Courte information sur le système UNIX et les réseaux téléinformatiques.

version 2.105 du 21 février 1995

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 <*pjcagnar@di.epfl.ch*> pour sa relecture attentive ainsi qu'à Norbert EBEL <*ebel@lcao.epfl.ch*> et Isabelle SAINT-SAËNS <*iss@deneb.inrets.fr*> pour leurs nombreuses corrections et suggestions.

Résumé

Ce guide, composé de deux parties, traite essentiellement du système UNIX et des réseaux de communication sous UNIX. 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 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 de profiter aux mieux des ressources réseaux, ne saurait remplacer un bon livre de référence UNIX ni un livre traitant en profondeur des réseaux informatiques. Ce guide a été rédigé dans l'espoir qu'il serait utile: il est rendu disponible tel quel et l'auteur décline toute responsabilité quant à son contenu.

Droits d'auteur

Table des matières

Table des figures

1 Introduction

1.1 Buts de ce résumé

Ce résumé traite du système 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 `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 _ (un caractère blanc souligné) lorsqu'une confusion est possible. En général on se rappellera que toute commande doit être séparée de ses arguments par des espaces.

Quand un mot se trouve dans le glossaire ou dans la table des matières, il s'écrit comme **dans cet autre exemple**. Ce mot se retrouve automatiquement 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 les touches *CTRL* et *X* du clavier.

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

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 catalogues de manuel, de plus, la commande `man -k` permet de connaître ce numéro.

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

1.3 Structure

Ce document est composé de deux parties, la première offrant une vue d'ensemble du système UNIX et la deuxième montrant 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 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**. Une version **DVI** est également disponible. Enfin, une version **HTML** est créée automatiquement par le logiciel `latex2html`.

2 Historique

2.1 Introduction

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

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

2.2 Origine du système UNIX

UNIX 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 éléphantesque) système d'exploitation, qui était en développement à cette époque. On ne niera pas le fait que presque tous les mécanismes d'UNIX s'en sont inspirés.

Les premières versions ne furent pas diffusées en dehors des laboratoires Bell. La V4, contrairement aux précédentes, écrites en assembleur PDP-11, fut écrit 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** d'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 DEC pour **DCE** sous **OSF/1**). Des comités de normalisation définissent les bases de standards UNIX (**POSIX**, etc). On intègre à UNIX les protocoles **Internet** ainsi que de nombreuses améliorations de l'interface utilisateur (**X-11**). En 1993, Novell rachète la marque commerciale et la cède au groupe indépendant de normalisation **X/Open**.

Comment peut-on évaluer le futur d'UNIX? Les développements récents des systèmes parallèles ont amené des versions proposant les **thread**s (sorte de processus léger), ou même le **multi-threading** au sein du noyau, avec le support des architectures multiprocesseur. Des noyaux légers comme **Mach** ou **Chorus** sont aussi proposés. Enfin, la configurabilité de la machine (noyaux configurables sans recompilation[1], grâce aux **daemon**s externes et aux mécanismes de STREAMS) est un des buts des systèmes récents.

Les trois versions principales d'UNIX sont actuellement System V Release 4.2, BSD 4.4 et OSF/1 3.0. De très nombreuses implémentations empruntant à une ou plusieurs de ces versions ont vu le jour, la plupart redistribuables librement.

2.3 Origine d'Usenet et d'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**.

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 **forum**s modérés (anglais: *moderated*), la structure hiérarchique de nommage et de nouveaux mécanismes de contrôle, en s'inspirant des listes de distribution 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 d'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, dès le début a été soutenu 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* sur le réseau **ARPAnet** pour des raisons diverses, notamment le temps 'perdu'.

---

[1] Le futur système d'exploitation **HURD** du projet **GNU** offre, entre autres, cette possibilité *(voir section 2.5.3 page 9)*.

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 comme but de maintenir les communications entre les ordinateurs militaires en cas de catastrophe majeure[2] (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.

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 de la recherche. On développe le protocole **TCP/IP**. Usenet est de plus en plus transféré à travers Internet. De nouveaux protocoles sont développés. Le réseau s'ouvre aux commerciaux, des **backbone**s entiers sont privatisés. En 1990, le CERN développe le système hypertexte **WWW**.

