une :

xhost l'autorisation se fait sur la base de l'**adresse IP** de la machine cliente. Cela signifie que n'importe qui se connectant sur cette machine peut contrôler l'écran du serveur. Cette méthode est donc à proscrire en général.

xauth l'autorisation se fait sur la base d'une clé (anglais: *cookie*), qui peut être combinée avec un protocole de partage de fichiers en réseau comme **NFS**. L'autorisation est donc au niveau de l'utilisateur lui-même, ou du moins de la connaissance d'une clé d'accès. Malheureusement, les méthodes à clés révélées ne sont pas encore disponibles librement (voir **PGP**, **Kerberos**, **Secure-RPC**).

⁴⁹il existe aussi startx sur Linux

Il faut noter que ces commandes ne peuvent s'exécuter que sur le serveur, soit sur l'ordinateur qui exporte l'accès à l'écran (p.ex. un terminal X ou un ordinateur personnel qui tourne un serveur X et affichera les fenêtres d'applications distantes).

14.2 xhost

Cette commande est très simple et demande en général un paramètre composé d'un nom de machine (ou **adresse IP**) précédé d'un symbole indiquant si l'on veut autoriser (+) ou refuser (-) l'accès. Si l'on spécifie + ou - seul comme paramètre, cela signifie respectivement autoriser tout accès ou refuser tout accès, pour les machines qui n'ont *pas* été autorisées spécifiquement (cela signifie que la séquence `xhost +vulcan` suivie par `xhost -` autorise tout de même `vulcan` à accéder à l'écran). L'accès ne vaut que pour la session courante.

Ainsi en indiquant au client (`vulcan` par exemple) quel est le serveur X via la commande `setenv DISPLAY amiga1:0.0` si le serveur est `amiga1`, `vulcan` peut lancer des applications X sur la fenêtre d'`amiga1`. On peut afficher la liste d'accès actuelle avec `xhost` seul.

14.3 xauth

Cette commande est légèrement plus complexe en ce sens qu'il faut l'exécuter à la fois sur le serveur et sur chacune des machines susceptibles de se connecter en copiant très exactement la clé d'accès. Cependant, vu que l'information d'accès est stockée dans le système de fichiers (ce qui implique que son accès soit sûr), les états sont permanents.

Le choix de la clé (anglais: *cookie*) est important. Dans les exemples, on utilisera `1122334455`, ce qui est un mauvais choix. Certains serveurs (notamment en cas de **login** graphique via `xdm` sur la console) génèrent automatiquement une clé à chaque début de session. Dans ce cas, il suffit de faire un `xauth list` puis un `xauth add`.

Une façon très simple d'autoriser un site distant à accéder à votre écran est d'extraire de la commande `xauth list` la clé correspondant à la machine locale et d'exécuter sur la machine distante un `xauth add` suivi de la clé récupérée. On peut aussi utiliser les options `extract` et `merge` de la commande `xauth`.

Notons également que si votre but est juste d'autoriser le **root** de la machine locale à accéder à votre écran (p.ex. pour lancer un client X depuis un `su`⁵⁰), il vous suffit de repositionner la variable d'environnement `XAUTHORITY`, comme suit :

```
export XAUTHORITY=/users/schaefer/.Xauthority.
```

15 Etablissement d'une session

Dans ce paragraphe on supposera évidemment qu'une session **TCP/IP** est en cours. On donne l'exemple d'une configuration à base de **xauth**. Notez que le maniement de `xauth` peut être simplifié grâce à **rsh** (ou **ssh**) et au système de **pipe** si les deux systèmes concernés tournent sous UNIX.

On peut également lancer un **login** graphique en demandant la configuration de l'outil **xdm**.

15.1 De votre côté (affichage, serveur)

Il faut déterminer la clé d'accès à l'écran (en règle générale générée par le gestionnaire de connexion, comme p.ex. **xdm**) en l'affichant avec `xauth list`. Dans certains rares cas (comme dans l'exemple qui suit, il faut inventer soi-même une clé).

```
<amiga1> schaefer:/d2/users/schaefer> xauth
Using authority file /users/schaefer/.Xauthority
xauth> add vulcan.alphanet.ch:0 MIT-MAGIC-COOKIE 1122334455
xauth> exit
```

15.2 Du côté UNIX (client)

Il faut tout d'abord indiquer au système où doit se faire l'affichage. La variable d'environnement `DISPLAY` est utilisée pour ce faire, et on l'initialise, par exemple, avec

```
setenv DISPLAY amiga1:0
```

si **amiga1** est le nom de votre machine.

Ensuite, il faut reporter la clé définie du côté du serveur X via **xauth** pour autoriser l'accès à votre écran.

```
<vulcan> schaefer:/d2/users/schaefer> xauth
Using authority file /users/schaefer/.Xauthority
xauth> add amiga1.alphanet.ch:0 MIT-MAGIC-COOKIE 1122334455
xauth> exit
```

⁵⁰Si vous utilisez `su -`, les variables d'environnements et le répertoire par défaut ne seront pas transmis, y compris `$DISPLAY`.

15.3 Automatiser

15.3.1 SSH

Une méthode plus récente est d'utiliser **SSH** pour se connecter à un système distant (*voir section 26.6 page 90*). La connexion est alors cryptée, et l'établissement de celle-ci peut également servir pour différents **tunnels**, comme par exemple le protocole **X**.

```
schaefer@ecran:~% ssh -X client-x
schaefer@client-x:~% echo $DISPLAY; xauth list
client-x:10.0
client-x.alphanet.ch:10 MIT-MAGIC-COOKIE-1 0baec7a965c6ce5363a272
client-x/unix:10 MIT-MAGIC-COOKIE-1 0baec7a965c6ce5363a272
```

En fait, un serveur **X** virtuel, numéro 10 dans ce cas, est créé, et le client distant passe via ce tunnel crypté, le serveur **sshd** sur le serveur **X**, et les données sont ensuite envoyées comme d'habitude au **socket UNIX**.

Avantages :

- connexion cryptée, authentifiée par **SSH**
- transmission transparente et automatique des paramètres
- écoute sur un port **TCP** sur le serveur **X** non nécessaire.

15.3.2 Script rsh

Il est possible d'automatiser le tout sur deux systèmes UNIX. On utilisera le **script** suivant :

```
#!/bin/sh
#
# Ce script peut parfois être avantageusement remplacé par
# la commande xon (vérifiez avec votre manuel).
# Il fait les suppositions suivantes:
#
# 1. vous spécifiez sur la ligne de commande, en paramètre à ce
# script le nom de la machine distante (et éventuellement le
# nom d'utilisateur distant s'il est distinct précédé par -l)
# sur lequel vous avez configuré correctement le fichier .rhosts
# 2. les paths des exécutable sont ceux spécifiés au début du
# script
# 3. les conventions de nom d'écran et de machine sont respectées.
# Elles supposent entre autres que le nom de machine est dans $HOSTNAME
# et nom pas dans $DISPLAY qui ne contient que :0.
# 4. le shell distant est supposé être sh, sinon changer
# l'assignation de variables.
# 5. enfin le système d'authentification est supposé être xauth, ce
# qui est en général vrai pour les sessions lancées par xdm.

LOCAL_DISPLAY="${HOSTNAME}${DISPLAY}"; export LOCAL_DISPLAY
REMOTE_XAUTH=/usr/bin/X11/xauth; export REMOTE_XAUTH
REMOTE_XTERM=/usr/bin/X11/xterm; export REMOTE_XTERM

# Copie de la clé d'authentification
xauth extract - $LOCAL_DISPLAY | \
  rsh $* \(\export DISPLAY=$LOCAL_DISPLAY \; \
  $REMOTE_XAUTH merge -\)

# Lancement d'un xterm
rsh $* \(\export DISPLAY=$LOCAL_DISPLAY \; \
  $REMOTE_XTERM\) \&
```

16 Applications sous X

Les répertoires **X11** où se trouvent des exécutable standards se trouvent souvent dans `/usr/bin/X11`. Déplacez-vous dans ce répertoire avec `cd` car il s'agit peut-être d'un **link** (français : lien). En remontant ensuite, vous vous trouverez dans la base du système **X11**.

De nombreuses applications existent pour l'interface **X11**, il serait difficile de les lister toutes ici. De plus, il est évidemment probable que l'application que vous recherchez ne soit pas disponible sur votre machine. Dans ce cas, il peut être nécessaire de chercher l'application correspondante via **FTP** et ensuite de la compiler. Il est également évident que des applications commerciales comme **FrameMaker** ne sont pas disponibles gratuitement. Le statut de **Netscape** est quelque peu curieux.

Il faut également savoir que grâce à l'émulation de terminal **xterm** il est très possible de lancer toutes les applications **VT100** semi-graphiques dans une fenêtre, sans possibilité d'interagir avec le programme (autrement que par **copy-and-paste**⁵¹). L'une des caractéristiques intéressantes d'un **xterm** se trouve dans ses menus (utiliser la touche **CTRL** et appuyer sur un des boutons de la souris). On peut ainsi sélectionner la taille de la police et d'autres propriétés. Certains logiciels comme **Emacs** (voir section 6 page 33) utilisent des fonctionnalités de **X11** automatiquement.

Voici une courte liste :

– Fonctions de base :

xterm support de terminal virtuel.

xedit éditeur de texte simpliste.

xfilemanager gestionnaire de fichiers.

xfm gestionnaire de fichiers (voir la figure 7).



FIG. 7 – Fenêtre typique xfm sous X11

xclock et toutes les variantes d'horloges : **oclock**, **daliclock**...

xlock pour bloquer son écran lorsqu'on s'absente.

– Pour voir :

ghostview visualisation de fichier PostScript (voir la figure 8 page suivante).

⁵¹Sélection avec le bouton de gauche, **paste** avec le bouton du milieu. Certaines interfaces comme **KDE** demandent parfois l'utilisation d'un raccourci-clavier ou d'un menu pour la fonction de sélection (**copy**)

Ghostview, version 1.5

JDE4.dvi
Wed Sep 10 16:43:28 1998
(431, 425)

File 1
0
1 <
2
Page 3
Magstep 4
Orientation 5
Media 6

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

1. Introduction.

This paper deals with the Spatially Homogeneous Boltzmann equation

$$\begin{cases} \frac{\partial f}{\partial t}(t, v) = Q(f, f)(t, v) & \text{on } (0, +\infty) \times \mathbb{R}^3, \\ f(0, v) = f_0(v) & \text{on } \mathbb{R}^3, \end{cases} \quad (1.1)$$

where $f(t, v)$ is a non negative function which describes the time evolution of the distribution of particles which move with velocity v . In the right hand side, $Q(f, f)$ is the so-called collision operator,

$$Q(f, f)(v) = \int_{\mathbb{R}^3} \int_{\mathcal{S}^2} (f' f'_1 - f f_1) B(\theta, |v - v_1|) d\omega dv_1. \quad (1.2)$$

Here $f = f(v)$, $f_1 = f(v_1)$, $f' = f(v')$ and $f'_1 = f(v'_1)$, and v' and v'_1 are the velocities after the elastic collision of two particles which had the velocities v and v_1 before the encounter. One parameterization of these velocities is

$$\begin{cases} v' = \frac{v + v_1}{2} + \frac{|v - v_1|}{2} \omega, \\ v'_1 = \frac{v + v_1}{2} - \frac{|v - v_1|}{2} \omega, \end{cases} \quad (1.3)$$

where ω is a unit vector of the sphere \mathcal{S}^2 (see figure 1 at the end of the paper). In (1.2), θ is the angle between $v - v_1$ and $v' - v$. A different parameterization is given by

$$\begin{cases} v' = v + ((v_1 - v) \cdot \Omega) \Omega, \\ v'_1 = v_1 - ((v_1 - v) \cdot \Omega) \Omega. \end{cases} \quad (1.4)$$

With this parameterization, the collision operator still is of the form (1.2) with $d\omega$ replaced by $d\Omega$, except that now the kernel B takes the form $2B(\theta, |v - v_1|) \cos(\theta)$. This takes into account also that the second parameterization implies a double covering of the sphere from the first parameterization.

The precise form of the kernel B depends on the physical properties of the gas that is being studied. Here we consider the case of so-called hard potentials, and therefore

$$B(\theta, |v - v_1|) = b(\theta) |v - v_1|^\beta, \quad \text{with } \beta \in (0, 2]. \quad (1.5)$$

In this case b is even and continuous in $]-\pi/2, \pi/2[$. Furthermore, we assume that b satisfies Grad's angular cut-off condition, namely that $b \in L^1(]-\pi/2, \pi/2[)$. This obviously holds for the elastic spheres, but if B is derived from the interaction by an inverse power law, the integrability condition does not hold unless b is truncated in some way.

The main concern of this paper is the proof of the existence of a unique solution to (1.1) with minimal assumptions on the initial data, and moreover of a convergence result for a suitable

FIG. 8 – Fenêtre typique ghostview sous X11

acroread visualisation de fichier PDF.

xdvi visualisation de fichiers **DVI** produits par le logiciel **T_EX**.

xv afficheur d'images (voir la figure 9 page suivante) en **shareware**⁵².

⁵²L'image date un peu, je suis enregistré.

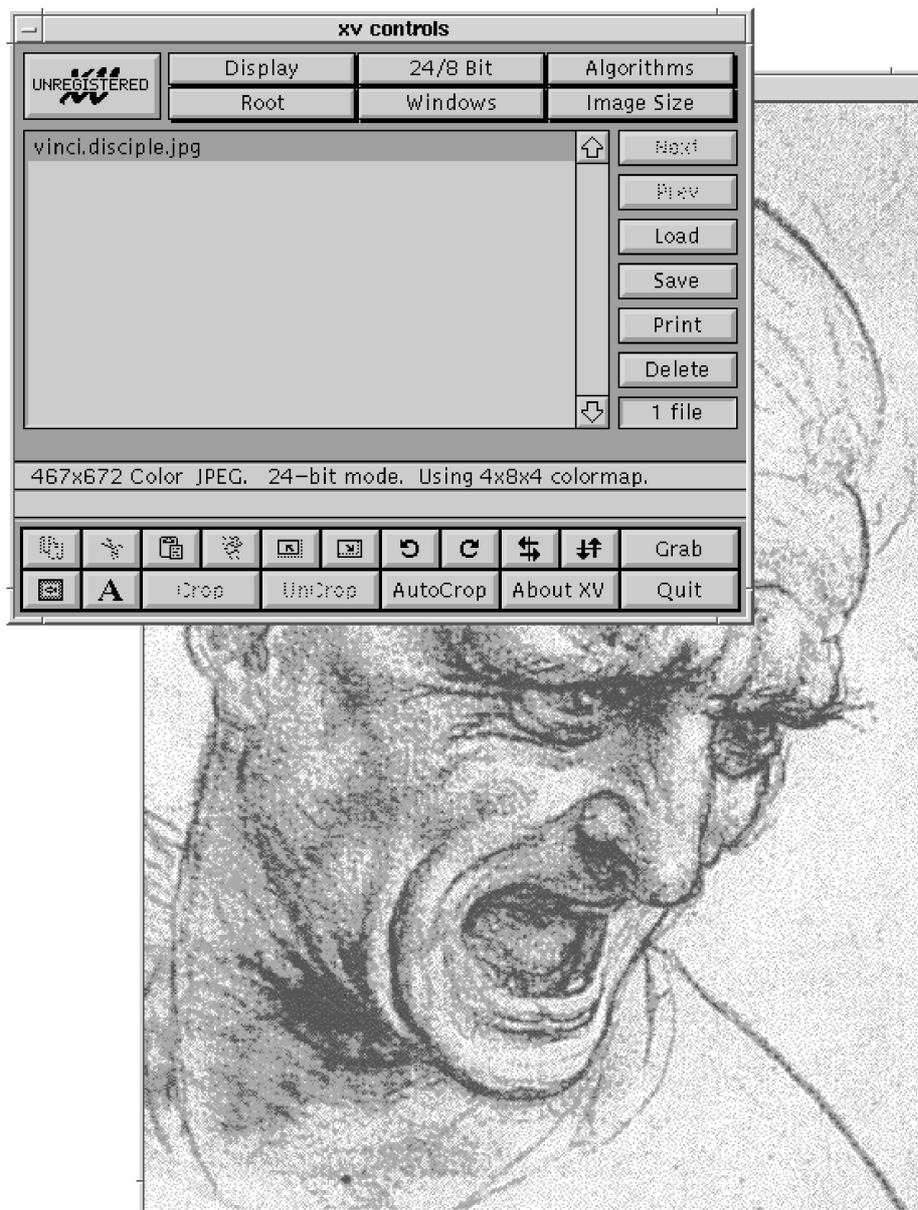


FIG. 9 – Fenêtre typique xv sous X11

ImageMagick visualisation et conversion d'images.

– Lié au réseau :

Mosaic, **Netscape**, **Arena** interfaces graphiques permettant l'accès au système hypertexte distribué **WWW**.

xarchie recherche de fichiers sur les serveurs **FTP** (voir la figure 16 page 85).

xbiff indicateur de nouveau courrier électronique (voir section 22 page 71).

xmh lecteur de **mail** (voir section 22 page 71).

– Graphisme :

xfig logiciel de dessin vectoriel.

xpaint logiciel de dessin.

xmorph morphing.

gnuplot, **PlotMTV** génération de graphes.

– Jeux (un très bref échantillon) :

xboard jeu d'échecs (interface à **gnuchess**).

xbill Le vilain Bill G. veut mettre Windows sur toutes les machines, sauvez-nous !

xrisk jeu de stratégie multi-utilisateur en réseau.

xdoom jeu de massacre 3D.

Les pages **man** donnent des informations quant aux **resources** et aux arguments des programmes : citons par exemple `-geometry` et `-font`.

17 Logiciels TCP/IP et X11 du côté serveur (affichage)

17.1 Introduction

En général, les logiciels permettant l’affichage (à travers un réseau éventuel) de clients X sur un serveur sont composés de plusieurs couches. La première couche est la couche de transport, par exemple **TCP/IP**. La seconde est la couche de gestion de l’écran proprement dite, le serveur. Ces deux couches (qui peuvent être combinées en un seul logiciel) forment le minimum pour utiliser **X11**. Certains systèmes offrent également la possibilité de gérer les interactions avec l’utilisateur de façon locale avec un logiciel nommé **window-manager**. Dans le cas contraire, cette couche peut fonctionner à distance.

17.2 UNIX

Les systèmes UNIX, lorsqu’ils sont livrés avec un écran graphique, le sont généralement également avec un serveur X autorisant l’affichage. Ce serveur est généralement nommé X. De plus, un client **window-manager**, tel que `twm`, `tvtwm`, `olwm`, `olvwm`, `mwm`, `ctwm`, ou encore `fvwm` et des applications sont également disponibles.

Par exemple, le système **Linux**, disponible pour PC, permet de faire tourner **XFree86**, implémentation libre de X Window.

17.3 Amiga

Au niveau de la couche **TCP/IP**, il y a actuellement trois solutions. La première est le logiciel du domaine public **AmiTCP**, la deuxième le logiciel commercial développé par **COMMODORE AS-225**, qui n’est plus vendu (ou alors par **AMIGA TECHNOLOGIES**), et la troisième la couche de support vendue par **GFXBASE**.

Au niveau de la couche serveur, il existe actuellement deux solutions, une domaine public (**DaggeX**) et une commerciale (**Gfx-Base**). Concernant la compatibilité avec la couche **TCP/IP**, il est recommandé de tourner DaggeX avec AmiTCP et GfxBase avec AS-225 ou la solution GfxBase, même si d’autres possibilités existent.

Au niveau de la couche **window-manager**, les deux solutions serveurs ci-dessus sont livrées avec quelques gestionnaires habituels.

A noter l’incompatibilité totale du **window-manager** de DaggeX avec les processeurs 68000, et les crashes assez courants avec ce logiciel, au niveau serveur, de nouveau sur 68000. Notons cependant que les versions récentes de DaggeX (qui sont les dernières, le développeur ayant perdu ses sources) sont beaucoup plus fiables.

17.4 Macintosh

Produit	coût	compagnie	version
MacX	??	Apple Computer Corp.	X11R5
eXodus	\$295	White Pines Software	X11R5

Notez que le logiciel **MacTCP** est nécessaire pour toute application TCP/IP. Il est livré avec le logiciel système dès la version 7.5. C’est un logiciel commercial.

XFree86 a été porté sur la plateforme **MacOSX**, permettant soit une ouverture de fenêtre dans l’interface graphique **Aqua**, soit un mode plein-écran. Il s’agit de logiciel libre.

17.5 MS-DOS

Les produits suivants sont disponibles⁵³ :

Produit	coût	compagnie	version
Xvision	\$395	VisionWare Soft, Inc	X11R5
PC-Xware	\$545	NCD, Inc.	X11R5
PC DECwindows	??	DEC	X11R4
PC Xsight	??	Locus Computing	X11R4
Micro X-Lite	\$75	StarNet Comm. Corp.	X11R4
X Appeal	\$350	Xtreme	X11R5
Xoftware	??	AGE Logic	X11R4
PC X-Kit	\$249	XLink	X11R5

⁵³voir la FAQ **pcnfs**.

Je ne sais pas si des logiciels supplémentaires (par exemple de couche TCP/IP) sont nécessaires, comme **Trumpet Winsock**. Une rumeur non vérifiée prétend que Microsoft distribue une version redistribuable d'un serveur X dans les milieux académiques.

17.6 MS-WINDOWS

Produit	coût	compagnie	version
HCL-eXceed	??	Hummingbird Software	X11R5
eXcursion	??	DEC	X11R5
MultiView/X	??	JSB Corp.	X11R4
PC-Xview	\$445	NCD Inc.	X11R5
Xoftware	??	AGE Logic	X11R4
eXodus	\$295	White Pine Software	X11R5
ReflectionX	\$469	WRQ, Inc.	X11R5

Ces logiciels nécessitent en général une pile TCP/IP, comme **Trumpet Winsock** ou encore MS-WINDOWS'95.

17.7 MS-WINDOWS NT

Il semble que X11R6 a été compilé pour MS-WINDOWS NT.

17.8 OS/2

Le portage de **XFree86** est en cours.

Produit	coût	compagnie	version
X Window	\$150	IBM	X11R5

17.9 Atari

Une version commerciale d'un serveur X11R5 pour Atari sous TOS est disponible chez :

X/software
 Michael Gehret
 Marktstrasse 8
 D-W-8944 Groenenbach
 Germany
 voice : +49 8334 1411
 fax : +49 8334 6245
 email : info@xsoft.uucp

18 Problèmes courants

18.1 Accès au serveur refusé

Cette erreur provient en général d'une variable DISPLAY mal initialisée ou d'un problème d'authentification (*voir section 14 page 57*). On peut également vérifier si le serveur fonctionne en se connectant via telnet au serveur sur le port 6000. Certains systèmes affichent sur la console du serveur des messages informatifs, ou les stockent dans le fichier X11.log.

18.2 Erreur de bibliothèques partagées

Certains systèmes d'exploitation (la plupart) offrent des exécutables de logiciels sous **X11** qui utilisent des bibliothèques partagées (anglais: *shared libraries*). Il se peut que le chemin de recherche de ces bibliothèques soit incomplet. On peut alors l'ajouter manuellement avec, par exemple :

```
export LD_LIBRARY_PATH="{LD_LIBRARY_PATH}:/usr/X11/lib"
```

Troisième partie

Réseaux informatiques

19 Introduction

Le système UNIX, par son ouverture, a toujours favorisé l'échange de données, que ce soit du courrier électronique, des forums de discussion, ou des fichiers. Après tout, les premières versions du système d'exploitation UNIX, en particulier la version **BSD**, furent distribuées, commentées et améliorées sur un réseau de messagerie (le fameux réseau **Usenet**). Encore aujourd'hui, de nombreux projets, comme **Linux** ou **NetBSD** n'auraient pu voir le jour sans des réseaux mondiaux d'échanges. Il est donc normal que UNIX soit toujours lié aux grands réseaux de communication planétaires, comme **Internet**, ou le réseau plus amateur utilisant le protocole **UUCP**. La figure 19 résume la connectivité mondiale des réseaux informatiques **Internet**, **BITNET** et **store-and-forward**.

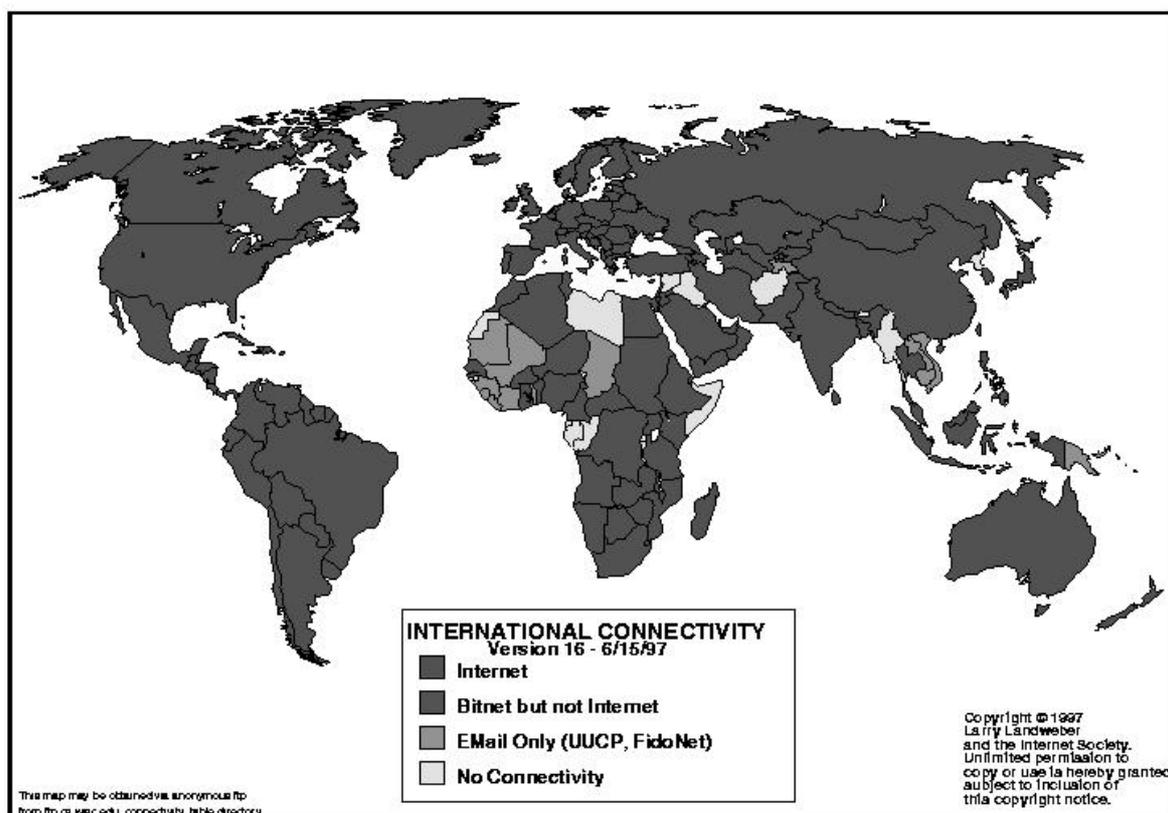


FIG. 10 – Connectivité internationale

20 Principaux réseaux UNIX

20.1 Internet

20.1.1 Introduction

Ce réseau est un réseau temps réel (ou connecté), où chaque ordinateur est relié aux autres par des liaisons fixes (par exemple des lignes louées⁵⁴) et via des passerelles entre réseaux locaux⁵⁵. Au départ réseau académique reliant les universités des États-Unis, il s'est étendu à la fois en espace, pour atteindre maintenant le monde entier, et en types d'organisations (gouvernements, sociétés, firmes, etc.). Internet offre un nombre très grand de services, et de nouveaux sont ajoutés de façon permanente. En raison du fait que les liaisons sont permanentes, les coûts associés sont extrêmement élevés (du moins en comparaison avec les autres méthodes d'accès).

⁵⁴On entend par ce mot des lignes dédiées et permanentes. On parle aussi de circuit loué par opposition au réseau commuté téléphonique normal (anglais: *switched*). Ces lignes sont de capacités diverses : par exemple 64kbps (États-Unis : 56kbps) ou encore T1 (USA : 1.5 Mbps) et T3 (USA : 45 Mbps).

⁵⁵Souvent sur réseau **Ethernet**, à ne pas confondre.

Il existe cependant des protocoles permettant des connexions temporaires (via ligne téléphonique commutée à l'aide d'un **mo-dem**, ou **RNIS**), nommés **SLIP** et **PPP**. Dans ce cas, les services sont les mêmes que lors d'une connexion permanente, mais à sens unique dans beaucoup de cas, le système n'étant pas connecté de façon permanente.

Il est évident que des méthodes **store-and-forward** d'accès à **Internet** existent : elles ne permettent en général cependant que l'accès au **mail**⁵⁶ et aux **news**. Parmi elles, citons **UUCP** (voir section 20.2 page 69).

20.1.2 Description technique

TCP/IP est un protocole **packet-switched realtime**, ce qui signifie qu'il permet l'émission de paquets de données (par exemple formés de 1000 caractères) depuis un poste ou nœud (anglais: *node*) du réseau jusqu'à n'importe quel autre poste du réseau en multiplexant le trafic sur des lignes partagées, et ceci sans stockage intermédiaire de nœud en nœud de l'information. Internet est composé de plusieurs sous-réseaux (des milliers) qui échangent entre eux des paquets TCP/IP (un exemple de sous-réseau est à la figure 11). Des **routers** permettent d'aiguiller les différents paquets entre sous-réseaux en fonction de leur adresse de destination.

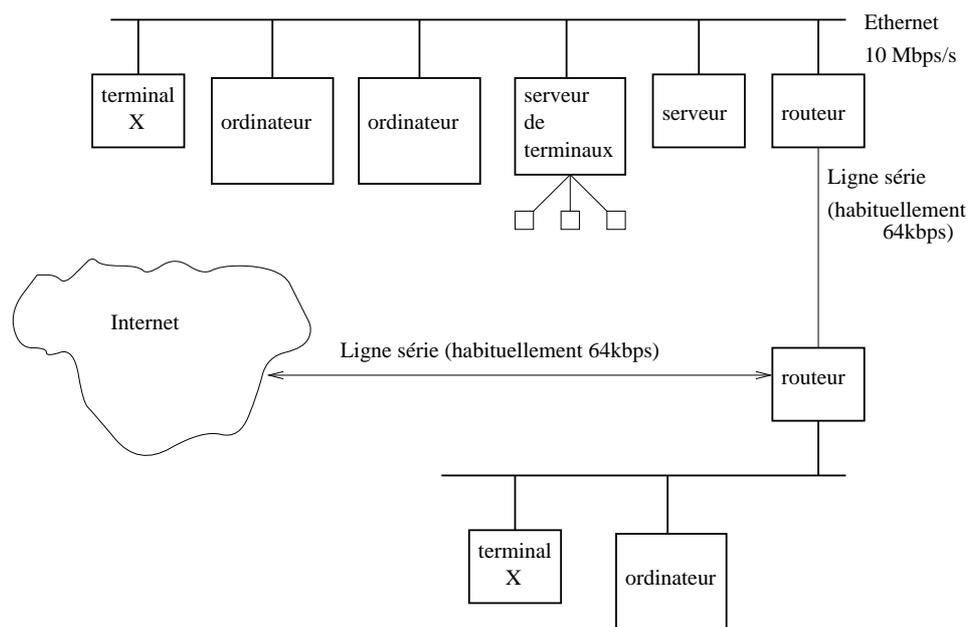


FIG. 11 – Exemple de configuration d'un réseau IP

Adresses Chacune des machines connectées sur Internet possède une⁵⁷ adresse numérique unique (de la forme a.b.c.d, où les lettres représentent des nombres décimaux (ou d'autres bases en syntaxe **C**) pouvant valoir de 0 à 255). On a donc de l'ordre de 4 milliards d'adresses différentes possibles sur 32 bits. Chacune de ces adresses numériques correspond à une ou plusieurs dénominations textuelles, la conversion étant automatiquement effectuée par un serveur de noms (voir section 31 page 100). Chacune des adresses textuelles⁵⁸ correspond également à une ou plusieurs adresses numériques (ou **adresse IP**) et vice versa.

Classes Il existe cependant plusieurs types d'adresses (aussi appelées classes) qui permettent une répartition hiérarchique de l'adressage afin de simplifier l'administration et d'augmenter la vitesse du routage des paquets IP entre les différents réseaux. On en distinguera trois: historiquement c'étaient les seules disponibles (voir la figure 12 on the next page); depuis assez longtemps on utilise le concept de sous-réseau (anglais: *subnet*) qui permet d'étendre cette définition à une répartition quelconque:

Classe A Le réseau est identifié par 8 bits d'adresse et la machine par les 24 bits restants. Ce réseau peut donc contenir de l'ordre de 16 millions de machines,

Classe B Le réseau est identifié par 16 bits, la machine par 16 bits. On a donc de l'ordre de 65000 machines,

Classe C Le réseau est identifié par 24 bits, la machine par 8 bits (256 machines).

⁵⁶Des passerelles mail pour de nombreux services normalement interactifs existent néanmoins.

⁵⁷Parfois plusieurs. En fait, au moins deux, si l'on compte l'adresse locale, ou **loopback**, 127.0.0.1 ou localhost. Certaines machines spéciales sont appelées routeurs et possèdent des dizaines d'interfaces et donc souvent autant d'adresses. Certains serveurs n'ont qu'une interface mais plusieurs adresses. Parfois même, une adresse IP est assignée à plusieurs machine via un écran de répartition de charge (anglais: *load-balancer*).

⁵⁸de la forme machine.sous-domaine.domaine.topdomain, avec un nombre de sous-domaines quelconque, fréquemment aucun.

	193	72	186	3	
Classe A	réseau	machine			0-127.1-254.1-254.1-254
Classe B	réseau		machine		128-191.0-254.1-254.1-254
Classe C	réseau			machine	192-223.0-254.0-254.1-254

Figure 12: Classes d'adresses IP originelles

Détails techniques: masque de sous-réseau En fait, ce que l'on vient de décrire n'est que la pointe de l'iceberg: si l'on considère une adresse IP comme un mot de 32 bits, on peut définir un **masque** de bit servant au routage. Le masque définit le sous-réseau: si le masque est aligné sur les octets, on retrouve nos classes ci-dessus: p.ex. 255.255.255.0 est équivalent à 24 bits à 1, donc 192.168.42.0/24 est le sous réseau de classe C de 192.168.42.0 à 192.168.42.255. Réfléchir ainsi permet de créer un sous-réseau (anglais: *subnet*) comportant moins d'adresses qu'une classe C (p.ex. 4 adresses peuvent suffire pour une liaison PPP y compris broadcast et adresse réseau propre). On peut aussi faire du sur-classement (p.ex. /10). On remarque également que les classes historiques se distinguent également par des bits à 1 ou 0 à leur début. Notons qu'il existe d'autres classes, comme des plages d'adresses privées ou multicast que l'on ne traitera pas ici.