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 à 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 pratique[3], car il existe toujours un moyen (légal!) 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 d'UNIX, mais ce n'est pas UNIX (d'où le nom 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 taper 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 système d'exploitation en 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 24)*,

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 diponibles 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,

---

[2] Il 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.

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

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 peuvent remplacer des commandes telles que `tar`, `awk`, `yacc`, `lex`, `find`, `ls`, `make`, etc.

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 15.4 page 51)*.

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

`Free Software Foundation, Inc.`
`675 Mass Ave, Cambridge, MA 02139`
`USA`

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 réseau). Actuellement, presque tous les constructeurs de machines l'utilisent, ou du moins utilisent une variante de UNIX, comme par exemple **BSD**, **OSF/1**, **AIX**, ou **System V**. L'avantage de UNIX par rapport aux autres systèmes d'exploitation, outre le fait qu'il est très répandu, est sa puissance alliée à sa simplicité. 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 (p. ex. **Linux**) jusqu'au très gros système (**Cray UNICOS**). La plupart du temps, une simple recompilation suffit. Parfois il faut modifier la source. Il faut également noter que de plus en plus de versions du système UNIX lui-même sont disponibles sans aucune licence, voire gratuitement.

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

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 restreindre[4] l'accès à des fichiers pour un groupe de travail ou d'intérêt: par exemple sur **mips**, les membres du club possèdent le groupe members et peuvent donc accéder aux fichiers présents dans le catalogue `/users/admin/admin` concernant le club, et eux seulement. Le système multi-tâche 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.

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 tel quel.

Pour interrompre de façon quelque peu sauvage tout programme il suffit de taper *CTRL–C*[5].

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

3.2.3 Arborescence

Le système UNIX définit les systèmes de fichiers depuis une racine unique, appelée **root** et notée `/`. Depuis cette racine il est possible de se déplacer dans les catalogues système ou utilisateurs pour autant que les permissions associées le permettent. Ceux-ci forment une réelle arborescence[6], les feuilles étant les fichiers et les noeuds des **catalogues** (**répertoire**, ou **dossier**; anglais: *directory*). La figure 1 en page 12 donne un exemple d'arborescence et de nommage de sentiers (anglais: *path*).

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

`/dev` périphériques système,

`/etc` administration système,

`/bin` commandes et utilitaires,

`/usr/bin` autre catalogue de commandes,

`/tmp` catalogue de fichiers temporaires,

`/users` catalogues utilisateurs,

`/share` exécutables, sources, archives, bibliothèques, documentations et manuels partagés installés localement,

`/aminet` logiciels Aminet,

---

[4] `chmod` *(voir section 5.6.9 page 19).*

[5] On peut aussi utiliser la séquence *CTRL–Z* puis `kill %%` *(voir section 5.4 page 16).* Cela est redéfinissable *(voir section 5.6.12 page 20).*

[6] En informatique, les arbres ont la fâcheuse habitude de pousser vers le bas.

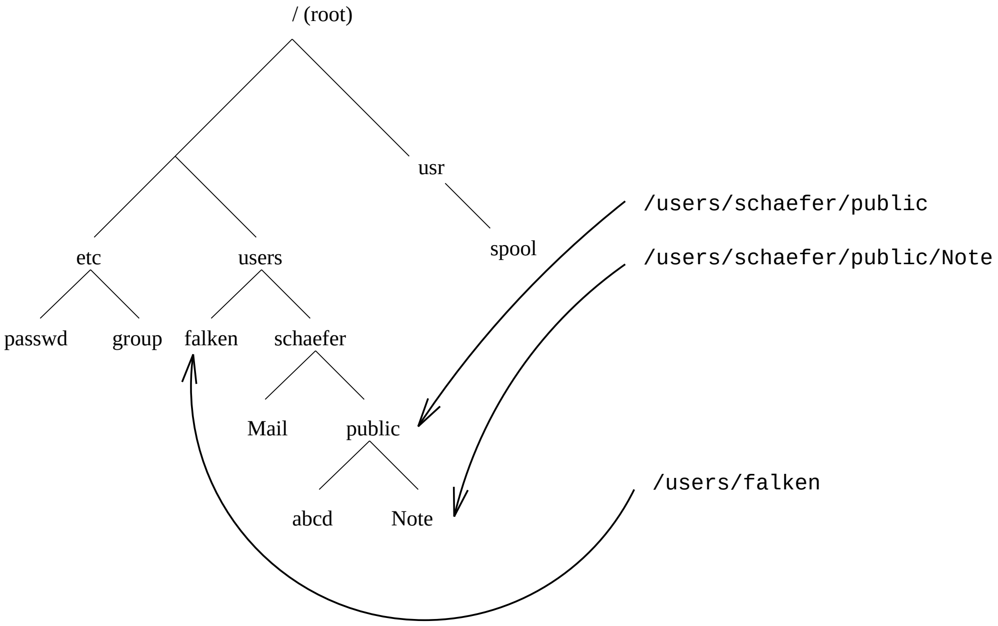


Fig. 1 - Arborescence du système de fichiers

`/scratch` disque supplémentaire non permanent,

`/news` **spool** des **news**.

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 usuels. Par exemple, les noms de fichiers sont limités à 255 caractères.

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 catalogue ou fichier désiré. Par exemple `/users/schaefer/Mail/un_fichier` est un chemin absolu jusqu'au fichier `un_fichier` du catalogue `Mail`, lui-même contenu dans le sous-catalogue `schaefer` du catalogue `users`. Un **chemin relatif** commence par un nom de fichier ou de catalogue, ou éventuellement par `.` ou `..` qui représentent respectivement le **catalogue courant** (anglais: *working directory*, **catalogue de travail**) et le **catalogue père** (celui qui se trouve immédiatement au dessus du catalogue 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 catalogue courant soit `/users/schaefer`:

`Mail/un_fichier` ou `./Mail/un_fichier` ou encore `../schaefer/Mail/un_fichier`, ou même, pour compliquer, `../schaefer/././../schaefer/Mail/../Mail/un_fichier`.

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 **catalogue courant**, on utilise la commande `cd` avec comme paramètre le chemin (relatif ou absolu) du **catalogue** où l'on veut se rendre. Pour revenir en arrière d'un sous-catalogue dans la hiérarchie, on peut utiliser `cd␣ ..` (où le caractère ␣ représente la touche *ESPACE* et non pas le caractère souligné).

Le **catalogue par défaut** (anglais: *home directory*) est le catalogue où l'on se trouve à la connexion. On peut y retourner en entrant `cd`.

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 catalogues en majuscule, comme dans Mail, News, Projets. On peut aussi utiliser la forme typée de l'utilitaire `ls` avec `ls -F`.

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 catalogue par défaut, par exemple /users/schaefer

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

username nom et prénom

office par exemple le lieu d'appel

On peut obtenir ces informations[7] 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 main[8]. Pour vous-même, vous pouvez utiliser `id` et `whoami`. Chaque fichier, ou catalogue, 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 5.6.9 page 19)*.

4 Accès à notre système UNIX mips

4.1 Appel du système

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 9 page 36)*),

Le modem répond au (038) 41.40.81, 24h sur 24, aux vitesses suivantes:

- V21, 300 bps,
- V22, 1200 bps,
- V22bis, 2400 bps,
- V23, 1200/75,
- V32, 9600 bps,
- V32bis, 14400 bps,

De plus, il supporte les protocoles V42, V42bis et MNP 1 à 5. Actuellement, le port série est fixé à 19200 bps (à l'avenir il est prévu de le faire passer à 38400 bps). Le système supporte les logins standards UNIX (interactifs (shells), non interactifs (**UUCP**) et multi-sessions (**SLIP**)) ainsi que **FidoNet CRASHMAIL**.

4.2 Le login

Le système décroche, et quelques instants plus tard, la connexion est établie. Le système affiche ensuite un message de bienvenue et un prompt Login:. 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 trompez[9], 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 10.2 page 40)*.

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

---

[7] sauf le groupe

[8] seulement possible sans **NIS**

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

4.3 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*[10] 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 sessions peut être dangereux du point de vue de la sécurité, même si la détection du raccrochage ne pose aucun problème en général au système **mips**.