A court d'adresses ? Internet manquant⁵⁹ de Classes A et B, des politiques restrictives ont été mises en place quant à l'attribution de segments d'adressage. Actuellement, il est presque impossible d'obtenir autre chose que des Classes C ainsi que des sous-réseaux de Classe C. Cette situation vient du fait que des abus manifestes ont eu lieu dans le passé. Historiquement, certains paquets d'adresses (p.ex. 35.b.c.d) étaient toutes des Classes A (pour simplifier le routage). Aujourd'hui, on alloue souvent des Classes C à l'intérieur de Classe B ou A, les classes B étant souvent allouées par pays ou par fournisseur de service. Notez que certaines adresses sont réservées (par exemple a.b.c.0 et a.b.c.255).

Protocoles TCP/IP offre deux services de transport de bas niveau : **TCP**, protocole de session (bidirectionnel, connecté et fiable) et **UDP** (protocole rapide mais non fiable⁶⁰), et par-dessus ces services de transport, d'autres services de plus haut niveau ont été définis (particulièrement dans les **RFCs**.). TCP/IP est un protocole ouvert et évolutif, de sorte que sans cesse de nouveaux protocoles de niveau supérieur sont définis. Les protocoles de haut niveau actuellement utilisés sont par exemple **TELNET**, **SMTP**, ou **HTTP**. D'ailleurs une révision importante du protocole est en cours de définition (**IPng**; anglais: *Internet Protocol, next generation*) afin d'augmenter la flexibilité, l'efficacité et la sécurité. Ce protocole permettra également de reléguer aux oubliettes tout problème de nombre d'adresses : au lieu des 32 bits d'adresses usuels, **IPng** utilisera 128 bits. Au bas niveau, TCP/IP peut être transmis par des média variés, comme **Ethernet**, **Token Ring** ou des liaisons sérielles.

20.1.3 Sécurité

Introduction Internet étant un réseau public, de surcroît coopératif, le risque n'est pas nul que l'information circulant soit interceptée, ou falsifiée. Il est, par exemple, parfois déconseillé d'accéder à votre compte à travers Internet, car quelqu'un pourrait intercepter votre mot de passe⁶¹. Des logiciels assurant une plus grande sécurité existent, mais ils ne sont pas encore très répandus. Parmi eux, citons *S/Key*, pour la connexion interactive par échange de mots codés, *ssh* pour le chiffrement **IDEA** ou **3DES** et l'authentification **RSA**, ou encore **Kerberos**, **SecureRPC**, pour les échanges sur des réseaux locaux.

De nombreuses méthodes documentées de piratage existent; parmi elles, la plupart sont basées d'une manière ou d'une autre sur la supposition (incorrecte) qu'une adresse IP ou un nom de machine peuvent être utilisés comme authentification.

Il existe cependant des solutions à ce problème, comme par exemple le **secure telnet/telnetd** ou les nouvelles versions projetées du protocole **WWW**. Il convient de se souvenir cependant que sans ces méthodes, l'utilisation d'Internet repose principalement sur la confiance mutuelle entre les sous-réseaux formant Internet. Cela est en train de changer (surtout avec la nouvelle révision du protocole **TCP/IP**, le fameux **IPng**).

Une entreprise peut également se protéger du réseau Internet en utilisant des **firewalls** combinés à un relais (anglais: *proxy*).

Terminologie La terminologie de la sécurité est relativement étoffée. On ne donnera ici que les termes les plus courants. Pour plus d'information, on se référera à un document traitant de la sécurité des réseaux informatiques (comme [8] ou encore les **FAQs** du forum de **news comp.security.misc**).

authentification On entend par ce mot le fait de vérifier et de s'assurer (ou d'assurer à un tiers) l'identité d'un utilisateur ou d'une machine à des fins de limitation d'accès ou de facturation. Tous les services Internet utilisent une méthode d'authentification à niveaux de sécurité variables, principalement pour assurer l'identité des utilisateurs à des fins de contrôle d'accès. Il existe

⁵⁹Idéalement, on a de l'ordre de 65000 classes B ou 256 Classes A. Pour des questions d'optimisation du routage, on a choisi au départ de dénoter le type de classe par la valeur du premier octet de l'adresse IP. Beaucoup de Classes A ont été allouées à des organismes n'utilisant qu'un faible pourcentage de ces adresses, réduisant ainsi de beaucoup les adresses disponibles.

⁶⁰Cela veut dire que la retransmission de paquets altérés par une erreur de transmission ou perdus doit être gérée par le programme utilisateur, au contraire de **TCP** qui se charge de tout.

⁶¹Un cas très sérieux de ce genre de piratage s'est produit en 1993 sur un grand réseau américain. A une plus petite échelle, vous n'êtes pas à l'abri d'un *snooping* de votre réseau **Ethernet** local par un appareil d'analyse de trafic ou un ordinateur personnel. Un bon survol des problèmes généraux de sécurité se trouve dans [8].

différents types d'identification qui sont décrits au prochain paragraphe. Certaines des méthodes sont acceptées devant les tribunaux. Souvent, les États interdisent certaines de ces méthodes.

chiffrement On entend par ce mot le fait de coder les données transmises de façon à rendre impossible l'interception de celles-ci par un espion (anglais: *snooper*).

Méthodes d'authentification Il existe différentes manières d'assurer l'identité de l'utilisateur. Certaines permettent également d'assurer la provenance de messages, comme **PGP** (voir section 7.6.1 page 44). D'autres sont très primitives mais très répandues.

par un mot de passe Il s'agit de la méthode la plus simple et la plus répandue. Elle souffre principalement du risque d'écoute (et donc d'interception des mots de passe) des réseaux coopératifs.

Avantages – pas de risque de piratage par remplacement de machine.

Inconvénients – le mot de passe circule en clair sur le réseau pour un espion éventuel.

par un contrôle de l'adresse de l'expéditeur Cette méthode a l'avantage de permettre la connexion distante sans mot de passe, en se basant simplement sur deux hypothèses : premièrement les deux machines ne sont pas publiques (ou du moins d'un niveau de sécurité et de reconnaissance multiutilisateur suffisants) et il est impossible de produire de fausses adresses. Ces deux hypothèses ne sont en général plus satisfaites pour deux raisons : la multiplication des stations personnelles et la mauvaise administration des réseaux. De toute manière, c'est partir sur une mauvaise base d'utiliser les adresses comme authentification.

Avantages – le mot de passe ne circule pas.

Inconvénients – des méthodes de piratage cependant complexes⁶² (anglais: *sequence number guessing*) rendent périlleux ce type d'identification sans une installation sûre (surtout au niveau du **router**). D'autres méthodes, notamment celle du **DNS reverse attack** existent, encore que cette dernière, sous réserve de certaines précautions, est facilement évitable.

par un système à clé révélée Cette méthode est la plus moderne et la moins répandue, à part dans le courrier électronique. Elle permet la signature vérifiable et le chiffrement d'informations, qu'il ne sera pas possible ni d'intercepter ni de répéter par la suite.

Avantages – le mot de passe ne circule que de façon cryptée et il est possible d'authentifier de façon sûre les correspondants.

Inconvénients – les problèmes sont surtout de nature non technique (problèmes de patentes, de compatibilité : il faut réécrire certains protocoles) (voir section 7.6.1 page 44).

Conseils d'accès Les conseils suivant peuvent être utiles :

- pas de mot de passe en clair (par exemple tapé lors d'une connexion telnet entre deux réseaux sur **Internet**),
- pas de `.rhosts` sur une machine externe à votre réseau,
- vérifier les modalités avec l'administrateur avant toute installation d'authentification basée sur les adresses (anglais: *source address check*). On préférera par exemple l'authentification `xauth` plutôt que `xhost`, surtout à travers un grand réseau.

20.1.4 Événements médiatiques

Cette section se veut un résumé des événements reportés (ou ignorés) par les médias sur le réseau **Internet** en apportant des précisions ou des corrections où cela s'avère nécessaire. Une bonne source d'informations est le **forum de news : comp.security.misc**.

The Internet Worm Le 2 novembre 1988, Robert MORRIS Jr., un étudiant de l'université Cornell aux États-Unis a créé un programme qui pouvait se transmettre de lui même sur **Internet** (alors encore appelé **ARPAnet**) grâce à un **bug** dans un logiciel présent sur toutes les machines du réseau : **sendmail**, le logiciel de réception et d'envoi du courrier électronique privé et à cause d'un effet de bord d'un logiciel nommé **fingerd**, permettant de répondre aux requêtes **finger**. Malheureusement pour lui, l'étudiant n'avait pas programmé correctement un test de condition booléenne devant empêcher de réinfecter des machines déjà infectées : le **worm** se mit à se transmettre à une telle vitesse que bientôt la plupart des machines furent contaminés et rendues inutilisables en raison d'une multiplication infinie de processus. Il écrivit un programme permettant de se débarrasser du **worm** mais, les communications étant devenues quasiment impossible sur le réseau totalement saturé, personne ne put l'installer dans les temps. La panne générale dura plusieurs jours et vint le moment des comptes : Robert MORRIS Jr. fut condamné à trois ans de probation, 400 heures de service communautaire, à payer les frais du procès et plus de dix mille dollars [10][24].

The Cukoo's Egg Clifford STOLL détaille dans [9] les circonstances qui l'ont amené à traquer (tracer, retrouver, anglais: *trace*) un pirate allemand à travers divers réseaux. Tout cela pour une différence de 75 cents dans un décompte de frais de communication. Son ouvrage et les nombreux films qui ont été réalisés sur le sujet montrent bien que la seule façon d'éviter ou du moins de limiter les dégâts est bien sûr la communication efficace entre les organisations et les particuliers, administrateurs systèmes ou simples utilisateurs.

⁶²Une annonce du **CERT**, un organisme de sécurité, en a parlé récemment (voir section 20.1.4). Cette méthode est cependant connue depuis 1985 quand elle a été publiée dans un journal de **Bell labs**.

Internet piraté (1) Cette annonce a été faite au milieu de l'année 1992. Elle faisait suite à une alerte du **CERT** qui avait reçu beaucoup d'indications de piratages systématiques aux États-Unis. Certains journaux allaient jusqu'à dire que tous les utilisateurs (à ce moment-là estimés à 20 millions) devaient changer leur mot de passe. Ce qui s'était passé était relativement simple. Le compte d'administrateur d'un fournisseur de services de connectivité réseau aux États-Unis avait été piraté. En raison d'une mauvaise configuration d'un routeur, ce pirate avait pu installer un logiciel d'espionnage (anglais: *snooping*) du réseau local. Tous les clients de ce fournisseur de service utilisant leur accès local pour atteindre des ordinateurs distants par un protocole d'identification par mot de passe avaient vu leurs autres comptes piratés. Cela concernait au maximum quelques dizaines de comptes. Cependant, cet incident a permis aux promoteurs du projet **IPng** (anglais: *Internet Protocol, next generation*) de faire accepter l'idée que certains protocoles (principalement ceux à identification par mot de passe) devaient être améliorés avec des méthodes modernes de cryptographie par clé révélée, avec tous les problèmes légaux que cela implique (voir section 7.6.1 page 44).

Internet piraté (2) Cette annonce a été faite au début de l'année 1995. Elle faisait suite également à une alerte du **CERT** après quelques cas de piratages utilisant la méthode du **random sequence number guessing**. Il faut savoir que cette méthode de piratage est facilement évitable par diverses méthodes (principalement au niveau des routeurs), au prix d'une certaine discipline des utilisateurs. Il existe également des méthodes pour empêcher tout piratage de ce type. De plus, cette faiblesse de la méthode d'identification par **source address check** (voir section 20.1.3 page ci-contre) a été décrite dès 1985 dans un journal de Bells labs et n'a donc rien d'une nouveauté.

Internet Virus Une rumeur circule à propos d'un virus, nommé *Good Time Virus*, qui infecterait tout ordinateur dont le propriétaire lit ce message. Il s'agit d'une légende qui a été lancée par un utilisateur d'**AOL**, un grand service interactif américain (anglais: *online service*). Cette légende a été (incorrectement) confirmée par un utilisateur ayant reçu un message avec un tel sujet (qui d'ailleurs l'avertissait du "danger") et qui, paniqué, a contrôlé son disque (ce qu'il avait oublié de faire depuis des mois). Ayant découvert un virus, il l'attribua évidemment au message qu'il venait de lire. La plupart des organismes de sécurité (**CERT** entre autres) ont annoncé qu'il s'agissait d'une blague, le phénomène de panique prenant une ampleur certaine (à cause de la diffusion par boule de neige du message d'avertissement, les utilisateurs croyaient vraiment être menacés personnellement).

20.1.5 Comment s'y connecter

Pour bénéficier de tous les services du réseau **Internet**, il faut obtenir un accès auprès d'un fournisseur de service (anglais: *ISP*). Ces fournisseurs sont de plus en plus souvent commerciaux. Il n'est donc plus nécessaire d'en fournir la liste (la publicité est assez abondante dans les journaux). Cependant, il est bon de savoir qu'en Europe, il existe, à côté des fournisseurs commerciaux (en particulier le réseau pan-européen **EUnet**, fondé en 1982 sur des bases non commerciales, révisées depuis), une multitude de moyens d'accéder à peu de frais au moins au courrier électronique (**mail**) et aux **news** pour un accès personnel. Il existe de nombreuses listes sur le sujet (par exemple **PDDIAL** ou **NIXPUB**, dans le forum **comp.bbs.misc**). Des réseaux amateurs comme **FidoNet** ou **FrancoMédia** offrent également du courrier électronique, des messages et des passerelles vers **Internet**. Dans certains pays, des associations offrent des accès complets à des prix raisonnables sous certaines conditions (notamment la non commercialisation).

20.1.6 Évolution — un peu de politique des réseaux

Certains prétendent que **WWW** est la meilleure chose qui soit arrivée à Internet. Mais d'autres assurent que c'est aussi la pire ! Internet est dans une phase de commercialisation : pas celle des *service providers* (fournisseurs d'accès au réseau) car cela est déjà le cas depuis un certain temps, mais plutôt, et c'est là où est le problème, celle des fournisseurs d'informations.

En effet, à l'origine les utilisateurs d'Internet ne payaient pas directement pour leur accès : ils étaient des étudiants, des chercheurs ou des employés. Ils avaient souvent une connectivité excellente (liaisons permanentes), et, de ce fait, pouvaient également offrir des services (offrir de l'information).

Cependant, de plus en plus, les gens accèdent à Internet via des *bornes* d'accès téléphoniques : ils ne sont plus sur le réseau en permanence, et ne disposent souvent plus d'un compte UNIX. Ils ne peuvent donc plus simplement offrir des informations, mais seulement les *pomper*. Et le plus vite possible (sous-entendu : le moins longtemps *online* possible) parce que souvent on paie à la minute.

Le jour où le déséquilibre entre fournisseurs d'information et utilisateurs sera trop grand, Internet basculera dans la non interactivité (la réception passive d'information), comme la plupart des services du type **videotex** (penser seulement à l'asymétrie des vitesses) [28].

On peut espérer que des initiatives publiques ou privées permettront de rééquilibrer cet état de fait : de nombreuses associations comme **Yinternet.org**, **trash.net** ou encore **ALPHANET NF** ont cet objectif.

20.2 UUCP

20.2.1 Introduction

Ce réseau est formé le plus souvent de systèmes reliés par des liaisons non permanentes établies, par exemple, plusieurs fois par jour, pour recevoir et transmettre des fichiers (courrier électronique, forums, fichiers). Par sa structure **store-and-forward**, les

services interactifs d'**Internet** ne sont évidemment pas possibles, à l'exception de ceux également fournis par des **mailservers**. De plus les délais de propagation sont assez élevés.

20.2.2 Description technique

Le système UUCP est composé d'une multitude d'utilitaires dont certains forment la base du système. La plupart d'entre eux sont lancés de façon transparente pour l'utilisateur par des logiciels de plus haut niveau.

uucico Cette commande est le programme actif. Elle peut être appelée de façon périodique pour appeler tous les systèmes pour lesquels des travaux sont en attente. Elle est également appelée en cas de connexion via **login** d'un utilisateur ayant cette commande comme shell par défaut⁶³. Cette commande se charge de l'établissement de la session et du transfert de fichiers en attente dans les deux sens.

uux Cette commande permet de mettre en attente un fichier, via la commande **uucp** et un fichier de contrôle dans la zone d'attente (**spool**), pour transmission ultérieure à l'aide de **uucico** puis exécution sur un système distant via **uuxqt**.

uucp Cette commande permet de transmettre un fichier à un ordinateur distant.

uuxqt Cette commande permet de lancer les différents programmes de gestion (**rmail**, **rsmtpt**, **rnews**, ...) en fonction des indications dans les fichiers de contrôle.

uustat Cette commande permet de connaître certains paramètres du **spool** et de détruire des travaux.

On remarque donc que **UUCP** n'offre que la couche de transport, et permet l'exécution de commandes à distance après transfert. Ces commandes prennent comme entrée standard le fichier de données associé et agissent sur celui-ci.

20.2.3 Sécurité

Les versions modernes de UUCP (en commençant par HDB (**BNU**) UUCP et **Taylor UUCP**) offrent des moyens de limiter effectivement l'accès au système par des machines non autorisées, tout en maintenant un **login** anonyme (**nuucp**). Les versions plus anciennes n'offrent pas un niveau de sécurité suffisant.

20.3 FidoNet

Par abus de langage, **FidoNet** est l'ensemble des réseaux amateurs sous protocole **FTN**. Ce n'est pas à proprement parler un réseau UNIX puisque ce n'est que depuis quelques années que des versions UNIX des logiciels nécessaires existent. Cependant, c'est sous UNIX que l'on trouve les meilleures passerelles entre les deux mondes **FidoNet** et **Internet**.

Ce réseau, de type **store-and-forward**, est caractérisé par son adressage à 4 (éventuellement cinq dimensions). En effet, un utilisateur **FidoNet** est identifié par son nom et son prénom (d'ailleurs dans un **EchoMail**, c'est-à-dire l'équivalent d'un forum de news, seul ces deux valeurs sont utilisées comme identifiant), par un numéro de point (qui peut ne pas exister), par un numéro de noeud (anglais: *node*), par un numéro de réseau (anglais: *net*) et par un numéro de zone (représentant souvent un réseau séparé ou un continent, par exemple 2 pour l'Europe et 101 pour le réseau **FrancoMédia**). On peut encore y ajouter un type de réseau accolé, mais qui n'est souvent pas utilisé.

La philosophie de ce type de réseau est très différente des réseaux UNIX. On trouvera plus d'informations à ce sujet sur <http://www.fidonet.org> et un guide en français sur <http://www.scdi.org/fidonet/>.

21 Principaux services offerts

La plupart des services sur **Internet** sont en fait interactifs, mais on peut tout de même séparer ce qui *doit* être interactif de ce qui *peut* être interactif. Dans ce qui suit, la machine distante peut toujours être la machine locale, bien évidemment. Dans ce sens, les services locaux sont étendus de façon presque transparente aux services globaux.

21.1 Services non interactifs (store-and-forward)

Par services non interactifs on entend des services qui n'ont pas besoin d'une connexion réelle au moment de la soumission de la commande. La plupart du temps les textes ou fichiers transmis seront mis en attente de la connexion suivante.

mail Courrier électronique privé. Permet d'échanger des messages avec des correspondants du monde entier (*voir section 22 page suivante*).

news Forums de discussion. Permet d'échanger des points de vue dans des conférences à thèmes spécifiques avec d'autres utilisateurs (*voir section 23 page 79*).

uucp Transmission de fichiers.

mailservers Serveurs d'informations via courrier électronique. Une grande partie des services interactifs sont disponibles par l'intermédiaire de serveurs. Nous exploitons pour le moment six serveurs⁶⁴, à savoir

⁶³Sur notre système, il y a un utilisateur par machine se connectant, les noms commençant par uu, et un utilisateur anonyme **nuucp** avec moins de possibilités.

⁶⁴Ces serveurs ne sont pas accessibles à l'extérieur du réseau **ALPHANET**.

`faxserv@alphanet.ch` permet d'envoyer des fax par courrier électronique.

`ftpmail@alphanet.ch` permet de faire du **FTP** via **mail**. Notez bien que ce serveur est local au réseau ALPHANET. Il existe de nombreux serveurs pour les autres réseaux (comme `ftpmail@decwrl.dec.com`)

`keyserv@alphanet.ch` permet d'échanger des clés **PGP**

`bbsserv@alphanet.ch` serveur de gestion de la liste des **BBS** suisses.

`newsserv@alphanet.ch` système de souscription automatisé pour les **news**.

`areafix@alphanet.ch` système similaire à newsserv, moins puissant mais plus facile à utiliser.

`test@alphanet.ch` permet de recevoir des confirmations de réception

`printmail@alphanet.ch` permet de sortir sur papier des mails

Notez qu'il est également possible de poster via **mail** sur la plupart des forums locaux en donnant le nom de forum (en remplaçant les points par des tirets) dans lequel on désire poster, comme par exemple

`alphanet-misc@alphanet.ch`

21.2 Services interactifs

Par services interactifs on entend les services qui nécessitent une connexion effective entre la machine locale et la machine distante au moment même de la demande. Dans la plupart des cas, même l'établissement de la communication au moment même de la demande n'est pas suffisant, car la plupart du temps des connexions permanentes sont établies entre beaucoup d'ordinateurs distants à la fois, de plus pendant des périodes de temps très grandes.

FTP Transfert de fichiers interactif (*voir section 24 page 85*).

Gopher Banques d'informations textuelles distribuées.

IRC Discussion de type CB distribuée (*voir section 29 page 97*).

MUD Jeux multi-utilisateurs distribués.

NFS Partage de fichiers.

POP Lecture de **mail** sur serveur distant.

NNTP Lecture de **news** sur serveur distant (*voir section 23 page 79*).

TALK Discussion interactive (*voir section 26 page 89*).

TELNET Appel de système distant (*voir section 26 page 89*).

WWW Services multimédia interactifs distribués (*voir section 28 page 91*).

X11 Interface graphique sur serveur à distance (*voir section II page 53*).

22 Mail (courrier électronique privé)

22.1 Introduction

Le système **mail** sert à l'échange de courrier électronique privé entre correspondants connectés aux réseaux **Internet**, **FidoNet**, **CompuServe**, **X.400**, **UUCP** et divers autres réseaux. Il existe de nombreux logiciels de courrier électronique dont `mail` et `elm`. Il est également possible d'utiliser ce service pour créer des listes de distributions permettant d'atteindre des personnes aux intérêts ciblés (**mailing list**). Cependant on préférera l'utilisation des forums électroniques pour des raisons de coût et de simplicité d'accès⁶⁵.

22.2 Savoir-vivre

Échanger des messages entre des utilisateurs d'une même machine, ou d'une machine distincte est très simple. Cependant il est aussi très simple de ne pas utiliser ces fonctionnalités de façon correcte. En règle générale, il faut

- Ne pas oublier que votre correspondant est humain.
- Ne pas critiquer les administrateurs de systèmes étrangers pour le comportement de leurs utilisateurs.
- Faire attention à ce que l'on dit à propos d'autres utilisateurs, et à propos de matériel (sens général) soumis à licence.
- Être bref et concis.
- Faire bonne impression, s'appliquer à présenter correctement un message, il n'en sera que plus apprécié et compris.
- Utiliser le champ Subject: (sujet du message) de façon appropriée. Il est de bon ton que celui-ci ait un rapport étroit avec le contenu du message, particulièrement s'il s'agit d'une réponse.
- Adapter votre message au(x) destinataire(s).

⁶⁵Sur ALPHANET on transforme les mailing lists en forums et on diminue ainsi la charge mail totale en centralisant les requêtes à `newsadm@alphanet.ch`.

- Marquer clairement les passages sarcastiques⁶⁶.
- Éviter les signatures gigantesques, il est usuellement considéré qu’une signature dépassant 4 à 5 lignes ne fera qu’agacer vos lecteurs potentiels.
- Limiter la longueur des lignes à une valeur raisonnable. La plupart de vos lecteurs utilisent, malgré eux cela va sans dire, des terminaux limités à 80 colonnes et souffriraient de ne vous comprendre que partiellement. Une valeur de 70 caractères est conseillée afin de permettre une citation aisée et respectant elle-même encore ce critère.
- Éviter les caractères de contrôle. Les accents français par exemple ne sont normalement pas tolérés, à part si votre correspondant vous y invite. Le seul jeu de caractères toléré pour les accents est **ISO Latin 1** (aussi appelé **ISO 8859-1**).
- Éviter de considérer un message privé comme contenant des informations publiables, le correspondant a le droit à sa sphère privée.
- Crypter les informations sensibles ou ne pas les envoyer par le courrier électronique privé (voir section 20.1.3 page 67).
- Ne croyez surtout pas tout ce qu’on vous dit ! Et surtout n’oubliez pas que sans méthode d’authentification sûre (par exemple **PGP**), il est impossible de s’assurer de l’identité de son correspondant.

Il faut également savoir que l’envoi de messages non sollicités (anglais: *junk mail*) à teneur publicitaire ou non est normalement considéré comme un manquement grave aux règles élémentaires de comportement sur les réseaux.

La communication électronique est impersonnelle et froide. Pour cette raison, le courrier électronique est rempli de symboles curieux à interpréter afin de comprendre l’humeur de votre interlocuteur et d’éviter des malentendus. La table 7 décrit quelques-uns de ces symboles. Il existe des **FAQ** à ce sujet ainsi que le logiciel *smiley* qui peut être utilisé pour expliquer un sourire. Si vous ne comprenez pas, tournez votre tête de 90 degrés vers la gauche. Vous verrez un petit bonhomme souriant.

: -)	Sourire standard	# -)	C’était une nuit d’enfer !
% -)	36 heures de terminal ...	% -{	Malade.
% -	J’ai travaillé toute la nuit.	(: -&	En colère.
: %) %	J’ai de l’acné.	: - (Triste.
: -0	Ne pas crier !	: -	Je jure !
: -]	Sarcasme mordant.	; -)	Sourire sympathique.
> : - (Irrité.	> : -)	Malin.
E - : -)	Radio-amateur.	[: -)	Walkman.
ï - =< ***	Lance-flamme.	: q	Utilisateur de vi voulant sortir de emacs.

TAB. 7 – Quelques expressions courantes

22.3 Interface utilisateur

De nombreuses interfaces existent. Elles diffèrent surtout par la facilité d’emploi et la présentation (terminal **VT100** ou interface **X11** (voir section 16 page 59)). On ne parlera dans cette section que des clients disponibles sous environnement UNIX. De nombreux logiciels existent également sur micro-ordinateurs permettant l’accès au mail via un serveur **POP** distant, ou via un **spool** local pour la lecture hors-ligne (anglais: *offline mailreader*). On citera aussi le logiciel **PINE** qui offre un environnement standardisé pour le **mail** mais aussi les **news**, ainsi qu’un éditeur simple à utiliser, *pico*. On citera enfin **GNUS**, lecteur de **mail** et **news** dans l’éditeur **Emacs**.

22.3.1 mail : interface ligne

Pour l’appeler, il suffit de taper `mail` (ou si l’alias n’a pas été installé chez vous, ce qui est peu probable, `mailx`). Ce lecteur de courrier privé est orienté ligne. Il est extrêmement simplifié, mais a l’avantage de ne pas nécessiter de terminal à codes **VT100** ou supérieur. Notez qu’il est possible de préciser plus d’une adresse de destinataires sur la ligne de commande, en séparant les adresses successives d’un espace. Lors de l’entrée du message, des commandes peuvent être effectuées en commençant la ligne par `~`, comme par exemple `~?` pour une courte aide ou `~!commande`, où `commande` est une commande du **shell**; ou encore `~q` pour annuler le message.

La figure 13 page ci-contre montre un exemple de **mail** envoyé par le professeur Falken à Bill CLINTON⁶⁷. On peut terminer un message par `CTRL-D` ou parfois par un point en début de ligne. Suivant la configuration indiquée dans le fichier `.mailrc`, la commande demande ensuite un destinataire `Cc:` (copie carbone). Pour ne pas envoyer de copie à un tiers, répondre simplement par `RETURN`. Comme `CTRL-D` est souvent interprété comme un terminateur standard, il est indiqué de savoir l’utiliser.

⁶⁶Le symbole `: -)` peut être utilisé à cet effet. Si vous ne comprenez pas, tournez la tête de 90 degrés vers la gauche : vous verrez un petit visage souriant. Il ne faut pas minimiser l’utilisation de symboles de ce genre. En effet, les messages textuels rendent difficilement l’atmosphère et le contexte d’un message. La table 7 décrit quelques-uns de ces symboles.

⁶⁷Non, je ne citerai aucun autre président étatsunien.

```
<mips> falken:/d2/users/falken> mail president@whitehouse.gov
Subject: Our appointment at 06:00 PM

Dear Billy,
I'm afraid I can't come for 06:00, however I will see you at the lunch.
CTRL-D
<mips> falken:/d2/users/falken>
```

Figure 13: Exemple d'envoi de message privé avec mail

Pour lire ses messages, entrer simplement `mail` et une courte liste des messages apparaîtra. On entre ensuite le numéro du message que l'on veut lire. On utilise enfin `R` pour répondre à un message.

22.3.2 elm, mutt : interface VT100

Pour l'appeler il suffit de taper `elm`. Ce lecteur de courrier privé est orienté terminal semi-graphique. Il est spécialement conseillé pour les novices et en cas de volume abondant. Il offre des possibilités étendues comme le support multimédia (**MIME**) limité et le désarchivage automatique d'archives. Il permet, comme **mail**, de spécifier un ou plusieurs destinataires sur la ligne de commande, en paramètre, ou pendant l'exécution du programme avec `m`.

L'utilisation est relativement simple : se déplacer sur le mail désiré avec les touches du curseur (ou simplement le numéro indiqué dans la première colonne puis `RETURN`), sélectionner avec `RETURN`. Ensuite, on entre en mode d'affichage de message. Dans ce mode, on passe à la page suivante avec `ESPACE`, et on revient au mode d'affichage de la liste des messages avec `i`. Notez que dans les deux modes, il est possible de répondre à un message avec `r` (`g` pour répondre à tous les destinataires), de le sauver avec `s` (sauvetage par défaut dans le répertoire `Mail/`) de le détruire par la touche `d`. Notez que si l'on quitte le logiciel avec `q`, les modifications de l'état de votre boîte aux lettres seront sauvées (destructions, lecture, etc.). Par contre, quitter à l'aide de `x` permet d'annuler toutes les modifications effectuées.

`elm` affiche en permanence l'état de chacun des messages (détruit, lu ou multimédia). L'éditeur par défaut pour la rédaction de messages peut être modifié (voir section 8.1.1 page 46).

`mutt`⁶⁸ est un lecteur de courrier très similaire à `elm` puisqu'il en reprend pratiquement toutes les fonctionnalités. Il en ajoute aussi d'autres qui le rendent plus intéressant, comme la possibilité de récupérer son courrier sur un serveur POP, ou de trier le courrier dans différents dossiers. De plus, il gère mieux qu'`elm` **MIME** et d'autres aspects (comme les signatures électroniques).

22.3.3 Logiciels clients pour ordinateurs personnels

En plus des logiciels disponibles sous environnements UNIX et de la possibilité de les lancer à travers le système **X11** (comme par exemple **xmh** ou **xbiff** pour la notification (voir section 16 page 59)), il existe de nombreux clients sur ordinateurs personnels permettant la lecture et l'envoi de messages privés (**mail**). La plupart présupposent une connexion interactive (permanente ou non). Certains peuvent être utilisés avec une couche permettant de compresser ou d'automatiser les transferts (comme **UUCP**). La table 8 liste ces clients pour différentes plates-formes.

Plate-forme	nom logiciel	accessibilité	type
Amiga	AmiPOP AmigaElm	libre shareware	transfert POP lecteur
Macintosh	Eudora	commercial ≥ 2.0	client POP
MS-DOS	PINE	libre	nécessite serveur spécial
MS-WINDOWS	Eudora PC/TCP	commercial commercial	POP, avec Winsock clients POP

TAB. 8 – Clients **mail** pour ordinateurs personnels

22.4 Tri du courrier (procmail)

Lorsqu'on reçoit beaucoup de courrier, ce qui arrive rapidement dès qu'on s'inscrit à des listes de diffusion (*mailing-list*), il devient intéressant de pouvoir trier son courrier dans différentes boîtes. Ce qui est encore plus intéressant est d'avoir un logiciel qui s'en occupe automatiquement à la réception du courrier, `procmail`⁶⁹ le permet.

Si ce logiciel est déjà installé sur votre machine, il suffit d'indiquer dans votre fichier `.forward` d'utiliser `procmail` et dans le fichier `.procmailrc` comment doit être trié le courrier. Pour cela vous indiquez dans quelle boîte mettre la lettre arrivant suivant qu'elle provienne de Pierre Durant, ou du site `jussieu.fr`, ou bien si le texte `guide unix` figure dans le titre, etc... Il est

⁶⁸cf <http://www.cs.hmc.edu/~me/mutt/index.html>

⁶⁹cf <http://www.ii.com/internet/robots/procmail/>.

également possible de demander à procmail de lancer un programme ou d'envoyer une réponse à la réception de tel type de courrier.

Voici un exemple de fichier .forward :

```
"|IFS=' ' && p=/usr/local/bin/procmail && test -f $p && exec $p -Yf- || exit 75 #ricou"
```

et un fichier .procmailrc :

```
# Si certains chemins de PATH n'existent pas, enlevez les.

PATH=$HOME/bin:/usr/bin:/bin:/usr/local/bin:.
MAILDIR=$HOME/Mail           # vérifiez que ce répertoire existe
DEFAULT=$MAILDIR/mbx
LOGFILE=$MAILDIR/from
LOCKFILE=$HOME/.lockmail
SENDMAIL=/usr/sbin/sendmail

:0
* ^From.*henry                # tout ce qui vient d'Henry
henries                       # va dans $MAILDIR/henries

:0
* ^(To:|Cc:).*@cru.fr         # ce qu'envoie le CRU dans sa liste de diffusion
lmb                           # va dans $MAILDIR/lmb

:0                            # ces lettres disparaissent et j'ajoute
* ^From.ricou*                # un "idiot" dans le fichier /tmp/idiot
| echo "idiot ">> /tmp/idiot    # (on peut lancer un programme plus intéressant)
# Attention: peut-être faut-il utiliser plutôt
# | (echo "idiot" >> /tmp/idiot; cat > /dev/null)
# si les mails dépassent la taille d'un pipe usuel
# (je vous laisse réfléchir pourquoi)

:0
* ^Subject:.*questions        # toutes les 'questions' vont dans
{
:0 c
question                      # $MAILDIR/question et

:0                            # on renvoie une gentille reponse
| (formail -r ; \
    echo "J'aime pas les questions...";\
    echo "-- "; cat $HOME/.signature \
) | $SENDMAIL -oi -t
}

# Tout ce qui n'a pas été pris en compte sera rangé dans $MAILDIR/mbx

Une autre façon d'utiliser procmail est de lui demander de trier une boîte aux lettres pleine. Voici le script qui permet de trier sa boîte (dans le cas où l'on a pas mis le fichier de forward pour lancer le tri à la réception) 70 :
```

```
#!/bin/sh

ORGMAIL=/var/spool/mail/$LOGNAME

if cd $HOME &&
test -s $ORGMAIL &&
lockfile -r0 -l3600 .newmail.lock 2>/dev/null
then
trap "rm -f .newmail.lock" 1 2 3 15
umask 077
lockfile -l3600 -ml
cat $ORGMAIL >>.newmail &&
```

⁷⁰les critères de tri sont toujours ceux indiqués dans le fichier .procmailrc.

```

cat /dev/null >$ORGMAIL
lockfile -mu
formail -s procmail <.newmail &&
rm -f .newmail
rm -f .newmail.lock
fi
exit 0

```

22.5 Personnes de contact

La table 9 liste les principales adresses de l'administration système. Il est évident que les conventions à ce sujet dépendent de votre administration locale. En général cependant, les adresses *postmaster* et *usenet* sont valides et désignent respectivement l'administrateur de la machine (et du mail) et le responsable du système de **news**. Certains systèmes définissent également *abuse* pour annoncer les abus.