Exemple: L'utilisateur falken quitte le système.

```
<mips> 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 statistiques 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.

5 L'interface de commande (shell tcsh)

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. Seul le shell **tcsh** sera présenté, car c'est un des plus pratiques installés sur **mips**.

5.1 Notions principales

Le shell définit les notions suivantes:

catalogue par défaut il s'agit du catalogue appartenant à l'utilisateur, accessible en tapant `cd`. Il correspond à la variable d'environnement **$HOME** et à l'abréviation ~/

catalogue courant le catalogue courant correspond à un dossier ouvert sur votre bureau. Pour connaître le catalogue courant, il faut taper `pwd`. Pour changer de catalogue courant, on utilise `cd`.

variables Les **variables d'environnement** contiennent des informations comme le type de terminal et sont modifiables avec `env`, `set` et `setenv`.

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.

expressions remplaçantes 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 /users/loginname, où loginname est le nom d'utilisateur courant. De même `~user` sera remplacé par le catalogue par défaut de l'utilisateur user, bien souvent `/users/user`. On appelle parfois les expressions remplaçantes des expressions régulières ou **regexp**.

complétion La touche *TAB* permet de compléter le nom d'un fichier en cours de saisie s'il n'y a pas de possibilités de confusion et que le catalogue 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 \.

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 **batch**es.

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 tube) est aussi intéressante, permettant de connecter la sortie standard d'une commande à l'entrée d'une autre commande.

correction Le shell peut corriger de façon limitée la syntaxe des commandes et éventuellement des arguments. Il présente alors une possibilité, et demande `y/n/e`[11]. Pour accepter, il faut taper *y*, sans appuyer sur *RETURN*.

commandes multiples On peut indiquer des séquences de commandes avec le séparateur *;* ou à l'aide de **scripts**.

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

---

[10] *(voir section 3.2.2 page 11).*

[11] Pour l'activer, si ce n'est pas déjà fait, utiliser `set correct=cmd` *(voir section 9 page 36).*

expansion L'expansion des arguments des commandes, particulièrement en présence d'expressions régulières (**regexp**) 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 section 5.3.2 page 16)*.

5.2 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 catalogue 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.

Les **variables d'environnement** commencent par $, et permettent soit de stocker des valeurs intermédiaires dans des **scripts**, soit par exemple de 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 9 page 36)*.

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

5.3 Expressions régulières

5.3.1 Introduction

Les expressions régulières, aussi appelées expressions remplaçantes, ou encore **regexp**, servent à extraire parmi une liste d'objets ceux qui correspondent à certains critères ou **motifs** (anglais: *pattern*, patrons). Par exemple, on peut effectuer une recherche dans un catalogue, 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.

Les **joker**s que l'on retrouve sous beaucoup de systèmes d'exploitation, comme * ou encore ? forment un sous ensemble des possibilités des expressions régulières d'UNIX.

5.3.2 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`, les motifs suivants provoqueront un succès:

- `sau`
- `par dess`
- `e`

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

- `saz`
- `pardess`
- `34`

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 **joker**s 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 débuter 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* `abcd`*, 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 "∧d2/local|∧d3/divers|∧var/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.

5.4 Multi-niveau (JOB CONTROL)

Note: Les facilités exposées dans toute cette section (y compris ses sous-sections) ne sont disponibles que sur le système **mips**. 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 5.6.20 page 22)*.

Les facilités de gestion des processus et des jobs du **shell** par défaut (tcsh) sont assez avancées, c'est pour cela que les utilisateurs novices ne devraient pas les utiliser, du moins pas dans un premier temps.

Le système de JOB CONTROL, en français contrôle de travaux multiples, permet le lancement de plusieurs travaux en parallèle, de les reprendre tour à tour comme travail de premier plan et de mettre un travail en 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**).

5.4.1 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-catalogues du catalogue courant, et un second recevant la liste du premier (via le **pipe**) 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 5.4.3 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 reférera 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 1. Des valeurs spéciales sont définies, par exemple `kill %%` tue le processus manipulé ou créé en dernier.

5.4.2 Obtenir la liste des processus

Il existe deux commandes principales, 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.

`ps -ef` liste de tous les processus du système.

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

5.4.3 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.

5.4.4 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** %%.

5.4.5 Détruire un processus

Pour ce faire, on utilise la commande **kill** 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 5.4.2.

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

`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

5.4.6 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 &
```

5.5 Syntaxe

La syntaxe générale des commandes est la suivante:

```
commande -options paramètre ... paramètre
```

Il est important de remarquer qu'un espace sépare chacun des paramètres et les paramètres de la commande elle-même. Par contre, on remarquera également que certains paramètres peuvent commencer par un tiret, noté -.

5.6 Principales commandes

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 du guide *(voir section II page 42)*. Pour une aide plus complète, utilisez la commande **man**, comme expliqué ci-après.

5.6.1 `man`, manuels en-lignes

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 standard[12] (commandes utilisateurs 1, administrateur 1M, appels systèmes 2, bibliothèques 3, format de fichiers 4, services réseau 7). Par exemple, pour obtenir des informations sur la commande `ls`[13], on peut écrire `man 1 ls` 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.

---

[12] Certains 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.

[13] Les documentations UNIX représentent souvent une commande suivie de sa section entre parenthèses comme dans ls(1).

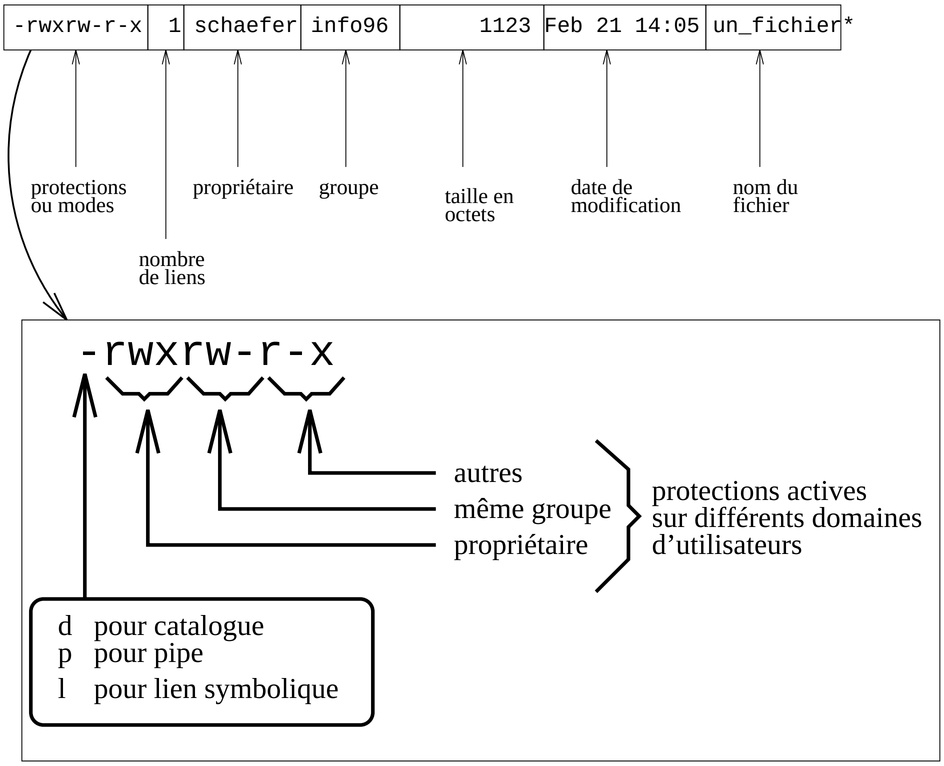


FIG. 2 - Détail d'une sortie de `ls -laF`

5.6.2 `passwd`, changement du mot de passe

Cette commande permet de changer de mot de passe. 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.

5.6.3 `cd`, changement de catalogue

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

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

`cd /` revenir à la racine.

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

5.6.4 `pwd`, affichage du catalogue courant

Elle permet d'afficher le catalogue courant.

5.6.5 `ls`, liste d'un catalogue

Elle permet de lister le contenu d'un catalogue. 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 (fichier, catalogue/, lien@, exécutable*).

La figure 2 en page 18 détaille la sortie de la commande `ls -laF`.

5.6.6 `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 5.6.7 page 19)* permet quant à elle un affichage page par page. Une utilisation intéressante de cette commande est pour concaténer des fichiers. Par exemple, `cat > tous a b c d` crée un fichier `tous` contenant la concaténation (mise bout à bout) des fichiers `a` à `d`, grâce à une **redirection**.

5.6.7 `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*.

5.6.8 `rm`, destruction d'un fichier

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

`rm -i fichier ... fichier` 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 d'avoir `rm` remplacé 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.

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. Il faut noter que vouloir récupérer un fichier préalablement effacé à l'aide des structures de catalogues est une opération la plupart du temps vouée à l'échec.

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`.