Adresse	description
sysadm	administration
acctadm	comptes
uucpadm	système UUCP
fileadm	archives de fichiers
fnetadm	système FidoNet
sladm	SLIP
faxadm	serveur de fax
usenet	
newsadm	système de news
postmaster	
mailadm	système de mail
admin	
root	Administration

TAB. 9 – Adresses **mail** usuelles des administrateurs

22.6 Aspects techniques

22.6.1 Transmission

Sur Internet, le courrier électronique privé est transmis de site à site via des connexions **TCP** et ceci presque instantanément. En général, le nombre de sites intermédiaires est très faible. Dans certains cas particuliers comme des passerelles administratives gérant un domaine particulier, il y a plus d'un site intermédiaire. Si le destinataire n'est pas sur Internet, le courrier est transmis à un site qui s'est annoncé comme pouvant router les mails sur la destination (**MX forwarder**). Dans ce cas, comme sur un réseau **UUCP** ou **FidoNet**, le nombre de relais peut devenir important. De plus, comme ces liens sont **store-and-forward**, il se peut que le message mette plusieurs jours avant d'arriver à destination.

22.6.2 Sécurité

La plupart des systèmes de **mail** n'offrent pas de moyen sûr d'identification de l'expéditeur d'un message. Après tout, c'est également le cas pour la bonne vieille poste. Cependant, il est parfois possible de retrouver la provenance d'un message grâce aux champs Received:. Il faut cependant savoir que ces champs sont établis à l'aide d'une authentification de type **source address check** (voir section 20.1.3 page 68). De plus, les messages transmis peuvent être interceptés par un espion (anglais: *snooper*) ou modifiés par un pirate (anglais: *cracker*). Une certaine sécurité peut être obtenue à l'aide de logiciels comme **PGP** (voir section 7.6.1 page 44) (pour l'authentification aussi bien que pour le chiffrement du contenu).

22.6.3 Spécification d'adresses

Toute adresse soumise au système de message doit être de la forme générale suivante :

user@computer.domain1.domain2.domain3.topdomain

Le nombre de sous-domaines peut être quelconque. Le domaine **.UUCP** ne doit jamais être utilisé, il vaut mieux n'utiliser aucun domaine. De même, toute adresse doit se conformer le plus possible à ce schéma. Cela vaut également pour les adresses en retour telles que spécifiées dans les champs From: et Reply-To:.

Il faut particulièrement éviter les symboles suivants dans une adresse mail :

% Sa signification est similaire à @, tout en étant non prioritaire : ce symbole peut donc servir à spécifier des passerelles. Par exemple `schaefer%sat.org@alphanet.ch` demande au système d'acheminer le message pour `schaefer%sat.org` via le serveur de mail de `alphanet.ch`. Ce symbole ne devrait être utilisé qu'en de rares cas.

! Sa signification est par contre ambiguë: par exemple, dans le monde **UUCP**, `vulcan!schaefer@alphanet.ch` signifie délivrer à `schaefer@alphanet.ch` via la machine `vulcan`. Au contraire, dans le monde **Internet**, cette même adresse signifie délivrer tout d'abord à `alphanet.ch` puis expédier à `schaefer@vulcan`. On évitera donc ce genre d'adresses dans tous les cas (% est bien plus clair). Notez que ! est parfois employé dans les mécanismes de transfert internes. Par contre l'utilisateur ne devrait pas s'en servir.

En cas de doute, *postmaster* (ou *postmaster@une_adresse_valide*) renseigne les utilisateurs. Il faut évidemment remplacer dans l'adresse précédemment citée l'adresse du site distant. On trouvera dans [13] une description des différents formats d'adresses. De plus, dans le forum **alt.internet.services** est postée une fois par mois la liste des passerelles disponibles (ce texte est archivé dans `/users/admin/info/faqs`).

22.6.4 Enveloppe et entête

On différencie l'**enveloppe** de l'**entête** (anglais: *headers*) d'un message. Les informations échangées par les serveurs de transfert (**MTA**) forment en règle générale l'enveloppe. Les informations acheminées avec le message consistent en un entête et un corps de message (anglais: *mail body*).

Le corps du message contient le message tel qu'il a été tapé. Les entêtes contiennent des informations administratives et formatées (sujet du message, destinataire, copie carbone, identification du message, expéditeur). La figure 14 montre l'entête et le corps d'un message

```
Return-Path: <linux-leman-owner@vulcan.alphanet.ch>
Received: from vulcan.UUCP (bsmtp@localhost)
    by defian.alphanet.ch (8.9.3/8.9.3/Debian 8.9.3-21) with bsmtp id JAA00905
    for schaefer@defian.alphanet.ch; Fri, 21 Jun 2002 09:00:51 +0200
X-Authentication-Warning: defian.alphanet.ch: bsmtp set sender to linux-leman-owner@vulcan.alphanet.ch using
Received: (from mdom@localhost)
    by vulcan.alphanet.ch (8.11.6/8.11.6) id g5L0jvd01407
    for linux-leman-outstuff; Fri, 21 Jun 2002 02:45:57 +0200
Received: from unix03.wehi.EDU.AU (unix03.wehi.edu.au [128.250.252.86])
    by vulcan.alphanet.ch (8.11.6/8.11.6) with ESMTP id g5L0jwo01383
    for <linux-leman@alphanet.ch>; Fri, 21 Jun 2002 02:45:33 +0200
Received: from pc249.alpha.wehi.edu.au (pc249.alpha.wehi.edu.au [192.168.65.35])
    by unix03.wehi.EDU.AU (8.12.1/8.12.1) with SMTP id g5L0jSb1006103
    for <linux-leman@alphanet.ch>; Fri, 21 Jun 2002 10:45:29 +1000 (EST)
From: Frederic Schutz <schutz@mathgen.ch>
To: linux-leman@alphanet.ch
Subject: 2 nouvelles en provenance de Debian: SELinux et developpeur a Geneve
Date: Fri, 21 Jun 2002 10:46:14 +1000
Organization: AAAA (Association Australienne Anti-Acronymes)
Message-ID: <63t4hukpv67h8f1mi3gc4vqbakvmudvgf7@4ax.com>
X-Mailer: Forte Agent 1.9/32.560
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 8bit
X-Scanned-By: MIMEDefang 0.6 (http://www.roaringpenguin.com/mimedefang/)
Sender: owner-linux-leman@alphanet.ch
Precedence: bulk
Reply-To: linux-leman@alphanet.ch
Status: RO
X-Status:
```

1) Peut-être que vous avez déjà entendu parler de SELinux (Security Enhanced Linux), ce patch créé par la NSA (<http://www.nsa.gov/selinux>) et qui incorpore de nouveaux mécanismes de sécurité dans le noyau Linux. Si vous voulez voir à quoi ça ressemble, Russell Cocker, un développeur Debian, a mis à disposition une machine avec SE Linux, cf <http://lists.debian.org/debian-devel/2002/debian-devel-200206/msg01187>.

FIG. 14 – Entête et corps d'un message **mail**

Il est très important de comprendre que les champs de l'entête ne sont pas certifiés corrects : ils peuvent avoir été modifiés à l'émission ou pendant le transfert. De plus, des conventions particulière peuvent provoquer des effets non évidents. Par exemple, j'ai reçu le message de la figure 14 page ci-contre, car je suis abonné à la liste de distribution (anglais: *mailing-list*) `linux-lemans`⁷¹. Mais le champ `To :` de l'entête n'est pas mon adresse, mais celle de la liste. La véritable adresse de livraison est transmise dans l'enveloppe du message, qui n'est pas visible ici. Vous pouvez cependant voir cette information reflétée dans une des lignes `Received`. En particulier lorsque vous recevez un message envoyé à une liste de distribution, ou qui vous est envoyé en copie carbone (`Cc :`) ou copie carbone invisible (`Bcc :`), vous n'aurez pas votre adresse dans le `To :`.

22.6.5 TOP domains

Les TOP domains sont les domaines terminant une adresse mail. Il existe énormément de TOP domains, dont par exemple :

edu Éducation aux États-Unis (universités).

gov Gouvernement des États-Unis.

com Organisations à but lucratif ou général.

org Organisations à but non lucratif.

net Organisations de réseaux.

codes ISO Les codes ISO à deux lettres correspondent à des pays. Par exemple :

ch Suisse.

us États-Unis.

de Allemagne.

fr France.

at Autriche.

au Australie.

jp Japon.

22.6.6 Routage mail sur des réseaux hétérogènes

Des réseaux parallèles offrent des passerelles depuis Internet jusqu'à leur système propriétaire de mail. Voici quelques indications pour que votre message arrive à bon port :

.UUCP Surtout ne pas ajouter de `.UUCP` aux sites sur le pseudo-réseau UUCP. Soit ne pas spécifier de domaine, soit trouver un adressage Internet ou une passerelle sur Internet comme dans

`user%computer@gateway.domain`

compuserve L'adresse interne est de forme `n1,n2`. Remplacer la virgule par un point et envoyer à

`n1.n2@compuserve.com`

commodore Pour le réseau ADSP, envoyer à `user@computer.adsp.sub.org`.

FidoNet ALPHANET NF administre une passerelle pour le réseau suisse. Veuillez si vous n'êtes pas sur le réseau **ALPHANET** utiliser une autre passerelle (voir le forum de news **alt.internet.services** pour la FAQ inter-réseaux).

Si l'utilisateur est :

Guillaume Tell, sur la machine 2:301/322.5

on utilisera l'adressage suivant:

`Guillaume_Tell@p5.f322.n301.z2.wfn.alphanet.ch`

On peut également atteindre des utilisateurs sous 2:301 à l'aide de l'adressage suivant :

`Guillaume_Tell@p5.f322.fn.alphanet.ch`

Enfin, pour être atteint par un correspondant **FidoNet**, il existe plusieurs possibilités:

- si l'on possède un compte sur le mips ou un alias:

alias sur 2:301/320

- sinon, votre correspondant doit envoyer un mail à

UUCP sur 2:301/320

et dans la première ligne du message il écrira:

To: votre_adresse_mail_alphanet

Il est important de noter que ces messages circulent aux frais d'ALPHANET NF. Toute utilisation commerciale ou suivie de ce service est interdite. Les envois de **mailing lists** et de fichiers sont également prohibés.

D'autres passerelles sont disponibles. Une **FAQ**, nommée *Internetwork Mail guide* est disponible dans le forum de news : **alt.internet.services**, ainsi que sur les serveurs de **FAQs**.

⁷¹voir <http://www.linux-gull.ch/>

22.7 Transfert de fichiers via mail

Il est possible de transférer des fichiers via courrier électronique privé avec **uuencode** (voir section 5.9.37 page 30) ou grâce au standard multi-format **MIME** (voir section 7.3.4 page 43). Certains logiciels propriétaires de **mail** permettent cependant d'attacher des fichiers à des messages (anglais: *file attachment*) de façon transparente. Certains utilisent le standard **MIME**, d'autres sont totalement incompatibles avec les autres logiciels.

Il faut noter qu'en général, le transfert de fichiers via **mail** doit être évité : en effet, en général le trafic mail est très coûteux (dans tous les sens du terme : puissance de calcul, charge de réseau, frais décomptés). Il vaut mieux, lorsque c'est possible, utiliser le service **FTP**. Votre administrateur local vous informera de la politique en cours sur votre site.

22.8 Caractères accentués français

Sur Internet, on utilise en général les accents **ISO Latin-1**. Pour que ces accents passent, il faut que les 8 bits de l'octet de chaque caractère soient transmis sans modification. Certains sendmail (**MTA**, Mail Transfert Agent) ont la mauvaise habitude de tronquer le 7ème bit (compté depuis 0, donc le bit de poids fort) ce qui transforme par exemple é en i.

Il y a deux solutions à ce problème, à part ne pas utiliser d'accents (caractères diacritiques) :

1. n'utiliser que des sendmails "**8 bits**"

Avantage : pas de logiciel spécifique au niveau client (transparence pour l'utilisateur).

Inconvénient irréaliste de faire changer tous les sendmail, surtout depuis qu'Internet est devenu bien commercial, il y en aura toujours un à problèmes.

2. coder les accents en 7 bits avec restauration du côté récepteur

Avantage: fonctionnera toujours quelle que soit l'horreur en face ou intermédiaire.

Inconvénient: nécessite du logiciel spécifique des deux côtés dans le **MUA** (Mail User Agent).

Une réalisation de la deuxième solution est **MIME**, *Multimedia Internet Mail Extension*, qui est supportée sous UNIX par `metamail`, `elm` et la plupart des lecteurs de mail et d'autres lecteurs sous environnements non UNIX.

Notez qu'au niveau des **news**, du moins dans la hiérarchie fr, la plupart des sites sont bien configurés (à part des systèmes commerciaux comme **FirstClass** ou divers clients mal configurés) pour permettre l'émission d'accents tels quels, **MIME** y est donc découragé pour son côté verbeux.

23 News (messages publics)



FIG. 15 – Graphe de connectivité mondiale Usenet

23.1 Introduction

Le terme news est l'équivalent de messagerie électronique publique. Les news sont des forums d'échanges qui sont regroupés en **newsgroups**, comme par exemple **comp.sys.amiga.announce**. Les noms de forums sont organisés hiérarchiquement, en parcourant le nom du forum de gauche à droite.

USENET est par définition l'ensemble des sites propageant des hiérarchies de news selon le document **RFC-1036** et ses dérivés. Chaque site ou serveur de news choisit quelles sont les hiérarchies qu'il désire transporter et/ou transmettre. Chaque site choisit également des politiques d'expiration et d'accès aux forums.

Certains serveurs sont très mal connectés ou ne disposent pas d'une configuration matérielle suffisante : un **full-feed** de news utilise environ 2GByte d'espace disque par jour, ce qui absorbe en général complètement une ligne 256 kbps permanente, et nécessite de 256 MByte à 1 GByte de mémoire sur un ordinateur puissant. (chiffres : milieu 2002)

Certains serveurs sont dits publics : cela signifie qu'ils ne sont pas réservés en général à une organisation particulière, et qu'ils sont donc pratiquement ouvert à quiconque en lecture (parfois en écriture). La plupart de ces serveurs sont gérés bénévolement et ne permettent que rarement l'accès en écriture (post) d'articles pour des raisons évidentes. On parle souvent de serveurs **NNTP** publics, NNTP étant un des protocoles de transport de news. Les puristes diront en général **NNRP** si l'accès se fait de façon interactive.

La plupart des serveurs de news gérés par des fournisseurs d'accès à Internet sont incomplets, mal gérés et peu fiables : il semble que seuls les serveurs à but non lucratif (en général des entreprises ou des universités) soient utilisables en 2002. Quelques fournisseurs de services (que l'on paie en général un peu plus cher) offrent de bons services de news.

L'audience d'un forum est très variable, à la fois en nombre et en types d'auditeurs. Pour cette raison il est conseillé de s'imprégner de l'atmosphère d'un forum avant de participer à une discussion quelconque. Dans certains cas, particulièrement les forums très spécialisés, il faut lire le résumé des réponses aux questions fréquemment posées (**FAQ**, de l'anglais: *Frequently Asked Questions*, voire du français : *Foire Aux Questions*).

Tout comme pour le courrier privé, de nombreux logiciels existent pour consulter les **news**, on citera entre autres **tin**(voir section 23.7.1 page 83), et **slrn**.

23.2 La modération

23.2.1 Introduction

La modération d'un **forum** de **news Usenet** vise à garantir une certaine qualité des interventions. Les rédacteurs postent un message normalement dans le forum modéré (ou envoient le message en courrier privé (**mail**) au modérateur, si leur site est mal configuré), mais celui-ci n'est pas rendu disponible avant d'être approuvé par le **modérateur** (ce dernier peut refuser de façon motivée un article, s'il ne correspond pas à la **charte** du forum). Peu de forums sont modérés, mais ce sont souvent les plus intéressants (notamment ceux finissant par `.announce`).

23.2.2 Le modérateur

Le modérateur est le garant du contenu du forum qu'il modère, il sert notamment à éliminer le bruit et ne conserve que le signal (les informations utiles), en supprimant notamment les messages non conformes avec la **charte** définie à la création du forum. Il vérifie également que les soumissions soient mises en page correctement le cas échéant. Le travail du modérateur prend parfois beaucoup de temps (**news.announce.newgroups** par exemple). Une équipe de modérateurs peut travailler en groupe : c'est par exemple le cas pour **news.answers**, où sont postées toutes les **FAQs** conformes à l'esprit **Usenet**.

On distingue différentes catégories de modérateurs :

modérateur approuveur C'est le cas des forums tels que **news.answers**. Le modérateur approuve le contenu des articles soumis (et postés régulièrement) et l'indique à l'auteur. C'est à ce dernier que revient le privilège de poster régulièrement l'article selon les indications du modérateur.

modérateur posteur Il lit les articles reçus et les rejette ou accepte de les poster, son travail se limite à ces actions. Il peut utiliser un script tel que `moderate` pour lui faciliter la vie. C'est par exemple ce qui se fait dans **fr.announce.important**.

modérateur posteur-contrôleur Il garde des pointeur sur ce qui se dit afin d'éviter la duplication de l'information ou des incohérences, éventuellement il poste des résumés d'activité. Un exemple, dans la hiérarchie `fr`, est le forum:

fr.announce.newgroups.

On distinguera particulièrement le modérateur au sens **Usenet** et celui au sens **FTN**, ce dernier étant plutôt un animateur. De ce fait, les modérateurs d'**Usenet** sont soumis à un travail énorme. Il n'est donc pas inhabituel que les modérateurs renoncent à leur travail ou se fassent seconder.

23.2.3 Modérer ou ne pas modérer ?

Rappelons que la création d'un forum est soumise à des règles strictes, qui ménagent un temps de discussion (anglais: *Call For Discussion* ; **CFD**) pour la définition de la **charte** du **forum** (propositions de contenu et de modération éventuelle). Ces règles sont postées mensuellement pour la hiérarchie `fr` dans le forum **fr.usenet.groups**.

La modération n'est pas une censure : il existera toujours la possibilité de discuter des pratiques du modérateur dans un forum approprié (comme **fr.usenet.divers**). De fait, les forums modérés sont souvent des forums techniques (par opposition à des forums où l'on échange des opinions qui sont difficilement modérables au sens Usenet).

23.3 Catégories

Les différentes catégories de forums disponibles se différencient par leur nom. Leur structure est similaire à un **système de fichiers**⁷², si l'on remplace les points par ce qui sépare les répertoires (sous UNIX: `/`).

comp Forums concernant l'informatique.

misc Sujets divers.

rec Activités récréatives.

gnu État du projet GNU.

alt Réseau alternatif.

news Administration.

biz Commercial.

sci Sciences.

soc Social.

talk Discussions-débats.

clari Forums de dépêches d'agences (commercial, non disponible).

ch Réseau suisse.

de Réseau de langue allemande.

fr Réseau de langue française.

⁷²d'ailleurs les news, sous UNIX sont disponibles traditionnellement dans `/usr/spool/news`.

23.4 Sélection de quelques forums

Cette section décrit quelques forums intéressants par leur contenu ou leur qualité. Régulièrement, des utilisateurs postent sur le forum `alphanet.misc` leurs forums préférés avec une courte description. La liste plus ou moins complète de tous les forums disponibles avec une très courte description est disponible dans le fichier

```
/usr/lib/news/newsgroups.all
```

ou par hiérarchies dans le répertoire

```
/usr/lib/news/hierarchies
```

et une liste complète des forums disponibles localement dans

```
/usr/lib/news/newsgroups
```

Ces fichiers sont des fichiers textes et donc peuvent être affichés avec **more**, ou édités⁷³ avec **vi** ou **Emacs**. Voici donc une liste de quelques forums importants, qui devraient être lus par tout nouvel utilisateur de news.

news.announce.newusers Explanatory postings for new users. (Moderated)

fr.announce.newusers Annonces destinées aux nouveaux utilisateurs. (Moderated)

fr.usenet.reponses Réponses à des questions fréquemment posées. (Moderated)

Notez que les forums marqués **Moderated** ne peuvent qu'être lus. Les éventuelles contributions sont à soumettre via **mail** au modérateur du forum, qui pourra les publier ou les rejeter.

N'oubliez pas que chaque hiérarchie a ses usages : lisez chaque forum avant de poster.

23.5 Savoir-vivre

Les règles de savoir-vivre du courrier électronique privé (*voir section 22.2 page 71*) s'appliquent aussi aux messages publics. Cependant, par la nature des **news** qui est de toucher un vaste public, il s'agit d'éviter les pièges supplémentaires qui sont exposés ci-après.

23.5.1 Comment rédiger

Poster des **articles** (messages publics) sur les **forums** de **news** est très simple. Cependant, il est aussi très simple de ne pas utiliser ces possibilités de façon correcte ou admissible par vos lecteurs. Donnez-vous un peu de temps pour vous habituer à tout ce qui composera un article de qualité. Avant de poster, il est bon de lire le forum concerné pendant quelque temps afin de s'imprégner de l'atmosphère et d'éviter de répéter les erreurs des autres. En général, les conseils suivants sont considérés comme acceptables :

- L'accès aux news n'est pas un droit fondamental. Il peut d'ailleurs vous être retiré par votre fournisseur de service en cas d'abus.
- Ne pas oublier que votre correspondant est humain.
- Ne sous-estimez pas la portée de votre article : un ami ou votre futur employeur est peut-être dans l'assistance.
- Faire attention à ce qu'on dit à propos d'autres utilisateurs et à propos de matériel (sens général) soumis à licence.
- Être bref et concis.
- Faire bonne impression, s'appliquer à présenter correctement un article, il n'en sera que plus apprécié et compris.
- Utiliser le champ Subject: (sujet de l'article) de façon appropriée. Il est de bon ton que celui-ci ait un rapport étroit avec le contenu de l'article, particulièrement s'il s'agit d'une réponse.
- Adapter votre article aux destinataires.
- Marquer clairement les passages sarcastiques. Évitez les passages trop confus et les sous-entendus peu clairs qui peuvent nuire à la compréhension de votre article.
- Éviter les signatures gigantesques. Il est usuellement considéré qu'une signature dépassant 4 à 5 lignes ne fera qu'agacer vos lecteurs potentiels. Après tout, une signature, comme son nom l'indique, ne sert qu'à renseigner les lecteurs potentiels sur votre identité, éventuellement sur votre situation. A quoi bon y ajouter des dessins gigantesques ou des citations de dix lignes.
- Limiter la longueur des lignes à une valeur raisonnable. La plupart de vos lecteurs utilisent, malgré eux cela va sans dire, des terminaux limités à 80 colonnes et souffriraient de ne vous comprendre que partiellement. Une valeur de 70 caractères est conseillée afin de permettre une citation aisée et respectant elle-même encore ce critère.
- Éviter les caractères de contrôle.
- Contrairement aux hiérarchies générales, les forums des hiérarchies **fr** tolèrent les accents **ISO Latin 1**. Aucun autre codage des accents n'est toléré pour l'instant. En particulier, les accents des systèmes **MS-DOS**, **Macintosh**, les accents que l'on trouve dans les documents **MIME** ou **HTML** ne sont pas tolérés : en effet, ces formats nécessitent des logiciels spécifiques. En cas de doute, consulter la **FAQ** de **fr.news.8bits**. Cela est valable également pour toute signature ajoutée automatiquement à votre article.
- Évitez de poster des binaires (uencodés, MIME et divers) dans les forums non spécifiquement réservés pour cela (*.binaries). Mettez plutôt une référence à votre serveur ftp ou WWW ou proposez d'envoyer les binaires par courrier électronique privé.

⁷³En général, vous n'aurez pas accès en écriture sur ces fichiers.

- Ne pas poster un message reçu en courrier privé (**mail**) sur les **news** sans l'accord de son auteur, car le correspondant a le droit à sa sphère privée.
- Ne poster un article qu'une seule fois. Il vaut mieux **cross-poster** que de poster un article plusieurs fois. Vous vous exposez à divers désagréments si vous postez plusieurs fois le même article aux frais de la collectivité.
- Ne pas reposer immédiatement un article parce que vous n'arrivez pas à le lire. Certains systèmes de **news** mettent un certain temps avant de rendre les articles disponibles. Si vous avez ce genre de problèmes, postez dans **fr.test** (afin de recevoir des quittances de réception après quelques jours). Avant tout, patience : un article posté plusieurs fois fait très mauvais effet.
- Un stage de lecture dans un **forum** peut être utile avant de poster quoi que ce soit.
- Contrôler la distribution Distribution: et la liste des forums Newsgroups:. Sachez cependant que le champ Distribution: n'est pas vérifié sur la plupart des sites : le concept de distribution n'est pas fonctionnel.
- Ne poster un article à consonance commerciale que si le bénéfice que les lecteurs potentiels peuvent en retirer est plus important que le vôtre, et cibler celui-ci sur les forums vraiment en rapport avec le sujet de l'article. Les hiérarchies **biz** et **fr.biz** sont particulièrement appropriées pour ce genre d'articles. Sachez cependant qu'elles obéissent à des règles très strictes : la lecture des **FAQs** spécifiques s'impose.
- Ne pas hésiter à coder un article dont le contenu peut froisser certains lecteurs (**rot-13**). Évitez cependant dans la mesure du possible tout article choquant à part dans certains forums réservés comme **fr.rec.humour**.
- Ne jamais écrire d'article lorsqu'on est excédé, énervé, ou sous l'influence d'un breuvage (solide, liquide ou autre) quelconque :-;
- Lorsqu'une question peut intéresser plusieurs lecteurs et que la réponse est catégorique (une discussion technique a en général une réponse catégorique), n'hésitez pas à demander que les articles vous soient envoyés personnellement afin de rédiger un résumé (anglais: *summary*) des réponses, que vous posterez dans le forum concerné. Vous pouvez utiliser le champ Followup-To: à cette fin en y mettant poster.
- Avant de poster dans un forum, lisez la documentations (**FAQs**) associée et le forum lui-même pendant quelque temps. Ensuite, vous pouvez poster votre premier article dans un forum de test (**fr.test** par exemple). Celui-ci sera automatiquement confirmé. Essayez également d'émettre des messages de **cancel** sur votre article (voir la documentation de votre lecteur de news ou le **RFC-1036** [30]). Un message de **cancel** permet de supprimer un article après son émission. En cas de problème, contactez tout d'abord votre administration système (usuellement *postmaster*).
- Ne tombez pas dans le piège de vouloir écrire sur tout et partout. Vous risquez le discrédit si vous vous faites remarquer par le volume plutôt que par la qualité de vos interventions.
- Mettez-vous le plus souvent possible dans la peau du lecteur et posez-vous la question de savoir si votre article est approprié.

23.5.2 Comment réagir

Répondre à un article en public (anglais: *follow-up*) est similaire à poster un article initial : cependant, il faut comprendre qu'en général, les réponses à un sujet donné sont souvent identiques et multiples, surtout lorsque la question posée était évidente (parce que souvent répétée, par exemple). L'aide-mémoire suivant peut vous aider à réagir au mieux après un article. Il s'applique en particulier à la lecture d'un article abusif (**spam** ou **scam**).

- Ne pas croire tout ce qu'on lit dans les news.
- Ne pas sur-réagir : ce n'est pas parce qu'un article vous semble offensant qu'il faut réagir tout de suite. Après tout, vous n'avez peut-être pas compris le sens de l'article. Il se peut que le rédacteur supposé ne soit pas l'auteur réel de l'article. Vérifiez également que personne n'a déjà réagi en public (**news**). Dans tous les cas, il faut éviter de poster une réponse publique à un article sauvage mais répondre à son auteur en courrier électronique privé (**mail**).
- Ne pas critiquer les administrateurs de systèmes étrangers pour le comportement de leurs utilisateurs. Si vous devez le faire car les abus sont répétés ou manifestes, faites-le en courrier privé en motivant vos griefs (en donnant des références d'articles, etc.).
- Ne pas critiquer autrui sur la forme, mais sur le fond.
- Lire toutes les réponses et éviter de redire ce qui a déjà été dit. Lorsque la question est évidente, attendez quelques jours avant de répondre ou répondez en courrier électronique privé.
- Ne pas fustiger (ou descendre en flammes) en public quelqu'un pour une question évidente ou fréquemment posée. Au contraire, répondre en courrier électronique privé (**mail**) en donnant, si besoin est, des références précises à des documents (**FAQs**). Une question évidente ne l'est pas forcément pour tout le monde.
- Citer, ne pas reposer en entier un article juste pour y ajouter quelques lignes de commentaires. Particulièrement, les signatures n'ont aucune raison, en général, d'être citées. Si la question vous intéresse également, ne postez pas un article demandant à l'auteur de vous communiquer les résultats de sa recherche : au contraire, envoyez-lui un message privé (*r* au lieu de *f* dans *tin*) lui demandant de vous informer, ou mieux, de poster un résumé des réponses (anglais: *summary*). Ne tombez pas dans l'excès inverse et ne mélangez pas les interventions en vous trompant de citation.
- En général, le caractère de citation sera > et on laissera une ligne vide entre le texte cité et son propre texte pour améliorer la lisibilité afin d'obtenir quelque chose du genre :
> la transmission série à 2400 bps à la vitesse de 2400/8 cps

non, plutôt 2400/10, soit 240 cps.

L'usage de séquences de citations plus longues préterite la lisibilité (elle dépend du nombre de caractères de la ligne). Notez que la plupart des lecteurs de news gèrent correctement ce problème (ainsi que la longueur de ligne).

- Vérifier les champs Followup-To: et Newsgroups: si le cross-postage était abusif et ne laisser que les forums concernés.
- Si vous êtes responsable d'un article abusif ou que vous avez commis une erreur en envoyant un article incomplet ou incorrect, vous pouvez détruire cet article en émettant un message de **cancel**. La plupart des lecteurs de news offrent une telle option. Par exemple, dans `tin`, c'est la commande `D`. D'autres lecteurs de news offriront peut-être des commandes telles que *Cancel Article* ou *Kill*. Cela sera bien plus efficace que des excuses en public, utilisant encore plus de bande passante réseau. Si vous réagissez suffisamment rapidement, vous pourrez poster une version corrigée de votre article sans que personne ne voie l'article initial. De toute manière, le message de **cancel** se propagera de la même façon que votre article initial et le détruira où qu'il se cache dans la plupart des cas.
- Il ne sert à rien de répondre publiquement aux articles anonymes qui provoquent votre indignation de façon calculée (racisme, etc.). Envoyez plutôt une copie de l'article ainsi qu'un paragraphe d'explication motivé et posé (sans hargne inutile) au postmaster du site expéditeur (`postmaster@site_concerné`) afin de lui expliquer la situation. En général, cela est bien plus efficace que de créer une discussion parallèle dans un forum.

Vous trouverez dans les forum de news **news.announce.newusers** et **fr.announce.newusers** des **FAQ** à ce sujet, comme *Emily Postnews*, qui est une compilation humoristique des faux-pas à ne pas faire, également traduite en français.

23.6 Que faire en cas d'abus ?

Les **news** étant un moyen de contacter beaucoup de gens rapidement par groupe d'intérêt, de nombreux abus sont possibles :

- poster des articles à contenu publicitaire⁷⁴, déguisé ou non,
- poster des articles choquants (parce qu'inappropriés ou franchement révoltants, ou encore illégaux (**scam**)),
- contacter les participants par **mail**,
- diverses formes de **junk mail** (courrier non sollicité, très mal vu),
- **spam** ou cross-postage abusif.

Avant de considérer un article comme abusif, considérez les points suivants :

- êtes-vous sûr d'avoir bien lu ?
- l'auteur est-il celui qu'il prétend ?
- avez-vous dû décoder l'article (**rot-13**) ?

Il faut comprendre que ces actions sont vues comme des abus car le réseau **Usenet** est encore en grande partie administré de façon bénévole et les participants paient pour leur accès en temps et/ou en argent. Accepteriez-vous de payer pour la publicité qui remplit votre boîte aux lettres ?

Il existe bien des moyens de faire comprendre à une entreprise ou un particulier qu'il n'a pas intérêt à recommencer. En général, si son accès se trouve chez un des grands fournisseurs de service commerciaux, on contactera l'administration système (usuellement `postmaster@adresse`). Notez que cela entre en contradiction avec les règles de savoir-vivre, à juste titre : les fournisseurs de services commerciaux sont payés pour leur travail, au contraire des administrateurs des sites bénévoles ou de recherche. Cela est cependant bien plus efficace (et surtout bien moins coûteux) que de répondre en public à un article abusif : en général, les dégâts ne sont pas causés par l'article initial mais par les centaines d'articles qui viennent par la suite.

Il est également très efficace de simplement ignorer l'article et de laisser à d'autres que vous le soin de s'en occuper : les administrations de certains sites commerciaux sont très compréhensives et s'occupent elles-même de châtier les coupables.

Certains lecteurs de news permettent de cacher les articles en provenance de certains utilisateurs, de certains sites ou par sujets (anglais: *kill-file*).

23.7 Interface utilisateur

Il existe une pléthore de logiciels d'interface aux **news**. Les principales différences entre les différents clients sont surtout la possibilité de groupage par sujet (anglais: *threading*) qui n'est pas offerte par tous les logiciels et la possibilité d'interagir de façon graphique sous **X11** (voir section 16 page 59). Cette section ne listera que les logiciels disponibles sous UNIX : il en existe également sur les micro-ordinateurs pour lecture via **NNTP** ou depuis un **spool** local (utile pour la lecture hors-ligne ; anglais: *offline newsreader*).

23.7.1 tin : interface VT100

tin est un lecteur de news orienté terminal semi-graphique (VT100) permettant de se déplacer dans les forums de discussion (**newsgroups**), les articles et les thèmes. Son nom est un acronyme (anglais: *threaded interactive newsreader*) traduisible en français par *lecteur de news interactif orienté thèmes* : en effet, l'avantage principal de **tin** par rapport aux autres logiciels lecteurs de news est qu'il groupe les articles par sujet et les présente d'une manière attrayante. Comme autres avantages citons la présentation unique des articles cross-postés et l'extraction automatique dans les forums de sources ou de binaires. On lance le programme en tapant `tin` dans un **shell**. L'option `-U` permet de lire les **news** en même temps que **tin** prépare à l'avance l'indexation. L'option `-r` ainsi que la variable d'environnement **NNTPSERVER** permettent de lire les news à distance⁷⁵ via

⁷⁴ Attention, cela ne veut pas dire que toute publicité est interdite sur **Usenet**. Il existe des forums spéciaux (principalements dans les hiérarchies **biz** et **fr.biz**) réservés pour les articles commerciaux. Il faut savoir que les méthodes habituelles de publicité sont relativement mal vues. Il faut s'en tenir aux faits objectifs. La meilleure solution est de s'adresser à un forum modéré, ce qui vous fournira une large audience et vous permettra de corriger des erreurs de forme ou de contenu avec les conseils d'un modérateur. Avant de poster, lisez le ou les forums concernés et les **FAQs** concernées.

⁷⁵ Notez qu'il vaut mieux ne pas utiliser plusieurs serveurs mais bien un seul. En effet, **tin** stocke des informations quant aux articles lus dans le fichier **.newsrc**. Ces informations ne sont pas transportables d'un site à l'autre.

NNTP.

Tin répond à des commandes par touches (sans appuyer sur *RETURN* sauf aux prompts >). Les touches *TAB* et *q* sont valables dans tout le logiciel (à part aux prompts) et désignent respectivement le passage à l'article suivant non lu (ce qui permet une lecture de tous les nouveaux articles dans tous les forums par appui répétitif sur *TAB*) et la sortie du niveau courant. A chaque fois qu'une liste de forums, d'articles ou encore de rédacteurs est affichée, il est possible de se déplacer sur la ligne désirée avec les touches du curseur, ou de taper simplement le numéro de cette ligne (disponible dans la colonne de gauche), suivi de *RETURN*. Le logiciel est structuré en quatre niveaux d'interaction. Pour passer d'un niveau à l'autre on utilise *RETURN* pour entrer et *q* pour sortir.

liste des forums Ce niveau permet de visualiser tous les forums disponibles (commande *y*), de s'abonner ou de renoncer à certains forums (*s* et *u*), d'afficher ou de cacher les descriptions des forums (*d*), ainsi que d'effectuer diverses opérations de configuration du logiciel et de l'interaction. On peut aussi aller à un forum particulier à l'aide de la commande *g*.

liste des thèmes d'un forum Ce niveau affiche les différents **threads** ou thèmes en cours dans le forum. On visualise le premier article d'un thème avec *RETURN* (en se déplaçant bien sûr préalablement sur l'article qui nous intéresse comme indiqué plus haut). Notez que celui-ci peut ne plus exister dans la base de données de **news** locale : un message approprié sera affiché en bas de l'écran. On peut aussi aller au premier article non lu avec **TAB**, et visualiser l'ensemble des articles d'une discussion avec *l*.

liste des articles dans un sujet particulier (thread)

affichage d'un article proprement dit On tape *ESPACE* pour passer à la page suivante, *b* pour revenir à la page précédente, *s* pour sauver un article (par défaut dans le répertoire *News/*), *m* pour envoyer une copie d'un article via **mail** à un tiers, *r* pour répondre en **mail** (courrier privé) à un article, ou encore *f* pour répondre en **news** (public) à l'article. On sort évidemment de ce niveau avec *q*. On passe à l'article non lu suivant avec *TAB*. On peut détruire un article (anglais: *cancel*) avec *D*.

Lorsqu'une demande (prompt) intervient, on peut annuler l'opération à l'aide de la touche *ESCAPE*.

tin étant un outil très puissant, il est bon de se référer à la documentation (*voir section 5.9.20 page 28*).

23.7.2 knews, interface X11

knews est un lecteur de news sous X11, disponible avec source, et utilisant la bibliothèque standard X. Il est très rapide si le serveur dispose des extensions **XOVER**, comme **INN**. Il permet vraiment de traiter les sujets par thème (anglais: *threading*), d'afficher l'arborescence des réponses, de gérer des **kill-files**, etc.

23.7.3 Logiciels clients pour ordinateurs personnels

En plus des logiciels disponibles sous environnements UNIX et de la possibilité de les lancer à travers le système **X11**, il existe de nombreux clients sur ordinateurs personnels permettant la lecture et l'envoi d'articles de **news**. La plupart présupposent une connexion interactive (permanente ou non). Certains peuvent être utilisés avec une couche permettant de compresser ou d'automatiser les transferts (comme **UUCP**). Certains **browsers WWW** permettent également de lire les news sur un serveur **NNTP**. La table 10 liste ces clients pour différentes plates-formes.

Plate-forme	nom logiciel	accessibilité	type
Amiga	Grn	shareware	spool/client NNTP
	tin	libre	spool/client NNTP
MacOS	NewsWatcher	libre	client NNTP
	rnMac	libre	spool/client NNTP
MS-WINDOWS	WinVN	shareware	client NNTP
	FreeAgent	libre	client NNTP
OS/2	tin	libre	spool/client NNTP

TAB. 10 – Clients **news** pour ordinateurs personnels

23.7.4 Les archives

Certains forums sont archivés sur des serveurs **FTP**. Il existe cependant un serveur qui indexe et archive tous les forums : <http://www.dejanews.com/>, maintenant racheté par Google, un excellent moteur de recherche, à l'URL : <http://www.google.com/>

23.8 Aspects techniques**23.8.1 Transfert**

Les news sont transférées en général sur Internet via le protocole **NNTP**. Les délais de transmission sont assez faibles, même si le volume de news est de plus en plus grand (milieu 2002 : de l'ordre de 2 GByte par jour pour un site complet (anglais:

full-feed), dont 80% de **alt.binaries**). En général, la transmission se fait en “tâche de fond” durant toute la journée. Le protocole **NNTP** permet d’interconnecter plusieurs sites ensemble, créant donc un graphe avec cycles et permettant une redondance en cas de panne : les news fonctionnant donc par redondance, il est fort peu probable de perdre des messages ... sauf si les lignes physiques sont surchargées (ou les serveurs de news trop lents) et que le retard accumulé atteint la limite de durée de stockage des news (qui varie de site à site). Un système efficace de détection de doublons (anglais: *duplicates*) permet d’éviter l’apparition d’articles en plusieurs exemplaires⁷⁶. La propagation d’un article de **news** se fait donc par inondations successives de divers sites. La figure 15 page 79 donne une idée des liens mondiaux entre serveurs de news.

23.8.2 Sécurité

Les **news** n’assurent pas l’identité du rédacteur. Cependant, il est parfois possible de retrouver la provenance d’un article grâce au champ `Path:` . Il faut cependant savoir que ce champ est établi à l’aide d’une authentification de type **source address check** (voir section 20.1.3 page 68). De plus, les articles transmis peuvent être modifiés par un pirate (anglais: *cracker*). Une certaine authentification peut être obtenue à l’aide de logiciels comme **PGP** (voir section 7.6.1 page 44) (les fameuses signatures PGP de certains articles de news, considérées par certains comme du *gaspillage de bande passante*).

24 FTP

24.1 Introduction

FTP (anglais: *File Transfer Protocol*) permet d’échanger des fichiers entre sites distants. Il nécessite une connexion permanente entre les deux sites (ou éventuellement une connexion non permanente via **SLIP** ou similaire). Il est cependant possible également d’utiliser des passerelles à travers le courrier électronique. (voir section 21.1 page 70)

Une base de données mondiale permet de faire des recherches de fichiers, c’est le système **archie** accessible, entre autres, par le client **xarchie** sous interface graphique **X11** (voir la figure 16), ou via la page **WWW** à l’URL

<http://www.urec.fr/archie.html>

On peut aussi y accéder via le client en ligne de commande **archie** ou encore directement via **telnet** sur un des serveurs, comme **nic.funet.fi**, en entrant le **login** **archie**.

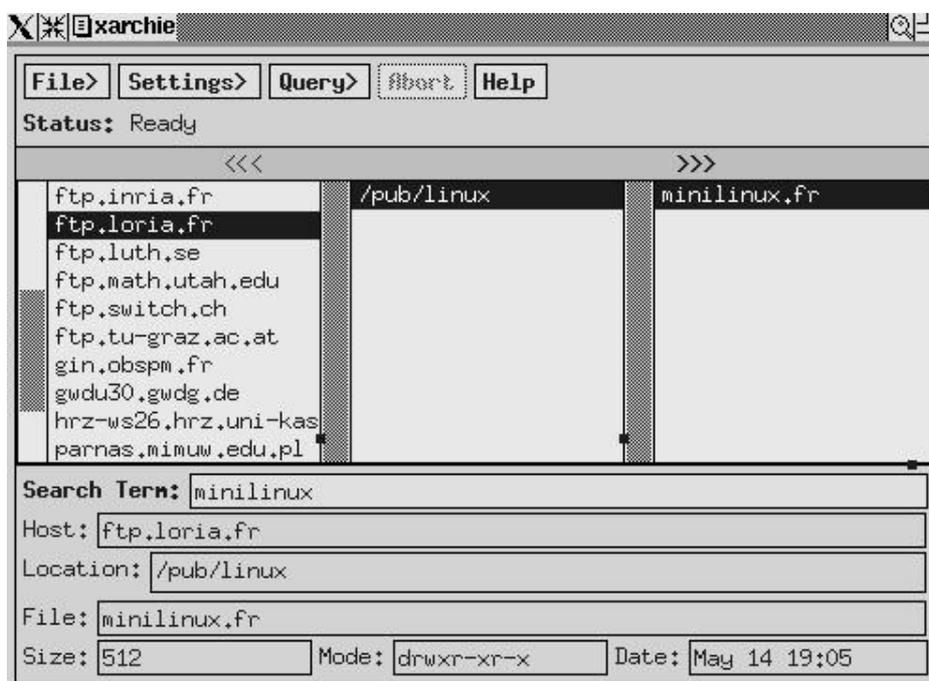


FIG. 16 – Fenêtre typique **xarchie** sous **X11**

24.2 Logiciels clients

Il existe de très nombreux logiciels pour accéder à des sites **FTP** distants. Certains utilisent directement le protocole **FTP**, comme le client **ftp** de la ligne de commande. (voir section 24.5 page suivante)

D’autres interfaces de plus haut niveau (comme **ftptool**, **xarchie**) existent. On peut aussi utiliser la plupart des clients **WWW** comme client **FTP**. (voir section 28 page 91)

⁷⁶De nombreux articles à double, triple ou même plus apparaissent parfois dans les forums : ce sont des utilisateurs mal informés et peu patients utilisant des outils non appropriés (p.ex. **Netscape** ou des passerelles vers d’autres technologies ne supportant que mal ce concept).

Sous UNIX, il est également possible d'utiliser un système de fichier auquel n'importe quel utilitaire pourrait accéder, de manière similaire à **NFS**, aux fichiers stockés sur un serveur **FTP** distant, avec la possibilité d'optimiser l'utilisation par un système de cache. Ce logiciel se nomme **Alex**.

24.3 Utilisation

On accède à un serveur **FTP** en spécifiant soit l'utilisateur anonyme public `anonymous` ou `ftp`. Dans ce cas, si l'accès public est autorisé sur le site distant, le système vous indiquera d'une manière ou d'une autre qu'il accepte votre requête et il vous demandera un mot de passe : entrez simplement votre adresse de courrier électronique (et non pas un mot de passe et surtout pas votre mot de passe utilisé sur votre système **UNIX**).

Dans certains cas, la machine distante refusera votre connexion (par exemple s'il y a déjà trop d'utilisateurs ou que les sessions ne sont tolérées que pendant une certaine période de la journée).

L'autre possibilité, si l'on possède un accès (compte) sur la machine distante, est d'utiliser son nom d'utilisateur et son mot de passe.

N'oubliez pas que beaucoup de sites **FTP** sont maintenus de façon bénévole et que votre utilisation charge le réseau : évitez les heures de travail *locales* à la machine distante et n'hésitez pas à perdre quelques minutes à consulter un service de recherche comme **archie** qui vous permettra de localiser le site le plus proche. La version **X11** permet d'ailleurs également de transférer automatiquement des fichiers depuis des sites FTP anonymes.

24.4 Problèmes courants

Une fois connecté, si votre logiciel vous propose cette option, vous devez en général sélectionner le mode de transfert *binnaire*, surtout pour les archives ou les exécutables.

Sur les serveurs FTP anonymes (ou publics, l'anonymat n'est souvent plus toléré) les fichiers se trouvent en général dans le sous-répertoire `pub/`.

De plus en plus de serveurs **FTP** sont surchargés et il arrive de temps à autres (de plus en plus souvent, malheureusement) que des transferts en cours soient interrompus. Dans ce cas, il peut être intéressant de continuer le transfert où celui-ci s'est interrompu. On utilisera pour ce faire la commande `restart` avec en argument le nombre d'octets déjà reçus (ceci pour la commande `ftp`, d'autres interfaces ont d'autres options). Notez que ceci est un ajout récent au protocole **FTP** et n'est donc pas supporté par tous les clients ni tous les serveurs.

24.5 La commande `ftp`

Cette commande très simplifiée permet de faire du transfert de fichiers depuis la ligne de commande. Il est possible d'exécuter des commandes **shell** depuis `ftp` en les précédant de `!`. Évidemment, pour changer de répertoire local cela ne fonctionne pas (un sous-shell est créé, change de répertoire et quitte, donc perd le nouveau répertoire). On utilisera alors `lcd`.

Exemple: Voici un exemple d'utilisation du FTP anonyme (anglais: *anonymous FTP*) avec la commande `ftp`

```
<mips> schaefer:/d2/users/schaefer> ftp ftp.alphanet.ch
Connected to ftp.alphanet.ch.
220 mips FTP server (from 4.3BSD Tahoe) ready.
Name (ftp.alphanet.ch:schaefer): ftp
331 Guest login ok, send ident as password.
Password: schaefer@alphanet.ch
Il est d'usage de donner son adresse mail comme réponse. Surtout pas votre mot de passe!
230 Guest login ok, access restrictions apply.
ftp> bin
200 Type set to I.
Je me suis mis en mode 'transparent' ; pour lister des répertoires on utilise ls ou dir
ftp> ls
200 PORT command successful.
150 Opening data connection for /bin/ls (193.72.186.3,1263) (0 bytes).
bin
etc
incoming
ls-laR
ls-laR.gz
private
pub
226 Transfer complete.
53 bytes received in 0.36 seconds (0.15 Kbytes/s)
Pour se déplacer : cd. Notez que c'est lcd pour se déplacer dans des répertoires locaux.
ftp> cd pub
```

```

250 CWD command successful.
160 bytes received in 0.16 seconds (1 Kbytes/s)
ftp> cd archives
250 CWD command successful.
ftp> cd gfx/postcards
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening data connection for /bin/ls (193.72.186.3,1267) (0 bytes).
bachalpsee.jpg
baden.jpg
castle.jpg
chillon.jpg
suisse.gif
226 Transfer complete.
144 bytes received in 0.21 seconds (3.5 Kbytes/s)
prendre un fichier
ftp> get suisse.gif
200 PORT command successful.
150 Opening data connection for suisse.gif (193.72.186.3,1268) (9986 bytes).
226 Transfer complete.
local: suisse.gif remote: suisse.gif
9986 bytes received in 0.039 seconds (2.5e+02 Kbytes/s)
Prendre plusieurs fichiers en supprimant les confirmations :
ftp> prompt
Interactive mode off.
ftp> mget *.jpg
200 PORT command successful.
150 Opening data connection for bachalpsee.jpg (193.72.186.3,1272) (53178 bytes).
226 Transfer complete.
local: bachalpsee.jpg remote: bachalpsee.jpg
53178 bytes received in 0.16 seconds (3.3e+02 Kbytes/s)
200 PORT command successful.
150 Opening data connection for baden.jpg (193.72.186.3,1273) (35954 bytes).
226 Transfer complete.
local: baden.jpg remote: baden.jpg
35954 bytes received in 0.17 seconds (2e+02 Kbytes/s)
200 PORT command successful.
150 Opening data connection for castle.jpg (193.72.186.3,1274) (62096 bytes).
226 Transfer complete.
local: castle.jpg remote: castle.jpg
62096 bytes received in 0.29 seconds (2.1e+02 Kbytes/s)
200 PORT command successful.
150 Opening data connection for chillon.jpg (193.72.186.3,1275) (61728 bytes).
226 Transfer complete.
local: chillon.jpg remote: chillon.jpg
61728 bytes received in 0.26 seconds (2.3e+02 Kbytes/s)
200 PORT command successful.
De manière similaire, on envoie des fichiers avec put/mput et on peut aussi afficher des données textes
ftp> cd ../screenshots
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening data connection for /bin/ls (193.72.186.3,1285) (0 bytes).
INDEX
README
admin
gif
iff
226 Transfer complete.
32 bytes received in 0.074 seconds (0.42 Kbytes/s)
Réception directe via un pipe

```

```
ftp> get README |more
200 PORT command successful.
150 Opening data connection for README (193.72.186.3,1286) (462 bytes).
These are some pictures of my Amiga screen under X-Window, networked with
a UNIX computer. The first and second pictures show common X11 utilities
and the third a Mosaic/WWW connection to the Internet. There are other
pictures. See the descriptions in INDEX.
```

```
schaefer@alphanet.ch
226 Transfer complete.
local: |more remote : README
462 bytes received in 0.81 seconds (0.56 Kbytes/s)
ftp> quit
221 Goodbye.
```

24.6 Quelques sites FTP

Il est difficile de donner une liste à jour des sites **FTP** : en effet, une liste serait presque immédiatement obsolète dès que publiée. Les forums de **news** comme **alt.internet.services** et les services de recherches **archie** et sur **WWW** permettent en général de localiser des sites.

Voici tout de même une liste de quelques sites **FTP** avec description. N'oubliez pas que les sites situés outre-atlantique doivent toujours être considérés en deuxième priorité.

ftp.wustl.edu Un très grand site FTP, contenant des logiciels pour toutes les plateformes, comme par exemple le réseau **Aminet** de logiciels **Amiga**, ainsi qu'une très complète archive d'images et de documents divers. Ce site est bien entendu surchargé. Il s'agit de l'Université de Washington, USA.

liason3.epfl.ch Un site de l'École Polytechnique Fédérale de Lausanne, en Suisse. Il contient de nombreux logiciels **GNU**, une distribution **T_EX**, des images **météo**, ainsi qu'une archive concernant l'intelligence artificielle.

ftp.cdrom.com Ce site est celui d'une entreprise fabriquant des **CD-ROMs** et qui les laisse à usage public puisqu'il ne s'agit que de compilations. Ce site est aux États-Unis. Il contient par exemple **Aminet**.

prep.ai.mit.edu Ce site, situé au MIT à Boston, est le site originel des logiciels **GNU**. Ce site est bien évidemment surchargé et de nombreux miroirs (anglais: *mirror*) existent.

rtfm.mit.edu Ce site, situé également au MIT à Boston, contient dans le répertoire pub/usenet la plupart des **FAQs** des **forums de news de Usenet**.

ftp.switch.ch Le site du réseau académique suisse **SWITCH**, qui contient de nombreux miroirs, comme **GNU**, **Linux** et pas mal de logiciels **Macintosh** et **MS-DOS**.

ftp.eunet.ch Un site relativement complet, avec par exemple **Amiga NetBSD** et **Aminet**.

ftp.germany.eu.net Un site complet, principalement pour logiciels UNIX, malheureusement parfois un peu en retard.

nic.funet.fi Un site très bien géré, mais un peu lent, situé en Finlande.

info.rz.uni-ulm.de Un bon site VMS situé en Allemagne.

sunsite.unc.edu Ce site des USA contient la plupart des distributions **Linux** ainsi que beaucoup de logiciels pour d'autres plateformes.

jplinfo.jpl.nasa.gov Ce site contient des documents graphiques et textuels sur les missions spatiales et les sondes, situé aux États-Unis.

ftp.ibp.fr Ce site contient quelques distributions **Linux**.

25 Communication à distance

25.1 Introduction

Cette section explique en détail le fonctionnement des commandes utilisables pour communiquer entre utilisateurs ou obtenir de l'information sur des utilisateurs ou systèmes locaux/distants de façon interactive.

25.2 finger

Cette commande permet de connaître les utilisateurs connectés actuellement sur le système. Elle permet également d'obtenir des renseignements plus précis sur un autre utilisateur.

`finger` liste les utilisateurs connectés

`finger user` donne des informations plus précises sur l'utilisateur `user`, entre autres affiche `~user/.plan` s'il est lisible.

Exemple: Mais qui est donc `falken` ?

```
<vulcan> schaefer:/users/schaefer> finger falken@kremvax.alphanet.ch
[kremvax.alphanet.ch]
Login name: falken In real life: Professor Falken
Office: NORAD, 555
Directory: /users/falken Shell: /users/falken/globalwar
Never logged in.
No Plan.
```

Dans cet exemple, on utilise la forme étendue `user@host.domain` qui permet de faire un **finger** à travers **Internet**.

25.3 who

Permet de connaître les utilisateurs en ligne.

25.4 w

Donne des informations comparables à celles données par `who`.

25.5 write

Permet d'écrire directement sur l'écran d'un autre utilisateur, pour autant qu'il n'ait pas interdit cela avec la commande `mesg -n`. La syntaxe est `write user terminal`, avec `terminal` correspondant au nom du terminal où se trouve l'utilisateur s'il est connecté sur plusieurs terminaux (dans le sens large). Si cette dernière information n'est pas précisée, le premier trouvé sera sélectionné. On termine avec `CTRL-D`.

25.6 talk

La syntaxe est similaire à celle de `write` sauf que la conversation est bidirectionnelle. On ne donne pas cependant le nom du terminal. L'autre utilisateur voit sur son écran un message l'invitant à répondre de façon identique à l'initiation d'appel. Pour un `talk` sur une autre machine, le nom devient similaire à une adresse **mail** dans la plupart des cas. Pour une discussion textuelle multi-utilisateur, on peut utiliser le système **IRC** (voir section 29 page 97).

26 Connexion à distance

26.1 Introduction

Les commandes à distance permettent d'ouvrir des sessions virtuelles sur des systèmes distants (réseau local ou réseau Internet). Ces commandes peuvent agir sur le système local ou sur le réseau local. Pour qu'elles puissent agir sur d'autres ordinateurs, il faut soit une connexion extérieure permanente, ou que la connexion extérieure soit établie pendant la session. Il est important de noter que certaines de ces commandes ne sont pas disponibles par défaut. Il faut les faire précéder de `/usr/etc/` (ou d'un autre préfixe dépendant⁷⁷ du système).

26.2 telnet

Cette commande permet d'ouvrir une session (connexion interactive) sur un site distant. Cela nécessitera dans la plupart des cas d'avoir un accès sur la machine distante. On peut aussi parfois indiquer un numéro de port identifiant le service demandé. La connexion n'est pas 8 bits, il est impossible de faire des transferts de fichiers sur une session **telnet** sans encapsulation. De plus, certains systèmes UNIX interprètent lors du login UNIX via **telnet** les caractères `#` et `@` de façon spéciale. Il est donc recommandé de ne pas les utiliser.

Le protocole **TELNET** utilise **TCP** comme transport, sur le port numéro 23, défini comme service standard (anglais: *well-known access point*) dans le **RFC-1700** et dans le fichier UNIX `/etc/services`. On peut donner en paramètre à la commande `telnet`, en plus du nom ou de l'adresse IP du système à contacter le numéro de port ou le nom du service. On peut ainsi, par exemple, tester le fonctionnement de services comme **FTP**, port 21, ou **NNTP**, port 119, la plupart des services **Internet** utilisant des protocoles textuels. Les données circulent en clair, sans authentification ni chiffrement.

⁷⁷La commande `whereis` permet parfois de trouver cela. Certains manuels systèmes indiquent le chemin complet des commandes.

26.3 rlogin

Cette commande permet, comme **telnet**, de se connecter à un site distant. Le protocole rlogin est, cependant, à la fois plus sûr et plus rapide que le protocole telnet. Il permet aussi, dans un réseau sûr (équivalent, où les mêmes utilisateurs existent sur chacune des machines) de ne pas donner de mot de passe si on est déjà connecté.

26.4 rsh

On peut exécuter des commandes **shell** à distance avec cette commande, pour autant que le fichier `.rhosts` ait été configuré (ou que les deux machines concernées soient équivalentes du point de vue de leurs utilisateurs).

26.5 rcp

On peut copier des fichiers à distance avec cette commande, avec les mêmes restrictions vues pour la commande rsh.

26.6 ssh

Plus qu'une commande, **SSH** est un protocole qui crée une connexion chiffrée entre un client et un serveur (`sshd`), sur laquelle peuvent être émulées les fonctionnalités des commandes `r-` : `ssh` remplace `rsh` et `rlogin`, `scp` remplace `rcp`. De plus, on peut créer des tunnels (redirections) entre des ports distants et locaux (et symétriquement), pour p.ex. rediriger le trafic **POP** via le tunnel sécurisé. Enfin, on se connecte soit à l'aide d'un mot de passe, soit à l'aide de méthodes similaires à celles des anciennes commandes `r-`, soit enfin – et c'est le plus intéressant – via une clé privée **RSA** par exemple (sans mot de passe) en stockant la clé publique sur le serveur sur lequel on veut se connecter. Un **agent**, `ssh-agent`, permet de ne taper qu'une fois le mot de passe de décriptation de la clé privée. On peut aussi limiter l'éventail des commandes possibles et créer une simple solution de sauvegarde sécurisée et automatisée, par exécution de commande distante. Ou connecter une session **SLIP** ou **PPP** et créer un réseau privé virtuel simple (anglais: *Virtual Private Network*, **VPN**). En bref, la suite d'outils **SSH** remplace avantageusement les commandes `r-` et `telnet`, tant au niveau des fonctionnalités que de la sécurité. Plusieurs implémentations interoperables existent, citons notamment l'excellent logiciel libre **OpenSSH** (cf <http://www.openssh.org>).

27 Diagnostic de problèmes de réseau

27.1 Introduction

Les commandes de diagnostic réseau permettent de déterminer la connectivité d'une machine sur Internet, la route que prennent les paquets pour atteindre celle-ci, les connexions réseaux actives sur la machine locale, les services en attente, le routage ainsi que des informations administratives de gestion de réseaux sur Internet.

27.2 ping

La commande ping permet de savoir si un site distant est atteignable et en combien de temps le cas échéant. Elle est principalement utile pour détecter des anomalies de configuration réseau (*voir section 30.5.1 page 99*).

27.3 traceroute

Cette commande permet de tracer le parcours de paquets IP jusqu'à un site distant. Son fonctionnement est similaire à celle de la commande ping. Cette commande provoque cependant une grande charge pour un réseau. C'est pour cela qu'elle est souvent d'accès restreint à root seulement (privilegiée).

Exemple: Voici un exemple d'utilisation de traceroute

```
<vulcan> schaefer:/users/schaefer> traceroute www.whitehouse.gov
```

```
traceroute to www.whitehouse.gov (128.102.252.1), 30 hops max, 40 byte packets
 1 mips.alphanet.ch (193.72.186.3) 1.556 ms 1.314 ms 1.191 ms
 2 Basel3.CH.EU.NET (146.228.11.3) 247.382 ms 237.834 ms 228.55 ms
 3 Basel5.CH.EU.NET (146.228.11.5) 243.731 ms 257.36 ms 238.476 ms
 4 Zuerich14-S1.CH.EU.NET (146.228.101.2) 253.888 ms 257.56 ms 288.421 ms
 5 Zuerich10.CH.EU.NET (146.228.10.88) 263.706 ms 458.718 ms 278.866 ms
 6 Amsterdam4.NL.EU.net (134.222.9.1) 428.529 ms 358.247 ms 280.03 ms
 7 Amsterdam6.NL.EU.net (134.222.85.6) 279.198 ms 378.299 ms 308.413 ms
 8 Vienna1.VA.US.EU.net (134.222.228.14) 578.679 ms 348.422 ms 478.684 ms
 9 Vienna1.VA.ALTER.NET (137.39.32.65) 537.86 ms 399.129 ms 398.594 ms
10 Vienna6.VA.ALTER.NET (137.39.11.11) 438.622 ms 388.4 ms 378.998 ms
11 San-Jose3.CA.ALTER.NET (137.39.13.5) 547.441 ms 498.442 ms 499.816 ms
12 San-Jose3.CA.ALTER.NET (137.39.13.5) 439.174 ms 478.236 ms 539.931 ms
```

```

13 ames3.mae-west.ep.net (198.32.136.43) 458.7 ms 489.076 ms 519.037 ms
14 n254-ed-cisco7010.arc.nasa.gov (128.102.64.254) 557.967 ms 518.398 ms 599.35 ms
15 razorback.arc.nasa.gov (128.102.252.1) 438.435 ms 448.794 ms 489.34 ms

```

27.4 netstat

Cette commande permet de connaître les sessions **TCP** ouvertes sur la machine courante, de visualiser les informations de routage ou les interfaces **TCP/IP** disponibles.

```

netstat affiche les informations quant aux sessions TCP ouvertes,
netstat -a affiche les informations quant aux serveurs en attente,
netstat -r affiche les informations de routage des paquets,
netstat -i affiche les informations sur les interfaces disponibles.

```

27.5 whois

Cette commande permet d'obtenir des informations sur des **domaines** Internet enregistrés et sur leur administrateur. Exemple: Voici un exemple d'utilisation de WHOIS sur le serveur ns.ripe.net, qui s'occupe des réseaux européens

```

<mips> schaefer:/d2/users/schaefer> whois -h ns.ripe.net ALPHANET
inetnum: 193.72.186.0
netname: ALPHANET
descr: ALPHANET NF
descr: Association for Research & Development without lucrative goals
descr: Colombier
country: CH
admin-c: Marc Schaefer
tech-c: Marc Schaefer
changed: fm@eunet.ch 940311
source: RIPE

person: Marc Schaefer
address: ALPHANET
address: Tempé 8
address: CH-2520 La Neuveville
phone: +41 38 41 29 29
e-mail: schaefer@alphanet.ch
nic-hdl: MS193
changed: fm@eunet.ch 940311
source: RIPE

```

On peut aussi accéder à ce service par la page WWW du NIC France à l'url <http://www.nic.fr/info/whois.html>.

28 WWW (World-Wide Web)

28.1 Introduction

WWW est un système hypertexte distribué et interactif à base de documents multimédia. Il s'utilise pour parcourir de la documentation (sous forme textuelle, graphique ou sonore) qui peut se trouver sur différents sites. De plus, les documents eux-mêmes peuvent contenir des références (que l'on appelle **liens**) à d'autres documents sur le même site ou ailleurs, ou même réagir (on parle ici de vraie interactivité lorsque l'utilisateur d'un tel système peut non seulement choisir ce qu'il veut voir, comme avec **teletexte**, mais en plus peut interroger des serveurs d'information et participer, par exemple, à des discussions ou des conférences, ou simplement envoyer un message public ou privé).

Les possibilités de ce système sont très étendues : on peut voyager à l'intérieur comme dans une ville ou une bibliothèque⁷⁸, une lecture en amenant une autre, ou une rencontre virtuelle. C'est pour cela que certains appellent cet espace informatique d'informations et d'actions le *village virtuel*.

Il faut faire attention lorsque l'on parle de WWW à ne pas confondre le système et les **clients** qui permettent d'y accéder. Par exemple, lynx n'offre que l'interface texte (même s'il peut lancer des programmes appropriés pour différents types de données, comme des images, des sons, etc.), au contraire de Mosaic ou de Netscape, qui intègrent le tout dans les documents.

WWW est souvent vu comme l'intégrateur des services d'Internet, car il permet d'accéder à la fois à **FTP**, **Gopher**, **news**, ainsi qu'aux possibilités multimédia proprement dites.

⁷⁸Certains parleront de kiosque ou de souk.

28.2 Terminologie

La terminologie WWW est relativement fournie, mais n'intéresse probablement que les concepteurs de documents HTML, de logiciels clients ou de serveurs. Ci-après se trouvent quelques explications concernant les mots les plus courants :

HTTP HyperText Transport Protocol. Il s'agit du protocole de transport des données hypertextes basé sur **TCP**.

HTML HyperText Markup Language. Il s'agit du langage textuel permettant de rédiger des **homepages** et des documents hypertextes qui seront ensuite référencés par un URL. Pour les connaisseurs, ce langage se rapproche de **L^AT_EX**⁷⁹ et est une application du standard **SGML**.

clients Un client est un logiciel permettant d'accéder à WWW. Plusieurs types de clients existent, et parmi eux, il y a parfois plusieurs implémentations. Les deux clients les plus connus sont **lynx**, le client texte, et **Mosaic**, le client graphique. Des implémentations de **Mosaic** existent sur plusieurs plates-formes, libres ou commerciales. Les clients ne fonctionnent en général pas seuls, ils nécessitent en fait la plupart du temps une couche de transport **TCP/IP**.

serveurs Il s'agit des logiciels qui autorisent l'accès aux clients. Plusieurs versions existent, même pour des ordinateurs personnels.

forms Il s'agit d'une sorte de formulaire qui se remplit d'une manière similaire au système **videotex** et qui permet, par exemple, des recherches dans des bases de données. Pratiquement, ce système permet l'exécution de n'importe quelle suite d'instructions de traitement sur les données enregistrées dans le formulaire sur le serveur distant.

homepage Il s'agit d'une page d'entrée d'un site ou encore d'un utilisateur particulier. Par exemple, ma page d'entrée est, dans la spécification d'un document **HTTP**, ou **URL**, `http://www.alphanet.ch/~schaefer`.

28.3 Protocole

Le protocole HTTP utilise des connexions **TCP**. Au niveau utilisateur, on donne une spécification de document (aussi appelée **URL** pour Universal⁸⁰ Resource Locator) qui permet de retrouver le document concerné. Par exemple, on a les URLs `http://www.alphanet.ch` ou encore `ftp://ftp.alphanet.ch`. Les échanges de données sont faits entre un client et un serveur distant, en général sur le port TCP numéro 80. La syntaxe permet cependant de spécifier ce port, comme dans, par exemple `http://www.alphanet.ch:80/members.html`. Un URL est en général sensible à la casse.

Forme plus générale :

```
protocole://utilisateur:mot-de-passe@domaine:port/sentier/
```

28.4 Sécurité

La version actuelle du protocole prévoit des extensions futures pour un protocole réputé sûr de communication entre serveur et client via des clés publiques (comme avec **PGP**), permettant l'installation de véritables services commerciaux, où le numéro de carte de crédit pourra être transmis sans risque d'interception par une écoute clandestine entre les deux points. Pour l'instant, il faut être conscient du fait que les communications se font en clair et donc sont interceptables par un espion bien situé.

28.5 Interfaces

Comme dit précédemment, diverses interfaces (ou clients) existent pour accéder à WWW. On ne citera que les plus répandues disponibles localement. En général, on lance le programme avec, sur la ligne de commande, la spécification de l'**URL** du document. Les logiciels permettent souvent de se déplacer vers un document quelconque avec une commande du type *go*, *goto* ou encore *Open Location*, et s'ouvrent souvent avec une page par défaut utile si l'utilisateur ne spécifie pas de document sur la ligne de commande.