5.6.9 `chmod`, changement des protections d'un fichier

Elle permet de modifier les attributs, ou protections, d'un fichier ou d'un catalogue, seulement si vous en êtes le créateur. Elle permet également de protéger vos données *(voir section 10.3.2 page 41)*.

Son format est le suivant: chmod ± quoi fichiers

Par exemple:

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

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

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

Notez que la création de catalogues ou de fichiers se fait avec un mode par défaut qui dépend de la commande `umask` *(voir section 9.1.2 page 37)*.

5.6.10 `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

5.6.11 `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.

5.6.12 `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 (défaut: *CTRL–C*), ou la touche d'effacement (défaut: *BACKSPACE*) *(voir section 9 page 36)*.

5.6.13 `du`, usage disque

Cette commande permet de connaître l'espace disque utilisé par un sous-catalogue. L'unité est en demi-blocs (soit 512 caractères). Un paramètre très souvent utilisé est `-s *`, qui permet d'obtenir la taille en demi-blocs de tous les objets du catalogue courant et récursivement.

5.6.14 `df`, disponibilité des systèmes de fichiers

Celle-ci permet de connaître les disques disponibles, leur capacité et leur charge.

5.6.15 `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 mode insertion

a mode ajout/insertion

d puis d destruction d'une ligne

x destruction du caractère courant

$ se positionner à la fin de la ligne

& répéter la substitution précédente.

D détruire jusqu'à la fin de la ligne

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

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 `d5` détruit cinq lignes depuis la position courante). En voici un échantillon:

`j` concaténation de lignes

`d` destruction de lignes

`x` sauver et quitter

`w` sauver

`q` quitter sans sauver

`n` passer au fichier suivant si plusieurs fichiers

`$` se déplacer à la fin du fichier

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

`set nu` numéroter 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.

Pour montrer quelques exemples de commandes complexes (il faut toujours ajouter un *:* avant et un *RETURN* après).

`1,$s/foo/bar/g` Remplace toutes les occurences de foo par bar

`1,$s/foo/bar/c` Remplace toutes les occurences de foo par bar, avec confirmation à chacune (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 *.*[]* ont des significations particulières, il faut les précéder du caractère d'échappement \ *(voir section 5.3 page 15).*

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.

Mode d'insertion Le mode d'insertion n'autorise ni déplacement ni commande, et doit être terminé par un *ESCAPE*.

5.6.16 `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.

5.6.17 `uemacs`, éditeur pleine page simplifié

Micro-emacs est un sous-ensemble de l'éditeur **emacs**.

5.6.18 `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.

5.6.19 `pico`, éditeur pleine page simplifié

Cet éditeur est très simple à utiliser et malgré cela relativement puissant.

5.6.20 `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)

? 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.

w liste des fenêtres.

m contrôle d'activité allumé/éteint.

5.6.21 `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 expérimentés. Une commande similaire, permettant cette fois de retarder l'exécution de travaux (par exemple pendant la nuit ou le week-end) est la commande `at`.

5.6.22 `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-catalogues du catalogue courant, y compris ce dernier:

```
find . -name "*t*" -print
```

5.6.23 `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, respectivement:

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

Dans ce dernier cas, on interrompt la commande avec *CTRL–C*.

5.6.24 `head`, affichage des premières lignes d'un fichier

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

5.6.25 `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

5.6.26 `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.

5.6.27 `mkdir`, création d'un catalogue

Crée des catalogues.

5.6.28 `alias`, définition de synonymes de commandes

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

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

5.6.29 `sz`, envoi d'un fichier via ZModem

Elle permet d'envoyer des fichiers via le protocole **ZModem**. 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. Dans certains cas, il faut renommer le fichier avant l'émission avec **mv**, pour que le nom de fichier soit compatible avec le système de fichiers distant.

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

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

5.6.31 `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` décompresse fichier.gz en fichier

5.6.32 `uuencode`, encodage de fichiers binaires pour transmission via courrier électronique

Grâce à cette commande, on peut envoyer des fichiers binaires par courrier électronique *(voir section 14.6 page 49)*.

`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 dans un fichier.

5.6.33 `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 fichier du catalogue courant et de ses sous-catalogues, compresse cette archive avec **gzip** (l'argument -9 sélectionne la meilleure (mais plus lente) compression) et stocke le résultat dans `truc.tar.gz`.