28.5.1 lynx

Cette interface texte permet cependant, après configuration appropriée, d'accéder également aux images et aux sons (grâce à des clients externes, et non pas en mélangeant graphisme dans le texte comme avec de véritables clients graphiques). Elle permet aussi de transférer (par exemple via **ZModem**) des pages ou des fichiers de types divers. Elle a l'avantage d'être, évidemment, très rapide et ménage la **bande passante**, puisqu'en général on ne transfère pas d'image. Un exemple est donné à la figure 17 page suivante. Notez que **lynx** ne fonctionne pas tout seul sous **X11**⁸¹ : ce logiciel nécessite un environnement terminal `xterm` (sans **X11**, un login normal suffit).

⁷⁹Il existe d'ailleurs des traducteurs automatiques.

⁸⁰Uniform était l'ancienne définition.

⁸¹D'ailleurs le seul intérêt de lancer **lynx** sous **X11** est de limiter les données transférées, ce qu'on peut aussi demander sous **Mosaic** avec l'option *Delay image loading*.

```

Xterm                                     Page d'entrée ALPHANET (p1 of 2)

[IMAGE]

                                INTRODUCTION

Bienvenue sur l'interface WWW d'ALPHANET NF qui contient des liens
hypertextes sur divers documents relatifs au club et aux possibilités
du réseau ALPHANET.

                                DIVERS LIENS

* Informations ALPHANET NF
* La page des membres
* Information générale sur les services ALPHANET
* Accès aux services ALPHANET
* Documentation plus spécifique
* Informations régionales
* Divertissement
* Bugs connus

-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

```

FIG. 17 – Fenêtre xterm contenant lynx sous X11

28.5.2 w3m

Ce client texte supporte notamment les tables, et **SSL** en mode standard : il est donc meilleur que lynx, même s'il peut sembler plus compliqué au premier abord.

28.5.3 Mosaic

Cette interface est graphique et est disponible sur la plupart des plates-formes. Une copie d'écran se trouve à la figure 18 page suivante. Il s'agit d'un logiciel librement redistribuable. Il est cependant obsolète de nos jours.



FIG. 18 – Fenêtre typique de Mosaic sous X11

28.5.4 Chimera

L'avantage de ce client est qu'il offre des capacités similaires à Mosaic, sans demander un support des bibliothèques **Motif**. C'est donc un candidat idéal pour les systèmes UNIX librement redistribuables.

28.5.5 Netscape

Ce client est très similaire à Mosaic. Il offre en plus un système de cache et de multitâche interne permettant d'améliorer l'interactivité. Cependant, ce logiciel est assez récent, et il nécessite l'acceptation d'une licence d'utilisation. En effet, ce produit est commercial. Son aspect est également légèrement plus attractif que Mosaic (voir figure 19 page suivante). Des versions pour la plupart des plates-formes sont disponibles. Une version open source de ce logiciel existe également, elle se nomme **Mozilla**.



FIG. 19 – Fenêtre typique de Netscape sous X11

28.5.6 Konqueror

Dans sa version 2.1, ce navigateur, qui fait partie de l'environnement graphique **KDE**, offre de nombreuses possibilités avancées. De plus, il s'agit de logiciel libre.

28.6 Comment créer votre propre page WWW

Pour ce faire, il faut un serveur WWW (souvent appelé **httpd**). Demandez à votre administrateur si un tel serveur est fonctionnel. Vous pouvez aussi l'installer vous-même, mais pas sur le port standard (80). Il existe des serveurs pour la plupart des plateformes. Ensuite, il faut connaître comment votre document va être référencé. Une méthode relativement standard est que si votre nom d'utilisateur est **user** et que la machine sur laquelle le **daemon** httpd tourne se nomme **www.alphanet.ch**, votre **homepage** est <http://www.alphanet.ch/~user/>⁸² et correspond au fichier **home.html** ou **index.html** se trouvant dans le répertoire **public_html** situé dans votre **homedir**. Le format de ce fichier est expliqué dans les documents à propos de **HTML**, disponibles par exemple sur <http://w3.org>.

Vous pouvez aussi prendre un document⁸³ existant comme modèle sur le réseau pour autant que cela soit autorisé et la sauver localement à l'aide de l'option **load to local disk** de la plupart des clients. Des logiciels d'aide à la rédaction de documents **HTML** existent : citons par exemple le mode **Emacs** **html-helper-mode**.

Il est extrêmement important de tester votre page avec plusieurs clients différents. Netscape est connu pour être beaucoup trop tolérant avec les erreurs de syntaxe, ce qui est très bien pour l'utilisateur final, mais un manque du point de vue du concepteur de documents. Il existe des logiciels, comme **htmlcheck** qui permettent de vérifier un document **HTML**. N'oubliez pas que certains utilisateurs n'ont pas de **browser** graphique ou préfèrent ne pas utiliser les images pour des questions de vitesse : les **alternates** sont utiles pour les images, ainsi que des simples listes de choix en remplacement d'une **image-map**. Comme règle générale, pensez plutôt au contenu qu'à la mise en page, car plus le client a de la marge dans son interprétation, mieux votre document supportera les changements de technologie (changer la couleur de fond ou faire clignoter ne devraient pas être nécessaires).

⁸²Le / final est normalement optionnel, mais certains serveurs **WWW** ont un comportement étrange si celui-ci n'est pas ajouté.

⁸³Dites plutôt "document" que "page", la deuxième version fait penser à **Videotex**.

28.7 Exemple d'un fichier source HTML

Le langage de description d'une page WWW se nomme **HTML**. De la documentation est disponible sur le serveur officiel du **W3 Consortium** : <http://www.w3.org>

En résumé, une source **HTML** est composée de lignes de texte comportant des séquences d'échappement (marques, anglais: *tags*) commençant par `<` suivi du nom de la séquence, par exemple `TITLE` ou `b`, d'éventuels paramètres et enfin de `>`. Comme dans **LaTeX**, on se trouve souvent dans un environnement spécifique initialisé par la séquence d'échappement. On quitte l'environnement par `</`, suivi du nom de la séquence et de `>`. Par exemple, la séquence `<H1>` permet de spécifier un titre de premier niveau, que l'on termine avec `</H1>`. Notez qu'un document **HTML** peut comporter des accents selon la convention du langage ou les accents standards. La figure 20 montre un exemple simpliste d'une telle page.

```
<HTML>
<HEAD>
<TITLE>Ma page d'entrée</TITLE>
<!-- Ceci est un commentaire. On définit dans la partie HEAD (qui finit -->
<!-- par /HEAD des informations sur le document) -->
</HEAD>
<BODY>
<!-- On définit un premier niveau (section) -->
<!-- On appelle une image en ligne avec un texte par défaut -->
<H1><IMG ALT="Moi ler" SRC="http://www.alphanet.ch/~schaefer/marc.png">
Ma homepage
</H1>
Bienvenue sur ma page d'entrée. Ceci est un petit exemple du format d'une
page HTML. Elle contient des <a href="http://info.cern.ch/">références</a>
à des informations sur WWW:

<!-- Ce qui suit est une énumération de liens -->
<ul>
<li><a href="http://info.cern.ch/hypertext/WWW/TheProject">
Les pages officielles du projet WWW (anglais)</a>
<li><a href="http://info.cern.ch/hypertext/WWW/Clients">
Liste de logiciels clients, avec documentation (anglais)</a>
<li><a href="http://info.cern.ch/hypertext/WWW/Daemon/Overview.html">
Liste de logiciels serveurs, avec documentation (anglais)</a>
<li><a href="http://info.cern.ch/hypertext/WWW/Technical.html">
Détails techniques, spécifications, etc. (anglais)</a>
<li><a href="http://validator.w3.org/">S'assurer de la validité HTML.</a>
</ul>
</BODY>
</HTML>
```

FIG. 20 – Exemple d'une page HTML

On distinguera entre styles logiques (section, sous-section, etc) et styles physiques (gras, italiques, tailles de générateur de caractères, etc). Le moteur <http://validator.w3.org/> permet de valider le code HTML, y compris de versions différentes.

28.8 Quelques serveurs WWW

Voici une liste de quelques serveurs **WWW**. Elle est donnée sans garantie, en espérant que ces sites seront toujours valides lorsque vous lirez ce texte.

– Les index

<http://www.google.com> le moteur de recherche que je préfère.

<http://www.yahoo.fr> archive les adresses HTML par thème, il y a aussi un moteur de recherche.

<http://heiwwww.unige.ch/switzerland> Les serveurs WWW suisses.

<http://www.ecila.com/> Index d'URL francophones, France.

<http://www.lycos.com> Un moteur de recherche.

<http://www.altavista.digital.com> Le moteur de recherche de DEC.

<http://cuiwww.unige.ch/meta-index.html> Systèmes de recherches universels, Université de Genève, Suisse.

<http://www.dejanews.com> archive les messages envoyés dans les news (vos messages aussi donc !).

<http://www.urec.fr/archie.html> archive l'ensemble des sites FTP anonymes.

<http://www.four11.com/> l'annuaire des adresses e-mail.

– Les organisations

<http://w3.org> Le serveur **WWW** officiel du consortium **W3**. Vous y trouverez les standards, de la documentation et des logiciels (serveurs et clients), ayant un rapport avec **WWW**. Ce site est désormais aux USA après avoir été au **CERN**.

<http://www.undep.or.at/unlinks.html> Les Nations Unies.

– Quelques serveurs universitaires et d'organismes de recherche

<http://www.epfl.ch> Serveur de l'École Polytechnique Fédérale de Lausanne, Suisse.

<http://www.alphanet.ch> Le serveur public d'**ALPHANET NF**, Suisse.

<http://www.cnrs.fr> Le Centre National de Recherche Scientifique, France.

<http://www.pasteur.fr> L'institut PASTEUR, Paris, France.

– Diverses choses

<http://www.meteo.fr/> La météo sur l'Europe, et <http://weather.yahoo.com/> pour avoir les prévisions dans les grandes villes du monde.

<http://wings.buffalo.edu/world> The virtual Tourist, USA.

<http://info.cern.ch/Space/Overview.html> Collection d'informations sur l'espace, CERN, Suisse.

<http://textius.ina.fr/CP/MondeDiplo/> Le monde diplomatique.

<http://www.hebdo.ch> L'HEBDO, magazine suisse romand.

28.9 Performances

WWW, par son protocole de transport **HTTP** mais aussi en raison de la façon dont certains documents sont réalisés, souffre de divers problèmes :

1. L'affichage d'un document **HTML** nécessite souvent l'accès à beaucoup de composants (images, textes) qui doivent être transférés indépendamment : en effet, **HTTP** ne permet pas le transfert de plus d'un "objet" par session **TCP**. Certains clients résolvent ce problème en ouvrant simultanément plusieurs sessions **TCP** (ce qui charge le réseau encore plus). Cela n'est qu'une façon détournée de passer outre une déficience du protocole **HTTP** tel que défini et implanté.
2. L'ajout de nombreuses images de grande taille et de diverses fioritures (utilisation de symboles d'énumération non standards qui doivent être chargés) ralentit encore plus l'accès aux documents et charge le réseau.

Une solution est d'installer un serveur **cache** qui relayera les demandes et sauvegardera en local les documents accédés récemment, soulageant le réseau et augmentant l'interactivité.

Une solution plus difficile à mettre en œuvre puisque touchant aux serveurs distants et aux clients locaux serait d'implanter un système d'indication de longueur d'objet à la **MIME** : en effet, dans la plupart des cas, le serveur peut connaître à l'avance la taille des objets à transmettre. On pourrait ainsi éviter l'ouverture coûteuse de nombreuses sessions **TCP** en utilisant un **cache de connexions**, utile dans le cas où de nombreux objets d'une page se situeraient sur un autre site. Les serveurs et les clients seraient autorisés à couper des sessions **TCP** après chaque objet, mais ne le feraient qu'en cas de nécessité : une page de bienvenue pourrait ainsi être transférée en une seule opération⁸⁴.

29 IRC : Internet Relay Chat

29.1 Introduction

IRC est à la fois un réseau, **EFnet**, un protocole basé sur **TCP** (port 6667) et un système de conférence interactive textuelle distribuée (ou relayée) sur de nombreux serveurs à travers le monde.

IRC est basé sur des canaux de discussion nommés (anglais: *channel*), comme par exemple #francais, #amiga, #linux, ou encore #scouting sur le modèle de la CB ou des systèmes de discussion multi-utilisateurs des **BBS**.

Un canal est automatiquement créé lorsqu'un utilisateur le rejoint. Cet utilisateur en devient automatiquement l'opérateur (anglais: *channel op*) et dispose alors de commandes de gestion du canal considéré (pour changer le titre, changer les attributs du canal ou éjecter des utilisateurs indésirables). On peut transmettre cette qualité à un autre utilisateur. Les administrateurs **IRC** (anglais: *irc ops*) peuvent, eux, agir directement sur de nombreux paramètres et exécuter des commandes restreintes (comme par exemple terminer la connexion d'un utilisateur indélicat).

Certains clients permettent d'utiliser les fonctions de discussion privée et de transfert de fichiers par **IRC**, même si ce dernier genre d'opérations est de plus en plus mal vu.

Il existe plusieurs réseaux de serveurs **IRC**. **EFnet**, le plus ancien, est également le plus surchargé et le plus lent. D'autres réseaux se sont créés pour essayer de retrouver l'ambiance initiale, comme **Undernet**.

⁸⁴Cela est implémenté dans HTTP 1.1

IRC a surtout gagné en popularité pendant la Guerre du Golfe, où les actualités étaient commentées en direct sur un des canaux. Certains utilisateurs d'**IRC** ayant utilisé ce moyen de communication fantastique pour des opérations illégales (transfert de logiciels piratés principalement), de nombreux serveurs ont été fermés. De plus, le trafic est gigantesque et peu de serveurs arrivent à suivre la cadence. On observe alors de nombreux délais (anglais: *netlag*).

La table 11 liste, par pays, quelques-uns des serveurs publics **EFnet**. La plupart des serveurs ne sont pas publics : ils sont réservés à un pays ou à une organisation, même si la plupart des administrateurs sont souvent tolérants tant qu'il n'y a pas d'abus. Créer son serveur **IRC** soi-même est voué à l'échec sans un nombre d'utilisateurs suffisant (au moins sur **EFnet**). Pour trouver un serveur, on peut se connecter à un autre serveur et donner la commande `/LINKS`. On peut aussi lire le forum de **news alt.irc** ou consulter l'**URL** <http://www.world.net/~pirovich/efnet.htm>⁸⁵.

29.2 Logiciels clients

29.2.1 irc

Cette commande implémente une interface très simplifiée d'accès à **IRC**. Les commandes sont textuelles et l'affichage du type **VT100**. Dans certains cas, particulièrement si votre émulation VT100 est déficiente, il est conseillé d'entrer `irc -d` pour supprimer les codes de positionnement.

Pays	Adresse
Suisse	irc.span.ch
	irc.datacomm.ch
France	ircnet.kaptech.fr
	irc.enst.fr
	ns.ensicaen.ismra.fr

TAB. 11 – Serveurs **IRC EFnet** par pays

30 Information sur le protocole SLIP

30.1 Introduction

Le protocole **SLIP** permet simplement de faire transiter des paquets **TCP/IP** sur une simple ligne série. Il permet non seulement de relier un ordinateur à un réseau, mais également de lier deux réseaux entre eux, chacun des ordinateurs des deux réseaux pouvant ensuite accéder aux ordinateurs de l'autre réseau ainsi qu'aux ordinateurs accessibles depuis les ordinateurs de l'autre réseau (et éventuellement d'autres réseaux connectés).

Vu qu'il implémente parfaitement et complètement le protocole **TCP/IP**, des sessions multiples peuvent être établies sur la même ligne série. Il faut noter que **SLIP** n'est pas optimisé lors, par exemple, du raccordement d'un seul ordinateur à un seul autre ordinateur, ou d'un seul ordinateur à un réseau. Une extension du protocole **SLIP**, nommée **CSLIP** permet, au moyen de la définition dynamique d'un cache des connexions **TCP**, de diminuer très sensiblement les données échangées de façon interactive (par exemple sous **telnet**, chaque caractère tapé individuellement provoquera l'émission de trois caractères sous **CSLIP** contre une quarantaine sous **SLIP**). Il existe également des protocoles plus généraux que **SLIP**, comme **PPP**, qui englobe également des optimisations (compression de données, utilisation optimale des deux canaux, négociation d'adresses, etc.).

Si votre fournisseur de service n'offre pas **SLIP**, il offre certainement **PPP**. Notez que le logiciel **SLIRP**, disponible sur plusieurs environnements **UNIX** permet de simuler efficacement une session **SLIP** et de vous connecter via **TCP/IP** à travers un simple compte **UNIX shell**. Du côté de votre ordinateur, il n'y a aucun logiciel supplémentaire nécessaire.

30.2 Information sur le protocole

Sur la liaison série s'écoulent des paquets et des confirmations de réception (en effet **SLIP** corrige⁸⁶ les erreurs éventuelles de transmission). Ces paquets contiennent l'information suffisante pour le routage, soit l'**adresse IP** du destinataire et celle de l'expéditeur, les numéros de ports associés, le type du paquet, le **TTL**, etc. Il faut noter que comme deux réseaux peuvent être connectés (et pas simplement deux ordinateurs), les adresses destination et expéditeur ne sont pas connues par simple identification de l'interface.

En pratique, il faut considérer deux valeurs qui indiquent la qualité d'un lien **SLIP** :

temps de réponse aussi appelé **ping delay**, c'est le temps de l'aller-retour d'un paquet entre les deux points de la liaison série, y compris le délai de gestion des paquets, le contrôle d'erreurs, etc. Cette valeur détermine l'interactivité d'une liaison. Plus cette valeur est grande, plus **telnet** et **rlogin** deviennent inutilisables en raison des délais d'écho. Ce paramètre peut être influencé très négativement par la correction d'erreur au niveau du modem.

⁸⁵Oui, même si cela peut paraître bizarre, l'**URL** se termine bien par `.htm` au lieu de `.html`. Certains environnements sont intolérants.

⁸⁶En fait, ce n'est pas tout à fait cela : **SLIP** ne corrige (ni ne détecte) rien du tout, mais **TCP/IP** s'en charge. La correction d'erreur est donc de bout en bout au lieu de point à point : **PPP** est meilleur.

vitesse de transfert c'est ce paramètre qui est intéressant lors de transferts de fichiers de masse via **ftp**, de mail ou de news. En général, la compression de données améliore ce paramètre.

Pour une liaison SLIP à 19.2kbps (vitesse réelle sans compression ni correction d'erreur, ni modem), on obtient des valeurs typiques comme 150 ms pour le temps de réponse et autour de 18kbps pour la vitesse de transfert.

30.3 Configuration d'un réseau SLIP

Cette section dépend évidemment fortement de votre réseau local : votre administration système pourra certainement vous renseigner sur les points de détail d'application. En outre, l'allocation définie est statique, ce qui ne convient pas à tous les réseaux. La configuration dépend bien évidemment du type de connexion entre les deux systèmes, des services demandés et surtout du système d'exploitation et des éventuels logiciels supplémentaires de la machine cible. Pour une liaison entre un ordinateur personnel et **mips**, la configuration, au niveau de l'ordinateur personnel est relativement simple. Elle consiste à indiquer au logiciel TCP/IP que tous les paquets ne correspondant pas à la machine locale doivent passer via la ligne série sous SLIP en direction du relais. Auparavant, il faut bien évidemment déclarer la liaison série au système local. Ces deux opérations se font en général par des commandes ressemblant aux lignes suivantes :

```
ifconfig networks/slip.device/0 localhost 193.72.186.3
route add net default 193.72.186.3 1
```

Ensuite, il faut indiquer à votre logiciel TCP/IP que mips.alphanet.ch, [193.72.186.3], est un **nameserver**, par exemple à l'aide du fichier `/etc/resolv.conf`. Du côté UNIX, la session SLIP s'effectue à l'aide d'un **login** spécial.

30.4 Problèmes courants

30.4.1 CSLIP

Cette variante du protocole SLIP, nommée **CSLIP** n'est malheureusement pas disponible sur le **mips**. Les deux protocoles sont incompatibles et cette incompatibilité se détecte au fait que les systèmes répondent aux **pings**, et les sessions **TCP** s'ouvrent, mais aucun paquet de données n'est reçu correctement.

30.4.2 PPP

Le protocole **PPP** n'est pas disponible.

30.4.3 Routage mal configuré

Sur la plupart des systèmes, une commande similaire à

```
route add net default mips 1
```

où mips doit éventuellement être changé en 193.72.186.3 permet d'assurer un routage correct des paquets.

30.5 Déroulement d'une session SLIP

En général, les produits **SLIP** disponibles ne permettent pas un appel automatique suivi du lancement du gestionnaire SLIP et de l'initialisation du routage. Une solution revient à demander au **modem** de ne pas tenir compte du signal **DTR** en provenance de l'ordinateur personnel, puis de composer le numéro du système **mips** à l'aide d'une émulation de terminal quelconque en vérifiant que la vitesse correspond à celle configurée dans SLIP, de se connecter avec le login spécial (qui commence par `sl` et qui vous sera donné par `sysadm@alphanet.ch`) et d'attendre le message de préparation du logiciel SLIP. Ensuite, on quitte le logiciel de communication et on lance le logiciel SLIP. La fin de communication peut s'obtenir en raccrochant le modem. Pour différentes raisons il faut éviter de laisser des sessions **TCP** ouvertes en raccrochant. Dans le pire des cas, cependant, le système répondra de nouveau après une minute.

Il existe de nombreuses variantes : certains systèmes offrent une négociation automatique des adresses ce qui peut être intéressant dans certains cas.

30.5.1 Diagnostic

Parfois, on veut être à même de diagnostiquer la cause d'un problème. Une des commandes qui permet de diagnostiquer certaines pannes est `ping`. Elle permet de vérifier qu'un ordinateur distant reçoit bien un datagramme et y répond. Elle permet aussi de connaître les éventuels goulets d'étranglement.

`ping` **répond avec une erreur** problème de routage probablement, on peut le vérifier avec `netstat -r` et `netstat -i` pour connaître respectivement le routage et les interfaces configurées.

`ping` **pas de réponse** vérifier les affichages d'un **transceiver Ethernet** ou d'un testeur de ligne série pour voir si le paquet est émis, si oui, le problème vient du routage inverse, d'un relais ou de la configuration distante : essayer sur la machine distante.

ping **marche, problèmes avec FTP** vérifier avec diverses tailles de paquet avec ping.

On peut aussi tester la plupart des services avec la commande `telnet` en spécifiant le nom de la machine puis le nom du service (ou son numéro lu dans `/etc/services`). La commande `netstat -a` permet de voir les services actifs sur la machine locale. Un délai de quelques secondes lors de l'ouverture d'une session **TELNET** ou **FTP** peut provenir d'une mauvaise configuration du serveur **DNS** local (le délai se produit avant le message de confirmation de connexion) ou être un effet de bord d'une tentative de la machine distante d'identifier la machine locale (requête **DNS** sur un domaine non configuré) ou plus probablement d'une tentative d'identification avec le service **IDENT**.

31 Information sur le système DNS (Domain Name Server)

31.1 Introduction

Le système **DNS** permet d'effectuer la correspondance entre les adresses textuelles, comme `ftp.alphanet.ch` et les adresses en format numérique (comme `193.72.186.3`), et vice-versa. On appelle la conversion nom vers adresse **resolving** et la conversion inverse **reverse mapping** ou **unresolving**.

Les serveurs DNS, aussi appelés **nameservers** sont normalement présents dans chaque **domaine** Internet et sont responsables de celui-ci. Lorsqu'une demande pour un domaine externe à l'organisation concernée est générée, ceux-ci transmettent plus loin la requête de façon hiérarchique, permettant de distribuer la base de données et les compétences en fonction de contraintes administratives. Pour des raisons évidentes, ils possèdent également un cache qui maintient en mémoire les adresses demandées récemment.

Plusieurs informations sont maintenues par un serveur DNS, allant du type de machine au serveur de mail, en passant par des informations sur l'organisation responsable du réseau, etc.

Les utilitaires réseaux, de même que les applications, sont compilés de manière à accéder aux services du serveur de noms de la machine principale, et éventuellement d'une machine secondaire en cas de panne. L'ensemble de ces services se nomme le **resolver**. Au niveau de la machine cliente de ce service, on configure souvent `/etc/resolv.conf`.

31.2 Commande associée d'interrogation : nslookup

La commande **nslookup** permet d'interroger n'importe lequel des serveurs de noms (**nameservers**) prévus ou d'en spécifier un nouveau manuellement.

Exemple: Voici un exemple d'utilisation

```
<mips> falken:/d2/users/falken> /usr/etc/nslookup
Default Server: mips.alphanet.ch
Address: 193.72.186.3

> server whitehouse.gov
Default Server: whitehouse.gov
Address: 198.137.240.100

> ls whitehouse.gov
[whitehouse.gov]
Host or domain name Internet address
whitehouse server = gatekeeper.eop.gov
gatekeeper 198.137.241.3
whitehouse server = whitehouse.gov
whitehouse 198.137.240.100
whitehouse server = icm1.icp.net
icm1 192.94.207.66
whitehouse server = icm2.icp.net
icm2 192.80.214.224
whitehouse 198.137.240.100
localhost 127.0.0.1
loghost 127.0.0.1

> set querytype=MX
> set domain=gov
> whitehouse
Server: whitehouse.gov
Address: 198.137.240.100

whitehouse.gov preference = 100, mail exchanger = whitehouse.gov
whitehouse.gov inet address = 198.137.240.100
```

N'oubliez pas que les adresses entrées sont toujours (pour la résolution) interprétées dans le domaine courant : par exemple, si le domaine est `alphanet.ch`, entrer `whitehouse.gov` essaiera de résoudre `whitehouse.gov.alphanet.ch`. On peut changer le domaine ou simplement ajouter un point final : `whitehouse.gov..`

```
<vulcan> schaefer:/users/schaefer> nslookup
Default Server: mips.alphanet.ch
Address: 193.72.186.3

> set querytype=ANY
> alphanet.ch.
Server: mips.alphanet.ch
Address: 193.72.186.3

alphanet.ch
origin = mips.alphanet.ch
mail addr = schaefer.mips.alphanet.ch
serial = 113
refresh = 10800 (3 hours)
retry = 3600 (1 hour)
expire = 604800 (7 days)
minimum ttl = 86400 (1 day)
alphanet.ch nameserver = mips.alphanet.ch
alphanet.ch preference = 0, mail exchanger = mips.alphanet.ch
mips.alphanet.ch internet address = 193.72.186.3
```

32 Problèmes de télécommunication

32.1 Introduction

Cette section a pour but de définir quelques notions de base en télécommunication (principalement afin de permettre l'utilisation de modems et de logiciels de communication). On passera en revue les diverses normes et multiples standards ainsi que différentes méthodes de compression de données et de transfert de fichiers. On citera également quelques **FAQs** sur le sujet.

32.2 Les modems

Les **modems** (modulateur/démodulateur) sont des dispositifs permettant à un signal numérique (provenant usuellement d'un ordinateur) d'être codé de façon analogique sur une ligne de téléphone normale (modem en bande de base). Il faut cependant savoir que la plupart des réseaux de télécommunication, au moins au niveau interurbain, fonctionnent déjà sur des réseaux numériques, d'où l'offre relativement intéressante d'utiliser directement le support numérique (*voir section 32.4 on page 106*).

Les données venant d'un ordinateur sont en général codées en série sur un certain nombre de bits (typiquement 10, soit 8 bits de données et deux bits de contrôle nommés bits **START** et **STOP** en mode asynchrone).

Ces données sont traitées par le modem et envoyées sur la ligne. Dans certains cas, à cause de vitesses différentes entre la vitesse de ligne et la vitesse entre l'ordinateur et le modem, un protocole est mis en œuvre pour assurer l'intégrité des données: par exemple **RTS/CTS**, appelé également contrôle de flux matériel (**hardware handshaking**), ou **XON/XOFF**⁸⁷. Dans la plupart des cas, on dispose également d'un protocole pour la détection d'appel et la confirmation de connexion. Ces protocoles peuvent être réalisés de différentes manières, par des fils physiques ou par des séquences spéciales de caractères.

Les données circulant entre les deux modems étaient à l'origine d'un format très similaire aux données circulant entre l'ordinateur et le modem. Cependant, les modems actuels implémentent tous des protocoles logiques supplémentaires pour garantir l'intégrité des données en cas de perturbation de la ligne ainsi que pour diminuer le temps de connexion en compressant (comprimant) les données pendant la transmission.

Une très bonne source de renseignements sur les modems est la FAQ de **comp.dcom.modems**. Il existe également une **FAQ**, plutôt orienté environnements non UNIX et **PPP**, en français: <http://www.sct.fr/~philb/lesdocs.html> ou sur le forum de **news fr.network.modems**.

32.2.1 Jargon technique

Le jargon technique des télécommunications regorge d'acronymes mystérieux et de non-dits. La plupart des termes sont d'ailleurs souvent utilisés de façon multiple (avec des sens différents). Voici quelques termes avec une explication dans le contexte des modems.

⁸⁷Notez que ce protocole n'est pas transparent, au contraire de RTS/CTS: en effet, il utilise des caractères spéciaux réduisant d'autant le jeu disponible. Certains protocoles de transfert de fichiers ne permettant pas de les remplacer, on ne peut donc les utiliser. Le contrôle de flux matériel est donc recommandé.

DTE Data Terminal Equipment. Votre ordinateur personnel.

DCE Data Communication Equipment. Votre modem.

full-duplex Se dit d'une communication simultanée dans les deux sens.

half-duplex Se dit d'une communication bidirectionnelle mais non simultanée : il y a d'ailleurs un délai de changement de sens de transmission ou **turn-around delay**.

asymétrique Se dit d'une communication où un des canaux a un plus fort débit que l'autre. Cette limitation a été longtemps vue comme une optimisation : on réservait une partie d'un canal à faible débit pour l'émission dans l'autre sens. Les avantages ne sont plus perceptibles actuellement en raison des techniques de suppression d'écho (**echo-cancellation**). Certaines normes (en particulier V23) ont poussé le vice jusqu'à forcer le canal lent en fonction de la direction d'appel.

correcteur En général, on parle d'un code correcteur par abus de langage : on veut plutôt dire code détecteur suivi d'une réémission. Les codes correcteurs sont souvent bien trop coûteux pour être mis en exploitation.

asynchrone Se dit d'un transfert de données dont l'horloge n'est pas contenue dans le signal et ne fonctionne pas en permanence. En général, un événement fait commencer une horloge locale (en général d'une fréquence normalisée) et un autre événement termine la séquence. Les caractères (octets) sont transmis un à un avec un espacement entre deux caractères. On ajoute parfois une parité qui sert de contrôle de l'octet (mise à 1 ou à 0 selon le nombre de bits à 1 de l'octet). Le format le plus commun en transmission série est le format 8N1, soit 8 bits de données (1 octet), 1 bit de **STOP**, sans parité, et avec un bit de **START** implicite. On remarque donc que sur chaque octet transmis, on perd de l'ordre de deux bits, soit 20 %. La plupart des communications entre ordinateur et modem se fait toujours en asynchrone, bien que tous les modems modernes discutent entre eux en synchrone.

synchrone Se dit d'une communication dont l'horloge est jointe au signal de façon à ne pas perdre la synchronisation. En général, toute séquence de caractères (usuellement, en synchrone, les caractères sont groupés en blocs) est commencée par des fanions de synchronisation et terminée par une séquence de fin. En général, l'efficacité du point de vue des bits utiles est bien plus grande qu'en asynchrone. Cela explique que certains protocoles synchrones, comme **MNP-4**, qui ne compressent pas les données, font gagner parfois jusqu'à 20 % de vitesse, en augmentant cependant le délai de réponse pour les petites séquences.

Le **CCITT** (nouvellement **ITU-T**) s'occupe de la normalisation en télécommunications. Les normes **Vxx** concernent plus particulièrement les modems.

32.2.2 Vitesse et modulation

La vitesse de transmission se mesure en général en **bits par seconde** ou **bps**. Cette unité peut être préfixée des lettres k (pour 1000 bps, se prononce *kilo*), M (pour un million, se prononce *méga*), ou encore G (pour un milliard, se prononce *giga*). La technique se borne, pour le moment, à quelques dizaines, voir quelques centaines de Gbits pour les lignes à très hauts débit, et beaucoup moins pour les lignes d'abonnés (64kbps pour **ISDN**). De nombreuses confusions se produisent avec l'unité **baud** qui dénote, elle, le nombre de transitions de la ligne par seconde. Il faut savoir que pour transmettre un signal à travers une ligne téléphonique, il est nécessaire de le moduler. On peut moduler un signal de différentes façons :

en amplitude On augmente ou on diminue l'intensité (le volume) du signal.

en phase On décale un signal dans le temps par rapport à une référence.

en fréquence On change la longueur d'onde (tonalité) du signal.

On peut combiner ces trois méthodes de modulation. Dans les premiers modems, cependant, on utilisait en général une modulation de fréquence (**FM**) : le zéro binaire était représenté par une fréquence et le un par une fréquence différente. La vitesse en bps était donc égale à celle en bauds. Les modems plus rapides (dès 1200 bps) combinent les méthodes citées plus haut afin de coder plusieurs bits en un seul changement : en effet, la largeur de bande de téléphonie (de l'ordre de 4 kHz) ne permet pas des vitesses en FM très élevées. Ainsi, un modem à 2400 bps est en fait un modem à 4 pseudo-canaux de 600 bauds : on appelle cela la modulation en quadrature d'amplitude (anglais: *QAM*).

Les modems à très hautes vitesses (14.4kbps et au-delà) utilisent des techniques raffinées de suppression d'écho (anglais: *echo-cancellation*) (ne pas oublier que les premiers modems hautes vitesses étaient **half-duplex**!).

Un très bon texte sur le sujet est `/users/admin/info/tech/modems/The_Road_To_9600`.

32.2.3 Les standards ITU-T et propriétaires

Normes de transmission La plupart des modems incorporent plusieurs de ces normes ou standards propriétaires et offrent une négociation automatique plus ou moins réussie. La table 12 page suivante définit quelques-uns de ces standards.

Norme	émission [kbps]	réception [kbps]	notes
V34+	33.6	33.6	non encore normalisé
V34	28.8	28.8	sur réseaux fiables
V.Fast	28.8	28.8	propriétaire, idem
TurboPEP	23.0	23.0	très propriétaire, asymétrique
ASL	21.6	21.6	très propriétaire
V32terbo	19.2	19.2	propriétaire
Zyxel	19.2	19.2	très propriétaire
	16.8	16.8	idem
HST	16.8	16.8	asymétrique, propriétaire
	14.4	14.4	idem
V32bis	14.4	14.4	
V32	9.6	9.6	
V23	0.075	1.2	asymétrique figé à l'appel
V22bis	2.4	2.4	
V22	1.2	1.2	
V21	0.3	0.3	

TAB. 12 – Normes ITU-T et standards propriétaires

Normes de correction ou de compression Les protocoles de compression ne s'appliquent pas sans protocoles de correction : par exemple, si l'on sélectionne **MNP-5**, alors on aura également MNP 1 à 4. En général, et plus particulièrement pour **V42bis**, on recommande une vitesse **DTE – DCE** du double ou de quatre fois la vitesse de ligne, ainsi que l'utilisation d'un protocole de contrôle de flux (matériel) (voir section 32.2.6 page suivante). La table 13 montre les quelques standards couramment utilisés.

Norme	type	notes
MNP 1 à 4	correction	propriétaire, gain de l'ordre de 20 % dus à la transmission par paquets (2 bits sur 10)
MNP-5	compression	idem, plus gains variables, supprimer impérativement avant de transférer des fichiers déjà compressés.
V42	correction	gain également par mise en paquets
V42bis	compression	adaptatif, se supprime automatiquement si fichiers déjà compressés.

TAB. 13 – Standards de correction d'erreur et de compression de données

32.2.4 Aspects techniques

La mise en œuvre de modems n'est pas ce qu'il y a de plus simple. Le mieux est de se référer à la documentation du constructeur du modem et de votre ordinateur. Il ne faut surtout pas croire que parce que le connecteur passe, la liaison va se faire sans problème !

Connecteurs Les connecteurs les plus courants sont le 25 pôles et le 9 pôles. En général, les connecteurs du côté DTE sont mâles et les connecteurs DCE femelles. Le minimum de signaux pour une connexion série asynchrone est de trois (RXD, TXD et GND). Cependant, on utilise souvent une version à neuf fils (RXD, TXD, GND et les signaux de contrôle DTR, DSR, DCD, RI, ainsi que RTS et CTS pour le contrôle de flux). Pour la connexion de deux DTE (par exemple deux ordinateurs pour un réseau sériel) on utilise un câble **null modem** à trois ou neuf signaux.

Signaux de contrôle Les signaux de contrôle les plus courants sont les suivants :

DCD Carrier Detect. La connexion est établie.

RI Ring Indicator. Le téléphone sonne.

RTS Request To Send. DTE prêt à émettre. Souvent utilisé dans le sens prêt à recevoir⁸⁸.

⁸⁸ A l'origine, **RTS** et **CTS** étaient utilisés dans les modems **half-duplex**. L'ordinateur (**DTE**) indiquait son désir d'émettre sur la ligne en levant **RTS**. Le modem (**DCE**) devait alors inverser le sens de transmission, ce qui durait souvent quelques dizaines de millisecondes, et informer le **DTE** que l'émission était désormais possible en levant son signal **CTS**. L'usage moderne sur les liaisons habituelles (donc **full-duplex**) est d'interpréter **RTS** comme prêt à recevoir (du point de vue de l'ordinateur) et **CTS** comme prêt à émettre.

CTS Clear To Send. DCE Prêt à émettre.

DTR Data Terminal Ready. DTE prêt.

DSR Data Set Ready. DCE prêt.

Signaux de données Les signaux de données sont **TXD** et **RXD**, respectivement pour l'émission et la réception de données (en se plaçant du côté **DTE**).

Commandes AT La plupart des modems modernes disposent d'un jeu de commandes textuelles qui peuvent être appelées en mode de commande (lorsque le modem n'est pas connecté à un modem distant ou qu'on l'a remis en mode de commande local). Les modems plus anciens ne disposent que des commandes par signaux de contrôle ou de commandes sous la forme de télégrammes. La norme **V25bis** (peu utilisée, avec raison) décrit un protocole complexe que certains modems supportent également.

Le jeu le plus courant de commandes textuelles est appelé le jeu **AT** ou encore **Hayes**, du nom de l'entreprise ayant commercialisé les premiers modems dits intelligents. Le jeu se compose de commandes simples préfixées par **AT**. En général, ces deux codes sont utilisés pour synchroniser le modem à la bonne vitesse. Il est donc recommandé de les taper avec la même casse. Les commandes agissent sur des registres internes qui sont manipulables directement. Il est souvent possible de sauvegarder la configuration dans une mémoire interne non volatile. Il faut savoir que la plupart du temps beaucoup d'extensions incompatibles ont été implantées : il n'existe pas à proprement parler de norme. On parlera cependant des commandes les plus courantes. En général, les fabricants fournissent avec le modem de la documentation suffisante pour la paramétrisation.

ATDT0328414081 Composition en **mode vocal (DTMF)** du numéro 032 841 40 81. Pour le mode par impulsions, utiliser un **P** à la place du **T**.

ATA Réponse manuelle à un appel.

ATS0=1 Positionner en mode réponse après un coup de sonnerie. En Suisse, on ne peut positionner ce mode à moins de deux sonneries. En effet, une sonnerie spéciale est réservée pour les pompiers. On met un **0** pour interdire la réponse automatique. On modifie ainsi un des nombreux registres du modem.

ATH Raccrocher.

AT&W Sauvegarde de la configuration en mémoire non volatile.

+++ Cette séquence spéciale permet de revenir au mode de commande, pour raccrocher par exemple.

Protocoles de correction d'erreurs Ces protocoles sont en général utiles lors de perturbations de ligne ou simplement pour augmenter la vitesse : en effet, ces protocoles transforment les caractères en mode asynchrone (10 bits par caractère) en mode synchrone (environ 8 bits par caractère). On gagne donc de l'ordre de 20 pour cent. Il faut évidemment mettre la vitesse de la connexion ordinateur modem à une valeur supérieure à la vitesse entre les deux modems. Cela nécessite l'utilisation de protocoles de **contrôle de flux**. Notez que toute utilisation de protocole de correction (compression impliquant également correction) augmente le taux de transfert mais diminue l'interactivité. Dans certains cas bien précis, il vaut mieux s'abstenir d'utiliser tout protocole de correction ou de compression, surtout si le protocole utilisé au-dessus de la ligne est déjà correcteur (*voir section 30 page 98*).

Notez bien que la liaison entre l'ordinateur et le modem, elle, n'est pas sûre même avec un modem corrigeant les erreurs. En particulier, sans protocole de contrôle de flux comme **RTS/CTS**, la transmission se fera certainement très hasardeusement.

Protocoles de compression de données En général, ceux-ci ne servent pas à grand chose : en effet, la plupart des transferts de fichiers se font avec des fichiers déjà compressés. En général, le taux de compression obtenu par compression **on-the-fly** est bien moins bon que celui obtenu par une compression à l'avance (*voir section 7.4.1 page 43*).

32.2.5 Les options avancées

Certains modems offrent des options très intéressantes comme la réception et l'envoi de **fax** ainsi que la possibilité d'utiliser l'ordinateur comme répondeur téléphonique intelligent (**voicemail**).

On distinguera, parmi les options de **fax** les normes **Class 1** et **Class 2**, qui se différencient principalement par leur niveau d'abstraction (Class 2 est de plus haut niveau et est plus simple d'implémentation car les contraintes de temps sont moins élevées). Il faut également savoir que certains modems fax bas de gamme n'offrent pas la différenciation automatique des appels en mode données ou fax.

32.2.6 Mise en œuvre d'un modem

Les **modems** sont des périphériques très utiles⁸⁹ mais dont le maniement est facilement incorrect, menant à des erreurs de transmission ou des vitesses non optimales.

⁸⁹Même s'ils sont parfois un peu dépensiers :-)

Le premier choix à faire est celui de la communication entre l'ordinateur et le modem. En général, on le fera de façon asynchrone, avec 8 bits de données, 1 bit **START** et un bit d'arrêt **STOP**. Il est difficile de trouver autre chose sur un ordinateur personnel⁹⁰. Par contre, il faut ensuite choisir une vitesse de transmission adaptée au modem mais aussi à l'ordinateur ou au terminal : en général, pour un modem sans compression **V32bis**, on utilise une vitesse du port série de 19.2kbps. Pour un modem avec compression **V42bis** ou **MNP**⁹¹ il faut plus : par exemple 38.4kbps ou 57.6kbps, ou même 115kbps. Le mieux est de faire quelques essais de transmission de fichiers.

Notez qu'à de telles vitesses, un protocole de contrôle de flux est nécessaire et donc un câble à 7 signaux ainsi qu'une configuration adéquate de l'ordinateur et du modem sont nécessaires. On choisira en général **RTS/CTS** pour sa transparence aux données transmises. Un des symptômes de non fonctionnement du contrôle de flux sont les erreurs de transmission avec **ZModem** mais pas avec **YModem** ou **XModem**.

Il faut également que le modem sache qu'il doit maintenir sa vitesse à celle sélectionnée en mode de commande (pas d'adaptation de la vitesse **DTE – DCE** en fonction de la vitesse de connexion).

Notez que la plupart des modems autorisent le filtrage de certaines vitesses. Vous pouvez l'utiliser pour forcer une certaine vitesse lors d'une connexion ou un raccrochage pour recomposer.

Le manuel d'utilisation de votre modem parle de la configuration. En cas de doute, reportez-vous à votre logiciel de télécommunication ou à votre vendeur.

32.3 Les logiciels de communication

On distingue, parmi les logiciels de communication, ceux qui offrent une émulation de terminal (y compris graphique comme **RIP**) ainsi que des protocoles de transfert de ceux qui offrent de véritables services réseau. Ces derniers, comme par exemple **TCP/IP** à l'aide de **SLIP** ou **PPP**, ou encore **DNET** ou **term**, offrent bien plus, comme par exemple le multiplexage de la connexion, l'accès à **Internet**, les sessions multifenêtres sous **X11**.

On ne traitera cependant dans cette section que des logiciels de communication classiques.

32.3.1 Terminologie

download Transférer un fichier du serveur distant au système local.

upload Transférer un fichier du système local au serveur distant.

32.3.2 Les émulations de terminal

La plupart des serveurs **BBS** offrent des éditeurs pleine page ainsi que des écrans semi-graphiques (**VT100**) ou même graphiques (**RIP**). Cependant, il est rare que ces émulations soient complètes. De plus, des ajouts non standards ont été faits sur les dérivés de **VT100** rendant difficile un bon rendu.

32.3.3 Protocoles de transfert de fichiers

Il existe une pléthore de protocoles de transfert offrant divers avantages. La table 14 résume ceux-ci.

Protocole	description
ZModem	négociation de nom, reprise, compression, etc.
YModem	négociation de nom, diverses variantes dont YModem-g sans correction d'erreurs utile lors d'un transfert avec un modem avec correction.
XModem	lent, pas de négociation de noms, un seul fichier.
kermit	lent mais efficace en milieu non transparent, très répandu sur les mainframes .

TAB. 14 – Protocoles de transfert courants

Des protocoles bidirectionnels existent également, mais la plupart du temps ils sont propriétaires.

⁹⁰Il existe cependant d'autres possibilités, comme le transfert synchrone, qui ne sont en général pas utilisées dans le grand public au niveau de la connexion **DTE – DCE**.

⁹¹En fait, la détection d'erreur seule suffit : les dispositifs de détection d'erreurs utilisent un débit synchrone entre les modems ce qui augmente de 20 % la vitesse nominale.

32.4 RNIS

32.4.1 Introduction

RNIS signifie Réseau numérique à intégration de services (anglais: *ISDN*). Il s'agit d'un réseau commuté, analogue au réseau téléphonique normal, avec **bande passante** garantie. RNIS est une extension du réseau numérique inter-centraux sur le poste de l'abonné : la liaison est de bout en bout numérique.

En Europe, la liaison RNIS de base se compose de deux canaux à 64kbps ainsi que d'un canal de contrôle à faible vitesse. On accède à ce réseau soit à l'aide d'une carte spéciale, offrant alors toutes les possibilités (voix, données), soit avec un faux modem se connectant au port série. Dans ce dernier cas, une vitesse minimum de l'ordre de 80kbps est nécessaire (conversion synchrone/asynchrone gain de 20%). Diverses normes existent permettant le transfert de fichiers à diverses vitesses ou encore l'utilisation de deux canaux.

En Suisse, ce service est fourni sous le nom **Swissnet**. En France, on parle de **Numéris**.

32.4.2 Aspects techniques

Une communication téléphonique normale utilise deux fils de cuivre du poste de l'abonné au central, où les signaux sont digitalisés (résolution 8 bits à une fréquence de 8 kHz) avec une **bande passante** de environ 3.1 kHz. D'un point de vue strictement numérique, on transmet donc une communication dans un canal de 64kbps. La transmission sur le réseau proprement dit est effectuée de façon numérique souvent en **multiplexe temporel** (par exemple, la liaison primaire à 30 canaux plus 2 de signalisation, appelée **T1** aux ÉTATS-UNIS). A l'autre bout, il faut bien sûr régénérer le signal et faire une conversion numérique "analogique" (entre guillemets car le signal est entaché d'erreurs de quantification, une erreur provenant du processus même de digitalisation, qui doit transformer une information en un nombre fini d'états, sans considérer évidemment les éventuelles erreurs de transmission).

Le prix des équipements électroniques ayant baissé de façon très significative, la différence de prix entre équipements terminaux "tout numérique" et analogiques justifie une extension de la liaison numérique : de plus, la qualité doit bénéficier d'une plus grande constance. Des services supplémentaires sont également facilement implantables.

La liaison d'abonné standard se compose de deux canaux bidirectionnels à 64kbps et d'un canal de contrôle à 16kbps, respectivement appelés canaux **B** et **D** dans la littérature. Sans entrer dans les détails, le canal D est utilisé pour la signalisation et dans certains cas des connexions **X.25**.

Pour la téléinformatique, cela devrait signifier la fin de l'utilisation des modems en bande de base, qui souffraient injustement de la limite des 4 kHz.

En fait, on remarque que la technologie "analogique" a une telle inertie (il y a de telles sommes en jeu) que les modems sont sans cesse améliorés pour utiliser au mieux la bande passante existante, en augmentant la qualité des filtres et l'auto-adaptabilité pour permettre une augmentation du nombre de bits par moment (ou par Hz). Évidemment de telles techniques ne sont envisageables que sur des lignes de qualité (ou carrément en assurant que la liaison central vers fournisseur est numérique ou a des caractéristiques correctes) et nécessitent des modems complexes. Il est triste de constater que toute cette technologie est utilisée pour une application obsolète, qui reste malheureusement la seule utilisable dans de nombreux pays qui n'offrent pas une palette numérique à prix abordable et de qualité.

La seule chose qui est certain est que tant qu'une partie de la ligne entre les deux modems est non numérique, la vitesse en bit/s ne peut pas égaler un canal similaire tout numérique : de plus, à volume de vente égal, la solution numérique sera moins cher et plus puissante. En plus, les avantages techniques de **ISDN** sont évidents.

Du côté de l'abonné, le système **ISDN** se compose d'un bus où sont connectés les appareils. La connexion avec la ligne se fait avec un **TA** ou adaptateur de terminaison, qui peut disposer de sorties analogiques permettant d'utiliser des téléphones normaux ou des modems. Au maximum 8 terminaux peuvent être branchés, et comme on dispose de deux lignes externes, deux appareils peuvent être utilisés simultanément. La transmission numérique pure se fait à 64kbps, mais il est également possible de lier deux canaux pour la transmission numérique, qui sera évidemment facturée plus cher. Notons également qu'il est l'usage sur les réseaux **ISDN** de ne pas facturer au volume (comme **X.25**) mais à la durée (comme le téléphone normal), sauf que l'établissement des communications (réussi ou non) est également facturé en plus de la taxe de base.⁹²

Un autre aspect intéressant est la possibilité d'affecter des numéros de téléphone à des fonctions. On peut par exemple ainsi utiliser un seul "**modem**" numérique pour gérer des connexions numériques, ou analogiques (modem, fax, serveurs vocaux) et différencier la fonction par un numéro différent. L'utilisation simultanée avec un seul appareil n'est pas possible, mais l'intérêt réside dans la distinction des types d'appels.

⁹²Apparemment cela ne se passe qu'en cas d'émissions de messages hors-bande.

Glossaire

- ACL** Les *Access Control Lists* permettent de contrôler l'accès à des objets du système de fichiers plus précisément que ce que les permissions UNIX usuelles permettent. Pour en bénéficier, il faut un système de fichier les supportant. Sous Linux, on a par exemple : **xfs** et des patches pour **ext3** existent.
- adresse IP** Une adresse IP permet d'identifier les ordinateurs connectés à Internet. Elle se compose de quatre nombres séparés par un point, chaque nombre pouvant valoir de 0 à 255 (avec quelques contraintes). Généralement, les adresses des sites sont données sous forme symbolique (par exemple `satcom.alphanet.ch`) et un **nameserver** se charge de transformer l'adresse symbolique en une adresse numérique, comme `193.72.186.2`.
- AFS** Ce système de fichiers spécial, nommé aussi Andrew File System, est à la fois le successeur de **NFS** et le prédécesseur d'un système plus général nommé **DFS**, implémentant de façon efficace la distribution de données sur plusieurs serveurs.
- alex** Système de fichiers permettant de considérer les serveurs **FTP** anonymes comme faisant partie de la même arborescence qui est montable localement. Cela signifie que l'on peut utiliser l'arborescence des serveurs FTP comme un seul système de fichiers auquel on accède par les méthodes usuelles. On peut donc utiliser n'importe quel utilitaire local sur ces fichiers ou catalogues distants.
- ALPHANET NF** Club de recherche, information et enseignement orienté vers les télécommunications. Il soutient de nombreux projets en télécommunications. Le club fournit à ses membres des services réseau pour le support de projets. Ces services sont regroupés sous le nom ALPHANET. Ce réseau existe depuis 1990. Des informations sur l'association sont disponibles sur le serveur de l'association (*voir section 4.3 page 17*) ou en envoyant un **mail** à `admin@alphanet.ch`. Un serveur **WWW** local, accessible avec l'URL `http://www.alphanet.ch`, offre également des informations.
- Aminet** Réseau de sites **ftp** partageant des archives Amiga.
- archie** Un système réparti qui permet d'effectuer des recherches de fichiers dans l'ensemble des serveurs **FTP** du monde entier en quelques minutes. Des interfaces via **mail**, **telnet** ou encore **X11** existent. Vu le nombre d'informations disponibles, les recherches de fichiers doivent être faites avec le plus grand soin et la plus grande précision pour limiter le nombre de lieux retournés.
- ARP** Address Resolution Protocol. Un protocole permettant la correspondance entre adresse **Ethernet** depuis et **adresse IP**. La correspondance inverse est également possible : on utilise alors **RARP**.
- ARPAnet** Advanced Research Project Agency network. Un projet du **DARPA**, débuté en 1969, qui avait pour but à l'origine de trouver un moyen sûr d'interconnecter des ordinateurs militaires pour une utilisation lors d'une guerre nucléaire totale, dans le contexte de la guerre froide. Très vite, ce projet a amené l'idée d'une confédération de petits réseaux et de la redondance des lignes, s'opposant au modèle fermé des lignes directes entre sites. Ce projet a donné naissance à divers protocoles, dont le protocole **TCP/IP** actuel (version 4 en 1986) et surtout au réseau **Internet**.
- ASCII** American Standard Code for Information Interchange. Jeu standard de caractères usuels, sans caractères accentués. Un autre jeu (anecdotique) encore en utilisation sur les **mainframes** IBM est **EBCDIC**. La commande UNIX `dd` permet de convertir entre ces deux jeux.
- atomique** Se dit d'une opération élémentaire insécable qui n'est pas soumise à des problèmes d'accès parallèles. Certains processeurs disposent d'une instruction machine, **TAS**, qui teste, puis assigne, atomiquement, le contenu d'une case mémoire. Cette instruction permet la réalisation de verrous de protection de ressources contre l'accès concurrent.
- AT&T** Un des vendeurs de l'un des systèmes d'exploitation les plus connus, AT&T System V Release 4 (en sous-licence chez presque tous les constructeurs, de DEC à IBM en passant par Bull, Apple, Silicon Graphics, Sun, HP, Microsoft, etc.). Accessoirement, un des leaders du marché des télécommunications.
- authentification** L'action de vérifier l'identité d'un correspondant ou d'un utilisateur, par exemple avec un système de signatures électroniques ou un échange de codes secrets, comme avec un mot de passe ou un système à réaction (anglais: *challenge-response key*).
- backbone** Historiquement, il s'agissait d'un site avec une connectivité élevée et pouvant dicter ses conditions aux machines se connectant. Ensuite, ce terme est devenu synonyme de grande artère d'un réseau.
- bande passante** La bande passante d'un réseau est le débit maximum que celui-ci peut supporter.
- batch** Littéralement : traitement par lot. Un travail batch est un travail qui, généralement, est décalé dans le temps par rapport à sa soumission et qui comporte souvent de nombreuses opérations ne nécessitant pas d'intervention de l'utilisateur. Par extension, le terme batch a pris le sens de **script**. On retrouve également ce terme en transmission de données **store-and-forward** : il s'agit alors d'un ensemble de textes et de fichiers, éventuellement compressés, en attente d'un transfert.
- BBS** Bulletin Board System. Un serveur d'information (messagerie, fichiers) accessible par modem ou **telnet**. Souvent, ces systèmes sont administrés de façon bénévole et reliés les uns aux autres par un réseau **store-and-forward** du type **FidoNet**.
- big endian** Se dit d'un format de représentation machine de données dont les éléments constitutants sont présentés avec le poids faible en premier, par opposition au format **little endian**. Exemples :
- l'ordre des numéros de **bits** dans un **octet** est 76543210 : c'est un ordre typiquement **big endian**.
 - l'adresse `0xC020AB12` se représente, en mémoire, sous la forme `0xC0, 0x20, 0xAB, 0x12`, sur les systèmes **big endian** : Motorola 68000, SPARC, PowerPC, et dans l'ordre inverse sur les systèmes **little endian** comme VAX, Intel ix86 et Alpha.

Lequel des deux est plus logique ? On peut trouver des arguments pour les deux parties. Ma préférence personnelle va aux architectures **big endian**. Il y a aussi des architectures mixtes dont l'ordre change avec la taille de mot, par exemple, ou dont le système d'exploitation et chaque programme utilisateur peut travailler dans un mode différent (processeurs MIPS récents).

bit Binary unit (ou binary digit). Une idée fondamentale binaire d'information pouvant prendre deux valeurs : 0 ou 1.

BITNET Because It's Time Network. Un réseau assez répandu dans les milieux académiques européens (substitut à Internet pendant longtemps). Composé de lignes louées formant un arbre (en général sans cycle), proposant des services principalement de messageries (les fameuses listes de distribution BITNET), ce réseau a été peu à peu remplacé par Internet. Il existe encore quelques sites principalement en Grande-Bretagne et aux Etats-Unis et dans les pays Arabes.

BNU Basic Networking Utilities. Un composant du système d'exploitation SVR4 d'AT&T comportant entre autres une version de **UUCP**, qui est désignée comme telle. On connaît également ce logiciel sous le nom **HDB**, pour des raisons bizarres et surtout obscures (HoneyDanBear).

broadcast Le fait de transmettre une information à toutes les machines d'un sous-réseau.

BSD Berkeley Software Distribution. Un UNIX gratuit développé par l'université de Berkeley en Californie. Gratuit mais pas du domaine public. System V est basé en grande partie sur celui-ci pour toutes les extensions réseaux (entre autres).

BSMTP Batched Simple Mail Transfer Protocol. Protocole de transfert de mail généralement compressé en vue d'une transmission par un protocole **store-and-forward**.

buffer Un tampon (zone de mémoire généralement temporaire) servant à stocker des données.

bug Erreur de programmation dans un logiciel. Ce mot vient de l'**ENIAC**, le premier ordinateur électronique. Une panne avait été diagnostiquée provenant d'une lampe. Auprès de celle-ci, la légende raconte qu'il y avait un insecte. Certains "traduisent" ce mot par *bogue* — du plus mauvais effet.

C Langage de programmation très commun sous environnement UNIX.

cache Un cache est un moyen de rendre l'accès à l'information moins coûteux (que ce soit en terme de temps d'accès, durée ou coût réel). Par exemple, on peut installer un cache **FTP** ou **WWW** qui permet de stocker sur un serveur d'un réseau local les pages récemment accédées, ce qui peut parfois mener à des gains appréciables et éviter de surcharger des lignes externes.

cancel Se dit d'un message de **news** possédant un format particulier et permettant de supprimer un message posté précédemment. Certains lecteurs de news offrent cette possibilité (la commande *D* de **tin** en mode de lecture d'articles, par exemple).

CERT Computer Emergency Response Team. Un organisme d'une université des Etats-Unis chargé de la communication des problèmes et des solutions en matière de sécurité⁹³. Plus généralement, un ensemble de personnes à même de gérer des alertes de sécurité sur un réseau d'une certaine envergure. De telles alertes sont postées dans le forum de news pré-cité.

chain-letter De temps en temps, un message est posté sur quelques forums, proposant une méthode pour gagner de l'argent rapidement. En général, ces méthodes sont basées sur un schéma pyramidal et les promesses ne peuvent être tenues sur le long terme. Par contre, ces messages génèrent en général un volume de messagerie très grand. Le mieux à faire est d'ignorer ces messages (*voir section 23.6 page 83*).

client Un client est un logiciel demandant des services à un programme local ou à distance qui gère l'information, le **serveur**. On nomme client-serveur le concept d'interaction sous-jacent. Par extension, il s'agit aussi de la machine qui exécute des programmes clients.

compress Programme et méthode de compression assez peu efficace, mais disponible en standard sur tout système UNIX. Les fichiers compressés avec cette méthode se terminent par convention par *.z*. La compression *compress* tend à disparaître dans le monde UNIX pour être remplacée par *gzip*, de la même façon que la compression *pack* a été remplacée par *compress* dans le passé.

core Fichier généré par une défaillance logicielle d'une application ou du système d'exploitation lui-même, pour un debugging ultérieur (debugging post-crash). En général, la génération d'un fichier **core** est précédée du message *core dumped* précédé de la cause de la panne (par exemple *Segmentation fault*, *Bus error* pour des erreurs inhérentes au logiciel, ou *Quit*, pour un arrêt de panne demandé par l'utilisateur (*voir section 5.9.18 page 27*)).

cracker Un cracker est un utilisateur d'une machine qui essaie de franchir des systèmes de sécurité divers (protection logicielle, matérielle, chiffrement, mot de passe, etc.) afin d'obtenir (de voler) des données. En général, un cracker doit très bien connaître le système qu'il veut pénétrer, d'où la confusion avec le terme **hacker**.

crash On appelle crash (ou plantage, défaillance système, etc.) l'arrêt prématuré d'une application ou d'une machine en raison d'une panne, le plus souvent logicielle, du système ou d'une application. Sous UNIX, le crash d'une application ne provoque en général pas l'effondrement de tout le système. En plus, selon la paramétrisation de l'environnement de l'utilisateur, une image mémoire du processus concerné est copiée sur disque à des fins de diagnostic après crash (debugging, déverminage) dans un fichier **core**.

crashmail Un mode de transfert de courrier privé sur le réseau **FidoNet** qui revient à appeler directement le site où doit être envoyé le message (via modem et téléphone). Les sites acceptant le crashmail doivent être disponibles 24h/24 et listés dans la **nodelist** avec un drapeau (anglais: *flag*) spécial.

⁹³principalement dans le **forum comp.security.announce** sur USENET

cross-poster Le fait de spécifier plus d'un forum dans le champ `Newsgroups`: en séparant les forums successifs par des virgules, sans espaces, comme dans

```
Newsgroups: alphanet.local,alphanet.misc,ch.general,fr.misc.divers
```

et donc de permettre aux personnes ne lisant que certains des forums cités d'avoir accès aux messages. Il faut être sûr que les forums cités sont réellement appropriés pour le message considéré. Si le logiciel de **news** le supporte, le message ne sera présenté qu'une seule fois aux utilisateurs ayant souscrit à plusieurs de ces forums. Notez que le message sera également transféré de façon unique aux sites distants. Le logiciel **tin** supporte cette facilité. Notez qu'un message posté plusieurs fois au lieu d'être cross-posté ne sera pas détecté par ce mécanisme.

CSMA/CD Carrier Sense Multiple Access with Collision Detection. Un système utilisé entre autres par **Ethernet** pour l'émission sur un réseau. Pour une station du réseau, le principe est très simple : j'écoute un moment (Carrier Sense), si rien ne vient j'émet. S'il y a collision car deux stations ont émis en "même" temps, on attend un temps aléatoire et on recommence le cycle. L'avantage principal de ce type de réseau est sa simplicité. En cas de trafic élevé, il a le désavantage de perdre rapidement son efficacité en raison de collisions nombreuses. En général, on segmente alors le réseau en plusieurs sous-segments.

CSMA/CR Carrier Sense Multiple Access with Collision Resolution. Un système utilisé entre autres par **RNIS** ou **SCSI** (bus parallèle) pour l'émission sur un bus partagé : si deux stations émettent en même temps, alors une règle précise et déterministe définit qui prend le contrôle du bus : respectivement par exemple le nombre de zéro dans les numéros émetteur et destinataires, ou la priorité (l'ID SCSI) du périphérique. Par rapport à CSMA/CD, ce système offre des performances en charge très bonnes, au prix, dans certains cas, d'un biais (famine de certains périphériques) suivant l'algorithme déterministe utilisé.

CSLIP Version optimisée du protocole SLIP permettant une bonne interactivité.

daemon Disk And Execution Monitor. Programme du logiciel système, s'occupant à l'origine des entrées/sorties, mais qui sur les systèmes UNIX a pris la signification d'un logiciel sans contrôle direct d'un utilisateur (notion d'arrière-plan). La plupart des serveurs tournent sous la forme de daemons.

DARPA Defense Advanced Research Project Agency. L'agence du gouvernement américain (on peut même dire du Pentagone) qui a subventionné la recherche dans le domaine de nouveaux protocoles distribués ayant donné naissance au réseau militaire, gouvernemental et académique **ARPAnet**, puis aux protocoles **TCP/IP** et au réseau coopératif **Internet**.

DCE Distributed Computing Environment. Derrière ce nom barbare se cache un modèle d'interaction distribuée entre systèmes, un concept de création d'applications et un support pour l'administration de grands réseaux ainsi que pour leurs utilisateurs. Basé sur une version étendue des **RPCs**, il comporte de nombreux services prédéfinis ainsi que des spécifications de niveaux de sécurité (authentification, système de tickets, etc.). Ce nouveau standard est amené à remplacer les anciens protocoles basés sur **RPC**, comme **NFS** (voir **DFS**) mais aussi les outils réseaux comme **DNS** et divers autres services. Les avantages principaux de **DCE** est la facilité avec laquelle il gère de petits réseaux comme des réseaux planétaires (anglais: *scalability*) et sa relative facilité d'emploi alliée à son intégration (ce qui est parfois un problème suivant les implémentations).

debugging On entend souvent par debugging la mise au point de programmes. Des termes équivalents existent en français : déverminage, débogage, mise au point. Le debugging après crash consiste en l'examen d'un fichier **core** généré lors de la panne d'une application, à l'aide d'utilitaires spécialisés.

DES Data Encryption Standard. Un algorithme de chiffrement décrit de façon exhaustive et cependant relativement peu sûr, utilisé dans un autre but (comme fonction de hachage) pour les mots de passe sous UNIX.

DFS Ce système de fichiers distribué (Distributed File System) implémente les services habituels de **NFS**, avec en plus des mécanismes supplémentaires d'authentification et de chiffrement, sans oublier l'indépendance de localisation des objets. De plus, ce système se prête mieux au partage de fichiers sur une grande et très grande échelle : **NFS** a été conçu pour quelques dizaines de machines partageant des fichiers (avec un ou deux serveurs), **DFS** pour quelques milliers ou dizaines de milliers de machines avec autant de serveurs. **DFS** est basé sur **DCE**.

DNS Domain Name Server. (voir section 31 page 100)

domain Une entité administrative du point de vue des mails ou de l'adressage IP.

DVI Device Independent. Format de fichier produit par le logiciel **T_EX**. Il décrit un document d'une façon similaire à **PostScript**, indépendante du périphérique de restitution jusqu'à un certain point.

EchoMail Messagerie publique sur réseaux à base de technologie **FTN**, correspondant aux **news**.

EFF Electronic Frontier Foundation. Une association dont le but est de protéger les droits des utilisateurs du réseau Internet, ainsi que de faire connaître ses possibilités.

Ethernet Un réseau "rapide" (10 megabits/s) disponible pour une grande variété de systèmes et développé au fameux XEROX PARC. Contrairement à **Token Ring**, sa structure n'est pas en anneau mais filaire. Sa technologie est **CSMA/CD**.

FAQ Une FAQ (de l'anglais, Frequently Asked Questions) désigne un résumé de réponses aux questions courantes. La plupart des forums ont leur FAQ postée une fois par mois ou en tous cas de façon régulière : il est très important de les consulter avant de poster. Une grande partie des FAQs sont stockées automatiquement sur des serveurs d'archivage, comme `rtfm.mit.edu:/pub/usenet`. Certains "traduisent" FAQ par Foire aux questions.

- FFS** Fast File System. Le nom d'un système de fichiers très répandu défini pour la version **BSD** du système UNIX, puis adopté par **System V Release 4** comme système de fichiers standard remplaçant le plus ancien **UFS**. Ses caractéristiques principales sont son efficacité (du point de vue vitesse d'accès mais aussi optimisation des ressources disque) et la liberté qu'il laisse à l'utilisateur concernant les noms de fichiers et de répertoires (limités à 255 caractères, contre 14 pour **UFS**).
- FidoNet** Réseau **store-and-forward** amateur de fichiers, messages privés et publics sur réseau téléphonique (à part entre les Etats-Unis et l'Europe, où la liaison passe par InterEUnet). L'adressage **mail** est assez caractéristique : 2:301/315.50 est une adresse de machine typique.
- finger** Protocole TCP (et commande UNIX) qui permet à une personne connectée sur un ordinateur relié à Internet via TCP/IP de savoir :
- qui est connecté sur un ordinateur distant,
 - si un utilisateur existe sur un ordinateur distant,
 - et s'il existe, des informations plus détaillées sur cet utilisateur (nom complet, numéro de téléphone, et aussi s'il a lu son mail, depuis quand il est connecté, et depuis quand il a quitté son clavier)
- firewall** Un firewall est une machine qui sépare un réseau interne **TCP/IP** d'un réseau global. Sur cette machine tournent divers processus qui relaient les requêtes internes au réseau global, processus appelés proxies .
- foobar** Originellement un terme militaire, *fooled up beyond all recognition*, signifiant à peu près : démoli dans des proportions ne permettant aucune identification, ce terme ou une variation est souvent utilisé pour remplacer un quidam ou un objet quelconque : foo.com dans les exemples.
- forum** La traduction française de newsgroup. Il s'agit, dans le jargon des **BBS**, d'une section de messages publics (**news**).
- FTN** Se dit des réseaux à technologie inspirée de **FidoNet**, organisés de façon hiérarchiques et basés sur un protocole **store-and-forward** et des règles strictes.
- FTP** File Transfer Protocol. Basé sur **TCP**, il permet de recevoir ou émettre des fichiers sur un site connecté via Ethernet, SLIP, **X.25**, ou autres supportant le protocole TCP/IP. Souvent les sites FTP contiennent des archives du domaine public et sont atteignables de façon anonyme, sans devoir posséder un accès (compte) sur la machine distante, on appelle cela du **anonymous FTP**. Exemple de site FTP : nestor.epfl.ch avec environ 1.2 tera d'espace virtuel disque. Autre exemple très connu, amiga.icu.net.ch, l'ancien amiga.physik.unizh.ch, avec 3 GB de disque. Sur notre réseau local, le serveur ftp.alphanet.ch fait office de serveur de fichiers anonyme.
- FTPMAIL** Petit programme qui reçoit des demandes de réception de fichiers par mail et expédie le résultat par mail. Notre version actuelle permet aussi d'obtenir des fichiers d'Internet via FTP. L'adresse à laquelle il faut envoyer les requêtes est *ftpmail@alphanet.ch*. Ce serveur est seulement accessible en local.
- FYI** For Your Information. Documentation traitant d'Internet pour les utilisateurs. Ces documents sont archivés avec les **RFC**.
- GNU** GNU signifie GNU is Not UNIX, récursif, ce qui signifie que dans la définition de GNU, GNU est utilisé (*voir section 2.5 page 12*).
- Gopher** Système d'information précurseur de WWW.
- GPL** General Public License. Licence définissant les conditions d'utilisation et de copie de logiciels maintenant le droit pour l'utilisateur final d'obtenir les sources et de conserver les mêmes droits que ceux initialement cédés par l'auteur du logiciel (*voir section 2.5.2 page 12*).
- hacker** Utilisateur maîtrisant un domaine particulier de façon exceptionnelle. On peut citer le **Jargon file** où se trouvent expliqués certains termes utilisés par les "connaisseurs". Une erreur à ne pas faire est de confondre le terme hacker avec le terme **cracker**, qui est, lui, la désignation d'une forme de vie maligne dégénérée.
- hypertexte** Un document hypertexte est composé de texte mais aussi d'une structure logique permettant de faire des références (explicitement sélectionnables) à d'autres parties du document et de d'y accéder directement selon un parcours personnel (non linéaire au contraire d'un texte habituel). Une extension de ce système permet l'inclusion de multimédia et de références à des documents distants. (*voir section 28 page 91*)
- inode** Un descripteur de fichier d'un système de fichiers sous UNIX. Par extension, l'entier représentant ce descripteur. Plusieurs entrées de catalogue peuvent pointer sur le même **inode** (voir **link**). Le nombre d'inode est en général fixé à la création du système de fichiers.
- Internet** Un vaste réseau, au départ militaire et gouvernemental, puis académique (universités), et qui s'ouvre depuis quelques années au monde commercial et aux amateurs. Il ne s'agit pas d'un réseau **store-and-forward**, comme FidoNet, ou UUCP, mais d'un vaste réseau local (ou un réseau de réseaux) connecté par des liaisons rapides à 64kbps, 128kbps, 2 Mbps, 100 Mbps, etc. Le réseau Internet comprend de l'ordre de 3 millions de machines, avec une augmentation moyenne de 30'000 machines par mois approximativement. Le nom Internet vient d'un rapport du **DARPA** qui considérait l'interconnexion de plusieurs réseaux afin de former un internet⁹⁴. Un internet est donc une fédération de réseaux. L'Internet (la majuscule peut en général suffire à faire la différence) est le réseau mondial des réseaux coopératifs en temps réel (le sens de ces derniers termes est à prendre comme : "connecté", et non pas au sens des systèmes temps réels).

⁹⁴Même en français, on a créé le jeu télévisé Interville et les Intercity.

interopérabilité A l'origine, les protocoles étaient définis pour une plate-forme, puis implémentés de façon unique. Le logiciel final était donc en fait le standard lui-même. Lors de la définition des nombreux protocoles de la suite TCP/IP, cette approche n'était pas réaliste : il existait autant de plates-formes différentes que de logiciels écrits par des équipes concurrentes. La question de l'interopérabilité (littéralement les échanges entre ces logiciels) se posa. On dut donc réfléchir à la définition de standards permettant cette interopérabilité. Encore aujourd'hui, les problèmes principaux dans la gestion des réseaux sont de cette nature. Ces exigences aboutissent finalement à des spécifications plus précises et donc des logiciels de meilleure qualité. La concurrence aidant, des améliorations sont sans cesse proposées. Un système fermé (composé de quelques constructeurs) n'aurait pas permis à Internet de voir le jour.

intranet Ce mot a été créé de toutes pièces comme jargon à utiliser de préférence dans un contexte de vente de produits, dans la même ligne que Multimédia, 32 bits, PC Internet et Active :-). Ce que ce mot veut dire est encore assez flou. Mais il y a deux acceptions plus ou moins comprises :

sens général un intranet est un internet local, soit un réseau informatique composé d'une fédération de réseaux locaux utilisant des protocoles de la suite TCP/IP, en particulier ceux communément utilisés sur **Internet**. Les spécialistes préfèrent alors tout simplement **internet** (i minuscule, en ajoutant local si la discussion est parlée). En bref, un intranet est quelque chose que nous connaissons, qui a simplement été *remarketé*.

sens plus restrictif/commercial un réseau informatique d'entreprise utilisant des solutions plus ou moins ouvertes de communication (par contraste avec des solutions fermées comme Lotus Notes), en particulier en utilisant WWW et l'hypertexte HTML comme environnement. Dans ce cas, on insiste plus sur le côté intégration de logiciels propriétaires à un environnement ouvert basé sur des protocoles à la mode comme ceux utilisés par les **netsurfers** – typiquement pas des protocoles OSI ou propriétaires, par exemple, jugés casse-pieds et pas *netsurfants* (à tort ou à raison, mais à mon avis ce n'est pas la bonne raison de les jeter :-)). Sans politiquement correct, je dirais que intranet est surtout une méthode que certains vendeurs ont trouvé pour ne pas jeter complètement leurs solutions désuètes et continuer à faire beaucoup d'argent. N'oublions pas qu'une grande partie des protocoles et services proposés sur un intranet sont disponibles sous licence libre et souvent gratuitement sur des environnements UNIX depuis de nombreuses années.

Notons qu'un internet (**intranet**) peut être ou non connecté à **Internet** (l'Internet global, majuscule). Mais en général, un intranet se situe dans un environnement commercial, éventuellement migré depuis un environnement propriétaire, et donc connecté à Internet via au mieux des passerelles, au pire par un système étanche (p.ex. carrément un firewall, avec peu de services supportés, typiquement SMTP et HTTP).

IPX Protocole d'échange de trame remplissant des fonctions similaires à **IP**, notamment sous environnements Novell anciens. Utilise une autre variante du protocole Ethernet qui n'utilise pas un type mais une longueur dans l'entête. Peut malgré tout coexister avec **IP** vu que la valeur du type de ce protocole est plus grande que tout celle de paquet Ethernet.

IRC Internet Relay Chat. Un système de conférence textuel en temps réel où les participants peuvent discuter en même temps dans des canaux de discussion ou de façon privée. Le système est basé sur un réseau de serveurs répartis dans le monde entier.

ISO Latin 1 Norme **ISO 8859-1**. Jeu de caractères étendu comportant des accents standardisés. C'est le jeu standard sur **Amiga** et sous UNIX.

ISOC Internet Society. Une association à but non lucratif visant à informer et influencer le fonctionnement du réseau Internet.

Jargon file Ce texte est une compilation de définitions de jargon UNIX et hackers. Il est disponible sous plusieurs formats, comme **ASCII**, **T_EX** et **Info** (système **hypertexte GNU**) dans le catalogue :
/users/admin/info/unix/jargon.

joker Caractère permettant de remplacer un ou plusieurs autres caractères dans une recherche (*voir section 5.6 page 20*).

junk mail Terme désignant le courrier électronique privé non sollicité.

kermit Protocole de communication très lent, car peu optimisé, mais possédant comme avantage sa popularité (même s'il est détrôné par **ZModem** ou encore les protocoles TCP/IP) et sa résistance aux environnements limités (7 bits, caractères interdits, longs délais de réponse, etc.).

kernel On nomme ainsi le noyau, le centre d'un système d'exploitation, en particulier UNIX : un kernel s'occupe en règle générale de gérer les aspects les plus bas niveau du système, et offre des concepts comme : la mémoire virtuelle, le multi-processus, le support de parallélisation et les fichiers. Les programmes en espace utilisateur (anglais: *user-space*) appellent le kernel via des appels systèmes (anglais: *system calls*).

link Un lien est une référence à un autre fichier ou à un répertoire. Par exemple, sur mips, le répertoire :

/users/admin/info

est en fait situé sur /share/archives/info. De même, le répertoire /scratch est lui situé sur /d3/scratch. Ou encore l'exécutable /usr/lib/sendmail est un lien à l'exécutable /usr/sbin/bin/sendmail. La commande `ln -s fichier référence` crée la référence symbolique référence à un fichier fichier. Avec la commande `ls -F` on peut distinguer facilement un lien d'un fichier normal : un lien est affiché suivi d'un @. Il existe deux sortes de lien : le lien physique (ou hard) et le lien symbolique (ou soft). L'argument `-s` de la commande `ln` spécifie un lien symbolique. Les liens symboliques fonctionnent sur des partitions différentes et sont d'une utilisation plus aisée. Les liens physiques sont implémentés directement au niveau de la référence du fichier (**inode**) d'une partition et ne peuvent pas être différenciés.

- Linux** Implémentation compatible avec UNIX **POSIX** (orientée System V pour les interfaces systèmes et l'administration, mais aussi **BSD** pour certains aspects) diffusable sous les conditions de la **GPL**, tout en restant la propriété de ses auteurs (Linus TORVALDS et al.).
- little endian** Se dit d'un format de représentation machine de données dont les éléments constituants sont présentés avec le poids faible en premier. Voir l'entrée **big endian** pour plus de détails.
- load average** Charge moyenne d'un système, telle que visible avec la commande `uptime`. En général s'exprime en pourcents ou en normalisant avec 1. Sur les systèmes à temps partagé, une charge dépassant l'unité ou 100 % indique que plusieurs tâches sont en cours d'exécution et attendent le processeur.
- logfile** Un fichier qui contient un journal des opérations effectuées et des problèmes rencontrés par un système d'exploitation. Par extension, un relevé des opérations de n'importe quel programme.
- mainframe** Un gros système centralisé tel que commercialisé par IBM, DEC et les autres constructeurs. On parle plus volontiers de nos jours de **serveur**.
- mail** Messages privés (expéditeur vers destinataire, en général uniques). On parle aussi de courrier électronique privé.
- mailing list** Une mailing list est un forum de discussion par courrier électronique diffusé seulement aux membres de la mailing list. En français, on parle également de liste de distribution. Une administration se charge d'ajouter et d'enlever des personnes (adresses de courrier électronique) sur demande (en général, si la mailing list se nomme `something` l'administration est disponible en envoyant un message à `something-request`. Il faut aussi connaître quel site redistribue les messages). Les avantages principaux sont que l'accès aux **news** n'est pas nécessaire (le courrier électronique étant plus répandu que les news) et que l'on peut contrôler facilement qui y a accès. Par contre, à la fois l'administration et la charge sur le réseau sont plus élevées. Une solution adoptée localement est de centraliser toutes les mailing lists et de les transformer en forums, ce qui limite le trafic mail entrant (*voir section 22 page 71*).
- mailservers** Programmes automatiques accessibles par une adresse **mail** permettant l'exécution de commandes (par exemple pour la recherche de documents) à travers un réseau de classe **store-and-forward** (*voir section 21.1 page 70*).
- MIME** Multipurpose Internet Mail Extension. Un standard permettant l'envoi d'accents, images, sons, etc. via le courrier électronique. Par extension, un système permettant de classer les types des fichiers : par exemple `application/binary`, `image/png` ou `application/postscript`.
- mips** Le système **mips**, fonctionnant sous UNIX offre les services de transmission de messages privés et publics, serveur de fichiers, serveur **UUCP**, serveur de calcul, serveur anonyme FTP, serveur anonyme nuucp, serveur zmodem, serveur crash-mail, passerelle FTPMAIL, serveur fax, serveur de clés publiques pour **PGP**, passerelle **FidoNet**. De plus, des connexions interactives sont également possibles grâce au **login** guest ainsi que des accès directement à l'interface de commande (**shell**).
- modèle OSI** Le modèle OSI permet de représenter les services d'un réseau de télécommunication en les regroupant en classes, ou niveaux, selon le type de fonctions qu'elles effectuent. Par exemple, une liaison TCP/IP entre deux systèmes est composée de la définition matérielle du support (Ethernet 802.3 : mécanique des connecteurs, tensions électriques, type des paquets sur le réseau), du protocole à paquet IP, des protocoles de transport (TCP/UDP), des protocoles de haut niveau (TELNET, SMTP, NNTP, HTTP, ...) ainsi que des applications utilisant ces protocoles (p.ex. Mosaic). Les premières pages de l'annuaire téléphonique papier des PTT suisses comportent une courte introduction à ce modèle.
- moderated** En français, se traduit par modéré. Se dit d'un **forum** dans lequel les contributions se font par **mail** à un modérateur qui trie et rejette éventuellement certains articles. L'avantage principal est la valeur élevée du rapport signal sur bruit : le forum concerné ne contient en général que des messages pertinents.
- Motif** Un système de développement d'interfaces graphiques sous X11. Celui-ci propose un certain **look-and-feel**. Ce **toolkit** est commercial et souvent disponible avec les systèmes d'exploitation UNIX commerciaux.
- mud** Multi User Dungeon. Jeu multi-utilisateur souvent textuel et apparenté aux jeux de rôles. Voir **netlag**.
- MX forwarder** Mail eXchange forwarder. Se dit d'un site s'annonçant, par l'intermédiaire du système **DNS**, comme pouvant rapprocher le courrier électronique privé à destination d'un site (ou domaine) donné.
- nameserver** Un serveur de correspondance entre adresse IP et nom de machine. On parle aussi de serveur **DNS**.
- net** Il s'agit d'une circonscription administrative dans les réseaux **FTN**.
- netmask** Un netmask permet de définir un sous-réseau afin de faciliter le routage ou le **broadcast**.
- NetMail** Courrier électronique privé (**mail**) sur réseaux à base de technologie **FTN**.
- netlag** Mauvaise interactivité d'un réseau, bien connue des joueurs de **mud** et des utilisateurs d'**IRC**.
- news** Le système de news **USENET** permet d'écrire dans des forums électroniques publics classés par thèmes. Les lecteurs peuvent ensuite répondre aux articles et les commenter, en public ou en courrier électronique privé (**mail**).
- NFS** Network File System. Permet d'utiliser un disque d'une machine distante comme un disque local. Associé à NIS (YP), il permet également des contrôles d'identité et d'accès. Associé à AFS, il permet un contrôle étendu des machines autorisées. Associé à **alex**, il permet de considérer les sites FTP anonymes comme un énorme système de fichiers. Le serveur de **mips.alphanet.ch** exporte les disques `/export`, `/aminet`, et `/share/archives`. Également décrit dans un **RFC**, ce protocole est basé sur **RPC** et a été spécifié par Sun Microsystems.

- NIS** Network Information Service. Permet de simplifier l'administration d'un réseau de stations comportant des utilisateurs identiques.
- NNTP** Network News Transport Protocol. Protocole de transfert de news utilisant le protocole TCP sur Internet. Un serveur, nommé **nntpd**, fonctionne sur **mips.alphanet.ch**.
- node** Terme traduisible par **noeud**, ou **site**. Dans le cas des réseaux **FTN**, désigne souvent un **BBS** membre d'un réseau, qui redistribue en général à l'échelon hiérarchique inférieur (le **point**) et reçoit ses données de l'échelon hiérarchique supérieur, le **hub**.
- octet** Un ensemble de 8 bits (anglais: *byte*).
- PGP** Pretty Good Privacy (*voir section 7.6.1 page 44*).
- pipe** Un mécanisme puissant de composition de commandes complexes basé sur les pipes UNIX, des tubes (tuyaux) de données permettant de connecter entre eux des processus évitant l'utilisation de fichiers temporaires et améliorant le parallélisme. En généralisant, il s'agit d'une communication fiable entre deux processus (pouvant être étendue à **TCP/IP**).
- PNG** Portable Network Graphics. Un standard ouvert et libre de licences pour stocker et interchanger des images avec compression sans perte ayant pour but de remplacer le format propriétaire **GIF**.
- point** Dans le cas des réseaux à technologie **FTN**, désigne un utilisateur individuel, équipé d'un programme spécifique qui lui permet d'échanger du courrier (**NetMail** et **EchoMail**) compressé avec le système (appelé le **boss**, ou **node**) dont il est un point.
- POP** Post Office Protocol. Protocole très léger, implémenté sur **TCP/IP**, permettant la lecture de mails en temps réel depuis un serveur de mail. L'intérêt principal de ce protocole est qu'il évite de configurer **SMTP** et un **sendmail** sur des stations avec peu de disque ou pas toujours connectées. Nécessite un serveur distant. Un serveur de type POP-3 est disponible sur **mips.alphanet.ch**.
- POSIX** Une norme définissant les services minimaux devant être offerts par les systèmes modernes. La plupart des versions d'UNIX ainsi que quelques autres systèmes d'exploitation (Windows NT, VAX/VMS récent) offrent ces services minimaux.
- PostScript** Langage de programmation à pile utilisé dans la description de page permettant une indépendance très grande du matériel (par définition vectorielle des formes et caractères à représenter). En général, un fichier PostScript est généré par des programmes de traitement de texte ou de dessin. Le langage est plus ou moins une marque déposée de ADOBE. Cependant, il existe un interprète (convertisseur) **GNU** nommé **Ghostscript**.
- PPP** Point to Point Protocol. Protocole normalisé par le CCITT permettant la connexion de deux ordinateurs ou de deux réseaux. A la fois plus complexe que **SLIP** (car bien plus général), mais parfois plus efficace et en tous cas plus souple néanmoins.
- premier plan** Un processus de premier plan (anglais: *foreground*) est un processus qui a le contrôle du clavier.
- proxy** Un proxy est un serveur qui relaie des requêtes, soit parce qu'il y a un **firewall** (implicite ou explicite, par exemple si l'on veut économiser des adresses) soit parce qu'il est bien configuré (p.ex. il sait ce qu'un **DNS** est), ou encore tout simplement parce qu'il a aussi la fonction de cache (ce qui évite de transférer un document éloigné à chaque accès), cache qui peut être organisé hiérarchiquement (au niveau local, régional, supra-régional, etc).
- regex** Regular Expression. Désigne les motifs de remplacement permettant de retrouver des suites ou des chaînes de caractères.
- registrar** Grossiste s'occupant de vendre les entrées d'un **registry** : p.ex. **NIC/SWITCH** pour **ch**.
- registry** Autorité chargée de gérer les serveurs de noms '.' (**root**), ou la délégation d'un domaine (comme **ch**. ou sous-domaine **co.uk**.). Noter que je ne fais pas la différence entre **top domain** (**ch**., **fr**., **com**.) et le reste. Il n'y en a pas au niveau technique. De plus noter mon utilisation de la notation finale pointée, une bonne idée lorsque l'on veut des enregistrements **DNS** absolus (par opposition à relatifs à la zone courante). On peut traduire par **autorité d'enregistrement**.
- RFC** Request For Comments. Textes officiels contenant des informations sur les protocoles utilisés sur Internet. Beaucoup de ces RFC sont en fait les protocoles eux-mêmes. Un nombre assez important de ces RFC sont consacrés également au point de vue utilisateur, même s'ils sont souvent assez techniques et rébarbatifs. Par exemple, le **RFC-822** définit le format **mail**, et il a été complété dans quelques RFC ultérieurs. Le **RFC-1036** décrit quant-à-lui le mécanisme des **news** et est en cours de révision.
- RNIS** Réseau numérique à intégration de services, ISDN en anglais.
- root** Un utilisateur spécial aux droits étendus réservé à l'administration du système. Par extension, l'administrateur du système lui-même. Désigne également la racine d'un système de fichiers.
- rot-13** Se dit d'une méthode simpliste de chiffrement d'un message, déjà connue des romains, qui consiste à décaler l'alphabet de treize signes. On utilise ce codage primitif surtout dans les **news** afin d'éviter que des personnes non averties soient choquées par un contenu inadapté. La plupart des **newsreaders** permettent un décodage automatique (la commande **d** de **tin**).
- router** Dispositif permettant le passage (le routage) de paquets **TCP/IP** entre deux réseaux. La plupart des ordinateurs peuvent être utilisés comme routers, même s'ils ne sont souvent pas optimisés pour cela.

- RPC** Remote Procedure Call. Il s'agit d'un protocole basé sur **UDP** permettant d'exécuter, de prendre contact et de terminer des procédures distantes, ceci permettant par exemple de répartir la charge d'un programme sur tout un réseau, ou encore d'appeler un service distant. Ce protocole a fait l'objet d'une spécification dans un **RFC**. Une couche supplémentaire nommée **XDR** permet de faire l'abstraction du type de machine. De nombreux protocoles (comme **NFS**) sont basés sur RPC et XDR. On doit la plupart du développement de ces protocoles à SUN MICROSYSTEMS.
- RTFM** Se voit souvent dans les **news**. Cela signifie qu'il faut lire le manuel (RTFM : Read The Fine⁹⁵ Manual).
- satcom** Ordinateur sous UNIX servant anciennement de passerelle mail et news pour ALPHANET. Actuellement n'est plus utilisé par le club. Voir **mips**.
- Secure-RPC** Système dérivé de **Kerberos**, utilisant des méthodes d'authentification et/ou de chiffrement basées sur **DES**, afin d'assurer un minimum de sécurité dans un réseau local.
- secondaire** Un serveur DNS normal qui prend cependant ses informations depuis un autre serveur DNS plutôt que depuis un fichier, pour simplifier. Cf **named-xfer** si on utilise **BIND**.
- serveur** Logiciel s'exécutant souvent en arrière-plan et traitant les requêtes de **clients**. Par extension, la machine exécutant les logiciels serveurs : serveur de mail, de fichiers, etc.
- scam** Publicité mensongère ou message de teneur illégale, comme par exemple les jeux de l'avion ou les **chain-letters**, en général diffusés sous forme de **spam** dans les **news**. (voir section 23.6 page 83)
- script** Un script est un ensemble de commandes **shell** comportant éventuellement des boucles et des tests. Il consiste en un fichier texte commençant généralement par une description de l'interprète nécessaire au fichier, comme `#!/bin/sh, #!/share/bin/tcsh, #!/bin/ksh, #!/bin/csh` ou encore `#!/share/bin/perl`.
- shareware** Logiciel distribué gratuitement pour une période d'essai : après celle-ci, il est nécessaire de payer des frais d'enregistrement (relativement faibles). Très peu courant sous environnement UNIX où la licence **GPL** prévaut (voir section 2.5.2 page 12).
- SLIP** Protocole permettant de raccorder un ordinateur (ou un réseau Ethernet, SLIP, ...) à un autre ordinateur (ou réseau). Il s'agit en fait d'une simple transposition (même pas une encapsulation) du protocole IP. Comme désavantages, citons le fait que les erreurs de transmissions devront être corrigées de bout en bout au lieu de point à point, comme dans le protocole **PPP**. Comme avantages, citons sa simplicité, et la variante **CSLIP**, très efficace en interactif.
- SMB** Server Message Block. Un système de fichiers réseau très commun sous environnements Microsoft. Ce protocole peut utiliser diverses couches de transport : **TCP** port 139 ou **NetBEUI**. Sous UNIX, une implémentation serveur SMB sur TCP est le logiciel libre **Samba**. La partie cliente existe sous deux formes : un client similaire à ftp : **smbclient** et un système de fichiers pour Linux, **smbfs**. Le service de nom utilisé peut être soit le **DNS**, soit un système via **broadcasts** réseau (**NetBIOS**), ou encore un système de nom non hiérarchique : **WINS**, tous supportés par Samba.
- SMTP** Simple Mail Transfer Protocol. Protocole de transfert de mail utilisant le protocole TCP sur Internet. Un serveur, nommé **sendmail**, fonctionne sur **mips.alphanet.ch**. Ce protocole est écrit dans le **RFC-821**, qui a été complété par la suite (**ESMTP**).
- spam** Le fait de poster un article sur un nombre élevé de forums en général sans liens avec le message posté. Il faut bien comprendre que la plupart des gens paient pour leur accès aux news (en argent ou en temps bénévole passé à administrer les serveurs de news). De tels abus sont particulièrement mal vus (surtout si le message a été posté de façon multiple ce qui provoque une transmission multiple des messages). (voir section 23.6 page 83)
- spool** Simultaneous Peripheral Operations Online. Actuellement, ce mot est employé dans un sens plus large. Il signifie qu'un stockage intermédiaire est effectué (par exemple sur un réseau **store-and-forward**).
- SSL** Secure Socket Layer. Protocole permettant de sécuriser des protocoles **TCP** en intercalant une couche de sécurité. Développé initialement par Netscape, Inc., il est désormais disponible librement.
- sticky bit** Se dit d'un mode spécial (voir section 5.9.7 page 25) permettant d'autoriser l'accès en écriture à plusieurs utilisateurs tout en empêchant la destruction mutuelle de fichiers. On utilise ce bit spécial par exemple pour le répertoire `/tmp`.
- store-and-forward** Se dit d'un réseau qui ne fonctionne pas en temps réel, où les messages partant du point A pour le point Z devront transiter par tous les sites intermédiaires en étant à chaque fois stockés de façon temporaire dans des fichiers **spool**. En plus, la communication entre les différents points doit être établie explicitement. Souvent les liaisons sont établies sous réseau X.25 ou commuté (réseau téléphonique normal). Par opposition à un réseau permanent (temps réel) comme Internet où les messages transitent directement et instantanément du site A au site Z.
- suid bit** Set User ID on Execution. Ce bit spécial permet de changer d'utilisateur lorsque l'on tape une commande donnée. Par exemple, le fichier de mots de passe (`/etc/shadow`) n'est pas accessible en lecture aux utilisateurs normaux. La commande `passwd` hérite donc des accès de l'utilisateur système `root` lorsqu'on l'exécute. Il existe également un bit similaire pour le groupe. Le maniement de ces bits n'est normalement pas nécessaire pour l'utilisateur normal (voir section 5.9.7 page 25).
- suid scripts** Il s'agit de scripts avec le bit `s` (visible avec `ls -l`) qui permet, lors de toute exécution, d'obtenir les privilèges de l'utilisateur à qui appartient le fichier. Pour différentes raisons, cette pratique est extrêmement déconseillée, il vaut mieux utiliser ce qu'on appelle un *wrapper* en un langage comme **C** qui aura plus de contrôle sur l'environnement d'exécution.

⁹⁵Il y a d'autres choix possibles : en anglais, on parle du *F word*.

- talk** Protocole TCP qui permet à deux personnes sur Internet d'échanger du texte (style BBS chat), sur le même ordinateur ou entre deux ordinateurs distants, avec la particularité que les deux utilisateurs peuvent agir en même temps.
- TCP** Transmission Control Protocol. Protocole offrant une liaison fiable et **full-duplex** entre deux sites d'**Internet**. Plusieurs connexions TCP peuvent être ouvertes simultanément, chacune d'elle étant assimilable à une sorte de circuit virtuel. La plupart des services interactifs (telnet, ftp, etc.) sont à base de connexions TCP.
- TCP/IP** Transmission Control Protocol / Internet Protocol. Le protocole multi-opération qui est utilisé sur Internet. C'est un protocole très ouvert qui définit les notions d'adresse de machine, de numéro de port (utilisable pour différencier les protocoles de haut niveau comme TELNET, FTP, HTTP etc.) et de datagramme (paquet de donnée). Il offre également deux protocoles de transport (TCP et UDP). De nombreux protocoles de haut niveau ont été implémentés allant du transfert de fichier au courrier électronique privé en passant par la réalisation de WWW. On trouvera une description plus générale des nombreux protocoles TCP/IP dans [12] et [11].
- Teletex** Extension du protocole telex pour l'utilisation avec des terminaux informatiques.
- Teletexte** Système monodirectionnel d'information ayant comme support les lignes inutilisées (en fait utilisées par le retour du faisceau) d'une émission de télévision. On peut voir la zone Teletexte en déréglant légèrement l'appareil de télévision de façon à décaler l'image. Une bande foncée scintillante apparaît alors. Dans celle-ci figurent des codes Teletexte.
- T_EX** Un logiciel de mise en page de documents, utilisé couramment dans les publications scientifiques et techniques, développé par Donald KNUTH et décrit dans [14]. Ce document a été mis en page grâce à une extension de T_EX nommée L^AT_EX.
- Token Ring** Un réseau en anneau principalement utilisé sur systèmes IBM. En français, on parle d'anneau à jetons (le protocole utilisé sur ce réseau nécessite des jetons pour attribuer la parole aux stations, au contraire d'**Ethernet** qui est un réseau **CSMA/CD**.)
- trojan horse** Un programme qui, tel le cheval de Troie, contient du logiciel qui ne remplit pas sa fonction première (p.ex. un virus).
- TTL** Time To Live : compteur situé dans un paquet de données **TCP/IP** décrémenté à chaque passage dans un **router** permettant d'éviter les boucles infinies (et accessoirement permettant le fonctionnement d'utilitaires comme **traceroute**).
- UART** Universal Asynchronous Receive Transmit. Un circuit très courant dans les ordinateurs et les dispositifs de communication permettant de décharger l'unité centrale de la plupart des opérations de bas niveau d'entrées/sorties séries asynchrone. Certains de ces circuit possèdent de la mémoire permettant de stocker des données pendant que le processeur central est occupé. Sans un tel circuit, le processeur central doit se charger lui-même des opérations. On risque alors des débordements de tampons et donc des pertes de données, causant des retransmissions coûteuses en temps, sans parler de la charge du processeur (anglais: *load average*).
- UDP** User Datagram Protocol. Protocole offrant la possibilité d'envoyer sur un site distant des paquets de données au format libre. Le protocole est non connecté et non fiable (donc plus rapide que **TCP**). Certaines applications de haut niveau utilisent ce protocole : c'est le cas de **NFS**.
- UFS** Unix File System. L'ancêtre du système de fichiers UNIX actuel. Heureusement plus trop répandu. Il est célèbre par ses limitations de la longueur des noms de fichier ainsi que par son inefficacité (en terme de stockage et de temps d'accès). Il est avantageusement remplacé par **FFS**.
- Unicode** Jeu standard de caractères permettant de représenter tous les alphabets (y compris le chinois). Il est prévu que ce jeu remplace à long terme le jeu **ISO Latin 1** et les divers jeux internationaux utilisés. Comme caractéristique importante, on notera que ce jeu utilise des caractères de 16 bits (au contraire des 8 bits des caractères usuels). Certains systèmes d'exploitation (p.ex. System V Release 4) offrent déjà des routines de traitement de chaînes de caractères *internationalisées* supportant des caractères multi-octet.
- UNIX** Système d'exploitation développé dès les années 1970 qui intègre le multi-utilisateur, multitâche, et la multi-architecture : un des avantages est bien sûr la portabilité des programmes. Les versions actuelles intègrent toutes le support pour les réseaux, le travail partagé et les systèmes de fenêtrage distribués. Plusieurs variantes de UNIX existent cependant, comme System V, BSD et OSF/1, après une concurrence acharnée et néfaste, on a maintenant une saine émulation entre les différents produits (mais cette concurrence disperse également les efforts de la recherche en technologie UNIX, en particulier face au monolithique et propriétaire **MS-WINDOWS NT**). Des standards existent afin de spécifier les interfaces nécessaires : **POSIX**, X/OPEN, etc.
- URL** Uniform Resource Locator. Un terme historique utilisé au débuts du WWW, devrait être remplacé aujourd'hui par **URI**. On ne vous en voudra pas si vous l'utilisez, cependant.
- URI** Uniform Resource Identifier. Il s'agit de courtes chaînes de caractères identifiant des ressources dans l'espace de nommage du WWW. La syntaxe générale est :

```
protocole://utilisateur:mot-de-passe@domaine:port/sentier/
```

Le / (slash) final n'est nécessaire que pour les répertoires. Les parties utilisateur, mot-de-passe et port sont optionnelles, on peut décrire cela plus précisément (entre crochets les parties optionnelles) :

```
protocole://[utilisateur[:mot-de-passe]]@domaine[:port]/sentier/
```

Parmi les protocoles possibles, citons par exemple **http** et **ftp**. Consultez également le **RFC-2396** serveur WWW du W3 Consortium : <http://www.w3.org/>, la source pour tous les standards liés à WWW.

- Usenet** Réseau coopératif informel recevant les **news**. On écrit souvent **USENET** en référence à l'association d'utilisateurs UNIX **USENIX**.
- USL** UNIX System Laboratories. Un laboratoire de recherche (à l'origine **AT&T**) en technologie UNIX.
- UUCP** UNIX to UNIX Copy Program. Protocole permettant de transmettre des fichiers à distance via une ligne non permanente (donc **store-and-forward**). Permet également d'exécuter des commandes à distance. C'est le protocole actuel qui nous relie à Internet (donc service mail et news seulement). Par extension, UUCP désigne également le réseau comportant les machines utilisant ce protocole. Son avantage principal est bien sûr que la transmission de données a lieu de façon compressée.
- videotex** Système développé par les PTT suisses ayant comme base la norme CEPTTEL qui permet l'interaction avec des banques de données. Son succès très mitigé est dû partiellement aux problèmes techniques mais surtout aux coûts associés. Son asymétrie de transfert ne permet pas également une véritable interactivité. La récente décision de le rentabiliser va probablement le renvoyer aux oubliettes qu'il n'aurait jamais dû quitter.
- VT100** Terminal supportant la norme VT100, permettant l'affichage du jeu de caractères standard ASCII, éventuellement **ISO Latin 1**, ainsi que des caractères graphiques et des modes d'affichage tels que clignotant, souligné, gras, inverse vidéo, et double largeur/hauteur. Ne supporte pas le graphisme point à point. De nombreuses extensions non standard ont été ajoutées rendant parfois incompatibles certaines versions.
- window-manager** Un programme gérant l'apparence et l'interaction (anglais: *look-and-feel*) d'une interface graphique. C'est lui qui décide de l'emplacement des boutons, menus ainsi que des modes d'interaction avec l'utilisateur. Diverses philosophies existent, l'avantage de **X11** est justement d'offrir une indépendance importante de l'environnement : le **window-manager** est un simple client qui gère l'interface de façon indépendante, assurant ainsi la compatibilité.
- WWW** World-Wide Web. Système d'information distribué et multimédia basé sur le protocole **HTTP**, implémentant des liens hypertextes. Plusieurs clients implémentent ce protocole, comme **Konqueror**, **Netscape**, **w3m**, **Mosaic** et **lynx**.
- X11** X11, ou X Window System, ou plus simplement X, est un protocole définissant les notions de client, gestionnaire d'interface et de surface d'affichage virtuelle. Le serveur affiche les différentes fenêtres sur un écran physique. Le gestionnaire d'interface (**window-manager**) autorise l'utilisateur à déplacer des fenêtres. Il gère l'essentiel de la présentation de l'écran et des interactions avec l'utilisateur. Un client peut ainsi afficher ses données indépendamment du logiciel et du matériel, à travers un réseau ou localement.
- Le point intéressant est que tous ces programmes peuvent très bien fonctionner sur des machines distinctes, distantes, de constructeurs différents et offrant des **look-and-feel** spécifiques, comme X11, OpenLook ou OSF/Motif. Cela permet par exemple de centraliser l'affichage de données sur un écran physique, provenant de plusieurs ordinateurs différents, ou de répartir la charge de fonctionnement des clients (p.ex. programme de calcul mathématique), du gestionnaire d'interface (gestion des fenêtres, interaction, etc.), et de l'affichage (contrainte de rapidité, etc.).
- X.25** Protocole normalisé d'échange de paquets. Sa réalisation dans les différents pays se nomme parfois TELEPAC, TRANSPAC, DATEX-P ou encore Tymnet. La taxation est basée en général sur une taxation au volume, parfois agrémentée d'une taxation également à la durée. En Suisse, son coût dépasse de loin celui du réseau commuté pour des transferts de données. De plus, sa vitesse est souvent bien inférieure. Pour le trafic international, la différence est encore plus flagrante. Par contre, en utilisation interactive en trafic national, X.25 peut se justifier.
- X.400** Protocole de messagerie standard (de mauvaises langues disent défini par un comité), généralement moins efficace et bien plus cher (dans tous les sens du terme) que les protocoles dérivés de TCP/IP comme **SMTP**. En plus, l'interface utilisateur n'est pas très adéquate, et la plupart du temps un accès se fera via **X.25**, ce qui n'est pas un avantage en général. X.400 était encore une possibilité du temps où l'accès à Internet était réservé aux non commerciaux et à la recherche. Actuellement il ne fait plus partie des choix pour la connexion à un réseau de courrier électronique en raison de son coût et sa faible interopérabilité.
- XDMCP** Protocole permettant à un serveur X (par exemple un terminal X) de rechercher tous les systèmes sur lesquels un **login** graphique est possible. Un serveur XDMCP tourne sur **mips.alphanet.ch**.
- XDR** eXternal Data Representation. Cette spécification permet d'échanger des données de façon efficace entre différents systèmes d'exploitation, et surtout, entre différents matériels. S'apparente au modèle **ASN-1** de l'**OSI**, à la différence que XDR a été spécifié par SUN MICROSYSTEMS puis versé dans le domaine public. Sa flexibilité et sa simplicité, ainsi que l'existence d'interfaces pour le langage C, l'ont propulsé au rang de standard de fait. On utilise XDR pour implémenter des applications sous **RPC**.
- XFree86** Une implémentation libre de licences (en fait, sous la **GPL**) de **X11** pour systèmes à base de processeurs Intel 80x86. La version courante supporte X11R6 et comporte un serveur et des clients. Elle tourne sous systèmes d'exploitation **Linux**, **NetBSD** et **FreeBSD** ainsi que d'autres non UNIX, comme **OS/2**.
- zone** Equivalent de **répertoire**. Dans les réseaux à technologie **FTN**, désigne une division géographique du monde. Par exemple, la zone 2 est l'Europe occidentale.

Plus d'informations

Le répertoire `/users/admin/info` contient beaucoup de textes d'aide sur UNIX et les réseaux, malheureusement en anglais. Un très bon texte sur Internet est dans [1]. On trouve dans [6] une introduction courte, très abordable à Internet. Malheureusement, ce texte ne comprend pas les évolutions les plus récentes. Le résumé [7] aborde les problèmes sous un angle plus méthodique. Pour une description fidèle du système UNIX, ou du moins de ses principes de base, [3] est un livre de référence car de la main même d'un des concepteurs d'UNIX.

Enfin les utilisateurs d'administration du système sont atteignables via **mail** à `sysadm@alphanet.ch`. Des questions peuvent évidemment être posées via **news** dans les forums locaux. Dans chaque **forum** sont postés chaque mois des résumés des questions les plus courantes avec, dans la plupart des cas, des réponses satisfaisantes. Nous archivons une partie de ces **FAQ** dans `/users/admin/info/faqs` et dans `/users/admin/faq` pour les **FAQ** locales. Le système automatique d'extraction de **FAQs** les stocke dans `/usr/local/archives/autofaq`. Une grande partie du jargon technique UNIX est expliquée dans le **Jargon file** disponible également. N'oubliez pas l'aide en ligne fournie par les commandes `man nom`, où `nom` est une commande et `man -k keyword`, où `keyword` est un mot-clé. De plus, la commande `help` donne des informations générales au débutant. Une expérience intéressante est la tentative de création d'un glossaire francophone et de traductions appropriées de termes anglais à l'**URL** <http://wwli.com/translation/netglos/glossary/french.html>.

Beaucoup d'informations se trouvent dans les forums de **news** **news.announce.newusers** et **news.answers** ainsi que sur les nombreux sites **WWW** et **ftp** (comme `rtfm.mit.edu` ou encore `grasp.insa-lyon.fr`). Un système de recherche par mots-clés de **FAQs** est disponible sur <http://www.pasteur.fr/other/computer/FAQ/>. L'association **FDN** a traduit nombre de documentations et les met à disposition sur ses serveurs **FTP** :

`ftp.fdn.fr:/FDN/Doc` et `http://www.fdn.fr/fdn/doc-misc/`.

Notre serveur **WWW** contient également de la documentation : <http://www.alphanet.ch>.

Bibliographie

Voici quelques titres de livres ou de documents qui peuvent aider à mieux comprendre le système UNIX ou le fonctionnement des réseaux TCP/IP et la sécurité informatique. Les documents disponibles électroniquement sont archivés sur Internet⁹⁶ et également comme RFC⁹⁷.

Références

- [1] B. P. KEHOE, *Zen or the Art of the Internet*, 1992, publié électroniquement sur Internet.
- [2] A. STROHMEIER & P. KIPFER, *UNIX, Une introduction en bref*, PPUR 1993.
- [3] S. BOURNE, *Le système UNIX (traduction)*, 1985, InterEditions.
- [4] B. W. KERNIGHAN & R. PIKE, *L'environnement de programmation UNIX (traduction)*, 1986, InterEditions.
- [5] P. W. ABRAHAMS & B. R. LARSON, *UNIX for the Impatient*, 1992, Addison-Wesley.
- [6] E. KROL, *RFC-1118 : Hitchhiker's guide to the Internet*, 1989, publié électroniquement sur Internet.
- [7] G. MALKIN & A. MARINE, *RFC-1206 : FYI on Questions and Answers - Answers to Commonly asked New Internet User Questions*, 1991, publié électroniquement sur Internet.
- [8] Gene SPAFFORD, *Practical UNIX security*, 1992, O'Reilly.
- [9] Clifford STOLL, *The Cuckoo's Egg — Stalking the Wily Hacker*, ISBN 0-385-24946-2, 1989, Doubleday.
- [10] Severo M. ORNSTEIN, 1980, *Communications of the ACM*, Vol 32 No 6.
- [11] J. DAVIDSON, *An Introduction to TCP/IP*, 1988, Springer-Verlag.
- [12] D. E. CORNER, *Internetworking with TCP/IP*, 1991, Prentice-Hall.
- [13] D. FREY & R. ADAMS, *!@% : : A directory of Electronic Mail Addressing and Networks*, 1989, O'Reilly.
- [14] D. KNUTH, *The TeXbook*, 1989, Addison-Wesley.
- [15] Oliver JONES, *Introduction to the X Window system, updated for X11R4*, 1987, Prentice-Hall.
- [16] Tim O'REILLY, Valerie QUERCIA & Linda LAMB, *X Window system user's guide*, 1989, O'Reilly.
- [17] Olivier RICOU, *Le RéZo — Histoire et guide pour le novice*, 1992, publié électroniquement sur Internet.
- [18] Mark BUTLER, *Internet ... tout de suite (traduction)*, 1994, Dunod.
- [19] John LEVINE & Carol BAROUDI, *Internet pour les nuls (traduction)*, 1994, Sybex.
- [20] Christian HUITEMA, *Le routage dans Internet*, 1994, Eyrolles.
- [21] Ed KROL, *The Whole Internet User's Guide and Catalog*, 1994, O'Reilly.

⁹⁶<http://www-internal.alphanet.ch/archives/info/unix>

⁹⁷La plupart des distributions Linux offrent les RFCs dans un **package** approprié.

- [22] Philippe CHARMAN, *Guide pratique d'Unix et de X Window*, ISBN 2-85428-365-1, 1994, Editions Cépadues.
- [23] W. Richard STEVENS, *TCP/IP Illustrated : volume 1, the protocols*, ISBN 0-201-63346-9, 1994, Addison-Wesley.
- [24] J. REYNOLDS, *RFC-1135 : Helminthiasis of the Internet*, 1989, publié électroniquement sur Internet.
- [25] G. MALKIN & T. LaQuey PARKER, *FYI-18 : Internet Users' Glossary*, 1993, publié électroniquement sur Internet.
- [26] Debra CAMERON & Bill ROSENBLATT, *Learning GNU Emacs*, ISBN 0-937175-84-6, 1991, O'Reilly.
- [27] Dale DOUGHERTY, *sed & awk*, ISBN 0-937175-59-5, 1990, O'Reilly.
- [28] Marc SCHAEFER, *Internet n'est pas une autoroute de l'information*, 1995, publié électroniquement sur Internet, <http://www-internal.alphanet.ch/~schaefer/articles.html>.
- [29] Marc SCHAEFER, *Les protocoles TCP/IP sur Internet*, 1995, publié électroniquement sur Internet, <http://www-internal.alphanet.ch/~schaefer/conferences.html>.
- [30] M. HORTON, *RFC-1036 : Standard for Interchange of USENET Messages*, 1987, publié électroniquement sur Internet.
- [31] *PGP*, 1995, O'Reilly.
- [32] Olaf KIRCH, *The Linux Network Administrators' Guide*, 1994, Linux Documentation Project, publié électroniquement sur Internet.
- [33] Eric DUMAS, *Le Guide du ROOTard pour Linux*, 1995, publié électroniquement sur Internet, <http://www.loria.fr/rootard/>.

GNU Free Documentation License

GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit

reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- * A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- * C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- * D. Preserve all the copyright notices of the Document.
- * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- * H. Include an unaltered copy of this License.
- * I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- * K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- * M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- * N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

- ∧, 21
- (), 21
- , 29, 45
- 0, 50
- font, 63
- geometry, 63
- ., 15, 21, 48
- .*, 21
- .., 15
- .Xdefaults, 54
- .Xmodmap, 54
- .Xresources, 54
- .Z, 27, 108
- .bash_logout, 46
- .bash_profile, 46, 47
- .bashrc, 46, 47
- .cshrc, 29
- .emacs, 33, 39
- .forward, 47
- .fvwmrc, 54
- .gz, 27
- .logout, 17
- .mwmrc, 54
- .netrc, 47
- .newsrsc, 83
- .plan, 46
- .rhosts, 47, 68, 90
- .sh, 47
- .signature, 38, 41, 47
- .tar.gz, 42
- .tgz, 42
- .tvtwmrc, 54
- .twmrc, 54, 55
- .xinitrc, 54
- .xsession, 54
- /, 14
- /dev/fd0, 43
- /etc/X11/Xmodmap, 54
- /etc/X11/Xresources/, 54
- /etc/group, 16
- /etc/passwd, 16, 52
- /etc/resolv.conf, 99, 100
- /etc/services, 89, 100
- /etc/shadow, 114
- /usr/X11R6/lib/X11/app-defaults/, 54
- [], 21
- \$, 18
- \$?, 19
- \$EDITOR, 25
- \$HOME, 18
- &, 23
- &&, 19
- ∖_{TeX}, 92, 96
- TeX, 61, 88, 111, 115
- ~, 18
- 3DES, 67
- 8 bit, 78
- A news, 11
- a2ps, 24
- accents, 43
- ACL, 16, 107
- acroread, 61
- adresse IP, 57, 58, 66, 98, 107
- AFS, 107
- agent, 90
- ainsi, 9
- AIX, 14
- Alex, 86
- alex, 107, 112
- alias, 24, 29
- ALPHANET, 70, 77
- ALPHANET NF, 69, 77, 97, 107
- alt.binaries, 85
- alt.internet.services, 76, 77, 88
- alt.irc, 98
- alt.security.pgp, 45
- alternate, 95
- Amiga, 88, 111
- amiga1, 58
- Aminet, 88, 107
- AmiTCP, 63
- anonymous, 86
- anonymous FTP, 86, 110
- ANSI, 11
- AOL, 69
- apropos, 38, 41
- Aqua, 63
- archie, 85, 86, 88, 107
- ARP, 107
- ARPAnet, 11, 12, 68, 107, 109
- arrière-plan, 22
- article, 81
- AS-225, 63
- ASCII, 14, 107, 111
- ASN-1, 116
- AT, 104
- at, 25, 26
- AT&T, 11, 107, 116
- atomique, 26, 107
- authentification, 107
- autorité d'enregistrement, 113
- awk, 24, 48
- B, 106
- B news, 11
- backbone, 12, 107
- background, 22
- backquote, 49
- backslash, 27, 50
- backtick, 19
- bande passante, 92, 106, 107
- basename, 49
- bash, 18, 46, 48
- batch, 14, 18, 107
- baud, 102
- BBS, 52, 71, 97, 105, 107, 110, 113
- bc, 48
- Bell labs, 68
- bg, 22

- big endian, 107, 108, 112
- BIND, 114
- bindings, 33
- bit, 107, 108
- BITNET, 65, 108
- bits par seconde, 102
- biz, 82, 83
- BNU, 70, 108
- boss, 113
- Bourne shell, 18
- bps, 102
- broadcast, 108, 112, 114
- browser, 84, 95
- BSD, 11, 14, 22, 65, 108, 110, 112
- BSMTP, 108
- buffer, 35, 108
- bug, 68, 108
- byte, 113
- bzip2, 43, 44

- C, 11–13, 18, 48, 50, 66, 108, 114, 116
- C news, 11
- C++, 13
- C-Shell, 18
- cache, 97, 108
- cache de connexions, 97
- calc, 13
- Call For Discussion, 80
- cancel, 45, 82–84, 108
- case-sensitive, 14
- cat, 24, 42
- CCITT, 102
- cd, 15, 16, 18, 25, 51
- cd \$HOME, 16
- cd ~, 16
- cd ~schaefer, 16
- CD-ROM, 88
- CERN, 97
- CERT, 68, 69, 108
- CFD, 80
- chain-letter, 108, 114
- challenge-response key, 107
- channel, 97
- channel op, 97
- charte, 80
- chemin absolu, 15
- chemin relatif, 15
- Chimera, 94
- chmod, 14, 16, 25, 27, 51
- Chorus, 11
- chsh, 47
- cjpeg, 43
- clé révélée, 44
- Class 1, 104
- Class 2, 104
- client, 53, 91, 108, 114
- commandes internes, 23
- comp.bbs.misc, 69
- comp.compression, 43
- comp.dcom.modems, 101
- comp.security.announce, 108
- comp.security.misc, 67, 68
- comp.sys.amiga.announce, 79
- compress, 27, 43, 108
- CompuServe, 71
- contrôle de flux, 104
- cookie, 57, 58
- copy, 60
- copy-and-paste, 60
- core, 27, 108, 109
- correcteur, 102
- cp, 42, 44
- cpio, 30
- crack, 52
- cracker, 75, 85, 108, 110
- crash, 108
- crashmail, 16, 108
- cron, 26
- crontab, 25, 26, 47
- cross-poster, 82, 109
- crypt, 44, 51
- csh, 18, 48
- CSLIP, 98, 99, 109, 114
- CSMA/CD, 109, 115
- CSMA/CR, 109
- CTS, 103
- ctwm, 63

- D, 106
- détacher, 23
- daemon, 11, 95, 109
- DaggeX, 63
- daliclock, 60
- DARPA, 12, 107, 109, 110
- DCE, 11, 102, 103, 105, 109
- dd, 107
- Debian GNU/Linux, 14
- debugger, 13
- debugging, 109
- DEC, 28
- DES, 52, 109, 114
- df, 26
- df ~, 26
- DFS, 107, 109
- diff, 26
- directory, 14
- directory browser, 33
- DISPLAY, 58, 64
- djpeg, 43
- DNET, 105
- DNS, 100, 109, 112–114
- DNS reverse attack, 68
- domain, 91, 100, 109
- dossier, 14
- download, 105
- driver, 11
- DTE, 102–105
- DTMF, 104
- DTR, 99
- du, 26, 42
- duplicates, 85
- DVI, 9, 61, 109

- EBCDIC, 107
- echo-cancellation, 102
- EchoMail, 70, 109, 113

- ed, 27, 30, 42
- EFF, 109
- EFnet, 97, 98
- egrep, 21, 27
- eject, 42
- elm, 71, 73, 78
- elm, mutt, 73
- Emacs, 12, 13, 27, 60, 72, 81, 95
- emacs, 26, 42, 43
- encryption, 44
- ENIAC, 108
- enscript, 24
- env, 18, 19, 46
- enveloppe, 76
- erase, 30
- ESMTP, 114
- Ethernet, 65, 67, 99, 107, 109, 115
- EUnet, 69
- exit, 17, 29
- expansion, 19
- export, 18, 47
- expr, 48
- ext2, 15
- ext3, 107
- extension, 32, 42

- FAQ, 11, 43, 67, 72, 77, 79–83, 88, 101, 109, 117
- FAQs, 77
- fax, 16, 104
- FDN, 117
- FFS, 15, 110, 115
- fg, 22
- FidoNet, 16, 69–71, 75, 77, 107, 108, 110, 112
- field number, 24
- file, 32, 42
- file attachment, 78
- File Transfer Protocol, 85
- find, 27, 42
- finger, 16, 46, 51, 68, 88, 89, 110
- fingerd, 68
- fingerprint, 45
- firewall, 67, 110, 113
- FirstClass, 78
- flag, 108
- FM, 102
- follow-up, 82
- foobar, 110
- foreground, 22, 113
- forum, 11, 29, 68, 80–82, 88, 108, 110, 112, 117
- forward, 38
- fr, 80, 81
- fr.announce.important, 80
- fr.announce.newgroups, 80
- fr.announce.newusers, 83
- fr.biz, 82, 83
- fr.network.modems, 101
- fr.news.8bits, 81
- fr.rec.humour, 82
- fr.test, 82
- fr.usenet.divers, 80
- fr.usenet.groups, 80
- FrameMaker, 60
- FrancoMédia, 69, 70

- FreeBSD, 11, 116
- Frequently Asked Questions, 79
- FSF, 12, 13
- FTN, 70, 80, 109, 110, 112, 113, 116
- FTP, 60, 62, 71, 78, 84–86, 88, 89, 91, 100, 107, 108, 110
- ftp, 44, 47, 85, 86, 99, 107, 114, 117
- FTPMAIL, 110
- ftptool, 85
- full-duplex, 102, 103, 115
- full-feed, 79, 85
- Fundamental, 33
- fvwm, 54, 63
- FYI, 110

- gcc, 12, 13
- gdb, 13
- GfxBase, 63
- Ghostsript, 113
- ghostview, 60, 61
- GIF, 113
- GNU, 10–13, 18, 23, 30, 43, 50, 88, 110, 111, 113
- GNU Bulletin, 13
- GNU Info, 38
- gnu.announce, 13
- gnu.misc.discuss, 13
- gnuchess, 62
- gnuplot, 13, 62
- GNUS, 72
- Gopher, 91, 110
- GPL, 12, 110, 112, 114, 116
- grep, 19–21, 27, 42
- gunzip, 27
- gzip, 27, 30, 42–44, 108

- hacker, 108, 110
- half-duplex, 102, 103
- hardware handshaking, 101
- Hayes, 104
- HDB, 108
- head, 27, 42
- header, 76
- help, 117
- history, 18, 19
- home directory, 16
- home.html, 95
- homedir, 95
- homepage, 51, 92, 95
- HPUX, 14
- HTML, 9, 81, 95–97
- html-helper-mode, 95
- htmlcheck, 95
- HTTP, 67, 92, 97, 116
- httpd, 95
- hub, 113
- HURD, 11, 12
- hypertexte, 110, 111

- id, 16
- IDEA, 44, 67
- IDENT, 100
- image-map, 95
- ImageMagick, 62
- Info, 111

- INN, 11, 84
- inode, 26, 110, 111
- Internet, 9, 11, 12, 14, 44, 52, 65, 66, 68–71, 76, 89, 105, 107, 109–111, 115
- internet, 111
- Internet Protocol, next generation, 67, 69
- interopérabilité, 111
- intr, 30
- intranet, 111
- IP, 111
- IPng, 67, 69
- IPX, 111
- IRC, 89, 97, 98, 111, 112
- irc, 98
- irc op, 97
- ISDN, 102, 106
- ISO 8859-1, 72, 111
- ISO Latin 1, 14, 43, 72, 81, 111, 115, 116
- ISO Latin-1, 78
- ISOC, 10, 111
- ISP, 69
- ITAR, 44
- ITU-T, 102

- Jargon file, 110, 111, 117
- job control, 22
- jobs, 22, 45
- joker, 19–21, 111
- junk mail, 72, 83, 111

- KDE, 54, 60, 95
- Kerberos, 52, 57, 67, 114
- kermit, 30, 111
- kernel, 111
- keyring, 45
- keyserver, 45
- kill, 22, 27
- kill-file, 83, 84
- knews, 84
- Konqueror, 95, 116
- Korn shell, 18
- ksh, 18, 48

- landscape, 24
- latex2html, 9
- lcd, 86
- lecteur, 14
- less, 28
- LGPL, 12
- lha, 43
- liens, 91
- link, 59, 110, 111
- Linux, 10, 11, 14, 18, 22, 48, 57, 63, 65, 88, 112, 116
- Lisp, 12
- little endian, 107, 112
- ln, 111
- load average, 112, 115
- load to local disk, 95
- load-balancer, 66
- log, 30
- logfile, 30, 112
- login, 17, 29, 46, 51, 54, 58, 70, 85, 99, 112, 116
- login shell, 46
- logout, 17
- look-and-feel, 53, 112, 116
- loopback, 66
- lp, 38, 45
- lpq, 45
- lpr, 38, 45
- lprm, 45
- lpspr, 28
- lpstat, 45
- ls, 16, 18, 23, 28, 42, 46
- lynx, 91–93, 116

- météo, 88
- Mach, 11
- Macintosh, 43, 81, 88
- MacOSX, 63
- macro, 28
- MacTCP, 63
- mail, 15, 28, 30, 38, 44, 45, 47, 62, 66, 69–73, 75, 76, 78, 80–84, 89, 107, 110, 112, 113, 117
- mail body, 76
- mailer, 33
- mailing list, 71, 77, 112
- mailing-list, 11, 12, 77
- mailservers, 70, 112
- mailx, 72
- mainframe, 105, 107, 112
- man, 9, 23, 28, 46, 63
- masque, 67
- mdir, 42
- metamail, 78
- mgetty, 16
- Microsoft, 42
- MIME, 42, 43, 73, 78, 81, 97, 112
- mips, 33, 99, 112, 114
- mips.alphanet.ch, 112–114, 116
- mirror, 88
- mkdir, 28, 42
- MNP, 103, 105
- MNP-4, 102
- MNP-5, 103
- modérateur, 80
- mode, 27
- mode vocal, 104
- modem, 16, 30, 66, 99, 101, 104, 106
- moderate, 80
- moderated, 11, 81, 112
- module, 11
- more, 23, 24, 28, 81
- Mosaic, 62, 91–94, 116
- Motif, 54, 94, 112
- motifs, 20
- mount, 43
- Mozilla, 94
- mp, 24
- mpack, 43
- mread, 42
- MS-DOS, 42, 43, 81, 88
- MS-WINDOWS NT, 115
- MTA, 76, 78
- mttools, 42
- MUA, 78
- mud, 112

- multi-threading, 11
- MULTICS, 11
- multiplexe temporel, 106
- munpack, 43
- mutt, 73
- mv, 26, 30, 42, 49
- mwm, 54, 63
- mwrite, 42
- MX forwarder, 75, 112

- named-xfer, 114
- nameserver, 99, 100, 107, 112
- ne, 27, 42
- net, 70, 112
- NetBEUI, 114
- NetBIOS, 114
- NetBSD, 11, 65, 88, 116
- netlag, 98, 112
- NetMail, 112, 113
- netmask, 112
- netpbm, 42, 43
- Netscape, 60, 62, 85, 91, 94, 95, 116
- netstat, 91
- netsurfer, 111
- news, 11, 15, 28, 47, 66–72, 75, 77–85, 88, 91, 98, 101, 108–110, 112–114, 116, 117
- news.announce.newgroups, 80
- news.announce.newusers, 11, 83, 117
- news.answers, 80, 117
- newsgroups, 79, 83
- newsreader, 113
- NF, 24
- NFS, 11, 44, 57, 86, 107, 109, 112, 114, 115
- nic.funet.fi, 85
- nice, 28
- NIS, 16, 51, 113
- NIXPUB, 69
- NNRP, 79
- NNTP, 11, 79, 83–85, 89, 113
- nntpd, 113
- NNTPSERVER, 83
- node, 66, 70, 113
- nodelist, 108
- noeud, 113
- nroff, 28
- NSFnet, 12
- nslookup, 100
- NUL, 50
- null modem, 103
- Numéris, 106
- nuucp, 70

- Objective-C, 13
- oclock, 60
- octale, 37
- octet, 107, 113
- offline mailreader, 72
- offline newsreader, 83
- olvwm, 63
- olwm, 63
- on-the-fly, 104
- online, 9
- online service, 69

- OpenSSH, 90
- OS/2, 42, 116
- OSF, 11
- OSF/1, 14, 22, 28
- OSI, 112, 116

- pack, 108
- package, 14, 117
- packet-switched, 66
- paquet, 14
- passwd, 25, 28, 51
- paste, 60
- patch, 26
- PATH, 26
- path, 14, 47
- pattern, 20
- pcnfs, 63
- PDDIAL, 69
- PDP-11, 11
- perl, 48
- PGP, 44, 51, 57, 68, 71, 72, 75, 85, 92, 112, 113
- pgp, 44
- pico, 28, 72
- PINE, 28, 72
- ping, 90, 99, 100
- ping delay, 98
- pipe, 19, 22, 28, 43, 48, 58, 113
- PlotMTV, 62
- PNG, 113
- point, 113
- POP, 72, 90, 113
- POSIX, 11, 18, 22, 26, 48, 112, 113, 115
- PostScript, 9, 24, 28, 109, 113
- PPP, 16, 66, 90, 98, 99, 101, 105, 113, 114
- premier plan, 22, 113
- Pretty Good Privacy, 44
- private, 16
- procmail, 73, 74
- prompt, 16
- proxy, 67, 110, 113
- ps, 22
- psnup, 28
- pstops, 28
- public, 16
- public-key cryptography, 44
- pwd, 16, 18, 29

- QAM, 102
- quota, 29

- répertoire, 14, 15, 116
- répertoire courant, 15
- répertoire de travail, 15
- répertoire père, 15
- répertoire par défaut, 16, 51
- racine, 15
- random sequence number guessing, 69
- RARP, 107
- rcp, 44, 90
- RCS, 26
- realtime, 66
- rec.humor.funny, 29
- recode, 43

- redirection, 24
- refresh, 41
- regexp, 18–21, 27, 34, 113
- registrar, 113
- registry, 113
- remote login, 12
- renice, 28
- repository, 45
- resolver, 100
- resolving, 100
- resources, 54, 57, 63
- reverse mapping, 100
- RFC, 12, 67, 110, 112–114
- RFC-1036, 11, 79, 82, 113
- RFC-1700, 89
- RFC-2396, 115
- RFC-821, 114
- RFC-822, 113
- RIP, 105
- rlogin, 90, 98
- rm, 29, 42
- RMAIL, 38
- rmail, 38, 70
- rmdir, 29, 42
- rnews, 44, 70
- RNIS, 66, 106, 109, 113
- root, 14, 26, 28, 58, 90, 113
- rot-13, 82, 83, 113
- router, 66, 68, 113, 115
- RPC, 109, 112, 114, 116
- RSA, 44, 67, 90
- rsh, 58, 90
- rsmt, 70
- RTFM, 114
- RTS, 103
- RTS/CTS, 101, 104, 105
- RXD, 104
- rz, 29, 44

- S/Key, 67
- Samba, 114
- satcom, 114
- satcom.alphanet.ch, 107
- scalability, 109
- scam, 82, 83, 114
- SCCS, 26
- Scheme, 18
- scp, 44, 90
- screen, 21, 29, 52
- script, 26, 59, 107, 114
- scripts, 11, 18, 19, 47
- scsh, 18
- SCSI, 109
- secondaire, 114
- secure telnet/telnetd, 67
- Secure-RPC, 52, 57, 114
- SecureRPC, 67
- sed, 20, 29
- sendmail, 68, 78, 114
- sequence number guessing, 68
- serveur, 53, 108, 112, 114
- set, 18, 19, 46
- set user/group id on execution, 25
- setenv, 19
- SGML, 92
- sh, 18, 26, 42, 48, 49
- shadow passwords, 52
- shar, 42
- shared libraries, 11, 64
- shareware, 61, 114
- shell, 11, 15–17, 19–24, 26, 27, 42, 46–48, 72, 83, 86, 90, 98, 112, 114
- site, 113
- slash, 29, 50
- SLIP, 16, 54, 66, 85, 90, 98, 99, 105, 114
- SLIRP, 98
- slrn, 79
- SMB, 114
- smbclient, 114
- smbfs, 114
- smiley, 72
- SMTP, 67, 113, 114, 116
- snooper, 68, 75
- snooping, 69
- socket, 14
- socket UNIX, 54, 59
- Solaris, 22, 28
- son of 1036, 11
- source address check, 68, 69, 75, 85
- sox, 42, 43
- spam, 82, 83, 114
- split, 42
- sploin, 42
- spool, 15, 70, 72, 83, 114
- SSH, 16, 59, 90
- ssh, 58, 67, 90
- ssh-agent, 90
- sshd, 59, 90
- SSL, 93, 114
- START, 101, 102, 105
- startx, 57
- sticky bit, 25, 114
- STOP, 101, 102, 105
- stop, 22
- store-and-forward, 65, 66, 69, 70, 75, 107, 108, 110, 112, 114, 116
- stty, 30
- su, 58
- su - utilisateur, 47
- subnet, 66, 67
- suid bit, 114
- suid scripts, 48, 114
- summary, 82
- Swissnet, 106
- SWITCH, 88
- switched, 65
- système de fichiers, 80
- system calls, 111
- System V, 14
- System V Release 4, 110
- sz, 30, 44

- T1, 106
- TA, 106
- TAB, 84
- tag, 96

- tail, 30, 42
- talk, 89, 115
- tar, 30, 42–44
- TAS, 107
- Taylor UUCP, 70
- TCP, 54, 59, 67, 75, 89, 91, 92, 97, 99, 110, 114, 115
- TCP/IP, 11, 12, 54, 58, 63, 67, 91, 92, 98, 105, 107, 109, 110, 113, 115
- tcsh, 17–19, 21, 46, 48
- teletex, 115
- teletexte, 91, 115
- teletype, 16
- TELNET, 16, 67, 89, 100
- telnet, 51, 64, 68, 85, 89, 90, 98, 100, 107
- term, 105
- TeX, 109
- Text, 33
- thread, 11, 84
- threaded interactive newsreader, 83
- threading, 83, 84
- time, 17
- tin, 79, 82–84, 108, 109, 113
- to resume, 29
- Token Ring, 67, 109, 115
- toolkit, 112
- top, 22
- top domain, 113
- tr, 43
- trace, 68
- traceroute, 90, 115
- transceiver, 99
- trash.net, 69
- trojan horse, 52, 115
- Trumpet Winsock, 64
- TTL, 98, 115
- tty, 14, 16
- tunnel, 59
- turn-around delay, 102
- tutorial, 33
- tvwm, 54, 63
- twm, 54, 63
- TXD, 104

- UART, 115
- UDP, 67, 114, 115
- uemacs, 27, 42
- UFS, 15, 110, 115
- umask, 25
- unalias, 18
- Undernet, 97
- undo, 41
- Unicode, 115
- UNICOS, 14
- UNIX, 11, 86, 115
- unresolving, 100
- unshar, 42
- upload, 105
- uptime, 112
- URI, 115
- URL, 92, 98, 115, 117
- USENET, 79, 112
- Usenet, 11, 12, 65, 80, 83, 88, 116
- user-space, 111

- USL, 11, 116
- uucico, 70
- UUCP, 11, 14, 16, 65, 66, 70, 71, 73, 75, 76, 84, 108, 112, 116
- uucp, 70
- uudecode, 30, 43
- uuencode, 30, 43, 78
- uustat, 70
- uux, 70
- uuxqt, 70

- V25bis, 104
- V32bis, 105
- V42, 103
- V42bis, 103, 105
- variables d'environnement, 17–19
- vi, 30, 42, 43, 81
- Videotex, 95
- videotex, 69, 92, 116
- Virtual Private Network, 90
- voicemail, 104
- VPN, 90
- VT100, 16, 19, 26–29, 33, 60, 72, 98, 105, 116
- vulcan, 14
- Vxx, 102

- w, 89
- W3, 97
- W3 Consortium, 96
- w3m, 93, 116
- wc, 32, 42
- well-known access point, 89
- who, 89
- whoami, 16
- whois, 91
- wildcard, 18
- window-manager, 53, 54, 63, 116
- WINS, 114
- working directory, 15
- worm, 68
- write, 89
- WWW, 12, 62, 67, 69, 84, 85, 88, 95–97, 107, 108, 116, 117

- X.25, 106, 110, 116
- X.400, 71, 116
- X/Open, 11, 14
- X11, 9, 11, 52, 54, 60, 63, 64, 72, 73, 83–86, 92, 105, 107, 116
- X11.log, 64
- XAPPLRESDIR, 54
- xarchie, 62, 85
- xargs, 50
- xauth, 58, 68
- XAUTHORITY, 58
- xbiff, 62, 73
- xbill, 63
- xboard, 62
- xclock, 60
- xdm, 54, 58
- XDMCP, 54, 116
- xdoom, 63
- XDR, 114, 116
- xdvi, 61

xedit, 60
XEmacs, 33
xev, 57
xfig, 62
xfilemanager, 60
xfm, 60
XFree86, 63, 64, 116
xfs, 107
xhost, 58, 68
xinit, 54, 57
xlock, 52, 60
xlsclients, 57
xmh, 62, 73
XModem, 30, 105
xmodmap, 57
xmorph, 62
XON/XOFF, 101
XOVER, 84
xpaint, 62
xrdb, 57
xrisk, 63
xterm, 60, 92, 93
xv, 61, 62

YModem, 105
Yinternet.org, 69
yppasswd, 51

zip, 43
ZModem, 30, 92, 105, 111
zone, 116
zsh, 18