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

`gzip -d < truc.tar.gz | tar oxvf -` extrait le contenu d'un fichier **tar** compressé avec **gzip**.

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.

5.6.34 `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 chaine `Bonjour` dans les fichiers du catalogue courant, on écrira

```
grep Bonjour *
```

Il existe plusieurs variantes à cette commande, dont **egrep**, qui permet de spécifier des conditions plus générales.

5.6.35 `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 une **macro** spécifique: `nroff -man foo.1 > foo.txt`.

6 Introduction à l'éditeur Emacs

Emacs est bien plus qu'un éditeur textuel. Pourtant c'est en tant que tel que nous allons en parler ici. Sous deux angles particuliers: la rédaction de texte et celle de programmes informatiques. Deux aspects fort utiles seront en outre évoqués: l'utilisation comme **mailer** et comme éditeur de répertoire. 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.

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, en appuyant séparément, contrairement à *CTRL* ou *Meta*, et la séquence *CTRL–S*, bien souvent nécessaire, est inutilisable. De plus, la touche *BACKSPACE* appelle le mode d'aide, il faut utiliser *DEL* pour effacer. Avec des **bindings** appropriées 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/`[14]. Le fichier **.emacs** est le fichier d'initialisation d'Emacs.

Note: ceci est la version 1.30 de cette introduction à 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.

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*).

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 long nom. (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.

---

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

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–v déroule l'autre fenêtre vers le bas.

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–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

Voilà 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 souvenir 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 la tuer) (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 buffer et registres. On peut éditer un buffer, le sauver mais ces opérations ne s'appliquent pas au contraire 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 reg. 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 remplacements

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'elle 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**[15] où viennent s'appondre 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 un arbre 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`. 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 9.

---

[15] ne pas confondre avec la commande **rmail**, décrite à la section 12.2.2,

6.22 Mémento des fonctions et raccourcis-clavier

6.22.1 Général

CTRL–x extension de caractère: suivit d'un caractère.

Meta–x exécute une commande avec un nom long.

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

CTRL–l rafraîchissemement (**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.

6.22.2 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.

6.22.3 Buffers

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

CTRL–x CTRL–b liste les buffers utilisés.

f permet une selection 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 (&&).

6.22.4 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.

6.22.5 Edition

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

CTRL–d efface le caractère suivant (delete).

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.

6.22.6 Fenêtres

CTRL–x 1 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.

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.

6.22.7 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.

6.22.8 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` appond la région à un registre.

6.22.9 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.

6.22.10 Macros

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

CTRL–x) termine cet enregistrement.

CTRL–x e appelle la dernière macro définie.

6.22.11 Répétition

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

CTRL–x ESC répète une fonction complexe.

6.22.12 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.

6.22.13 Commentaires

Meta–; insère ou aligne un commentaire (sauf en début de ligne).

CTRL–x fixe la colonne des commentaires.

Variables `comment-start` contient la chaine de début de commentaire.
  `comment-end` idem avec fin de commentaire.
  `comment-start-skip` chaîne contenant un commentaire vide.

Abbréviations *CTRL–x +* génère une abréviation du mot précédant le curseur.
  CTRL–x ' effectue l'expansion de 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 modifiéees.
  Meta–x `write-abbrev-file` écrit un fichier contenant les abréviations.
  Meta–x `read-abbrev-file` lit le fichier contenant les abréviations.

6.22.14 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 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.

6.22.15 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.

6.22.16 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.

6.22.17 Aide

CTRL–h a **apropos**, liste les commande 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

En général, on admet la convention de faire suivre le nom de fichier d'une extension formée d'un point ainsi que d'une abbréviation du type, souvent sur trois lettres. La figure 3 en page 34 liste quelques unes de ces extensions et indique les programmes de conversion ou de visualisation appropriés.

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 5.6.33 page 23)*.

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`.

Extension	**type**	**création**	**extraction**
.jpg	Image compressée avec perte	cjpeg	djpeg
.gif	Image	**pbmplus**	**pbmplus**
.xbm	Image (X bitmap)	**pbmplus**	**pbmplus**
.lha .lzh	Archive compressée	lha	
.zip	Archive compressée	zip	unzip
.gz .z	fichier compressé	gzip	gzip -d
.Z	fichier compressé	compress	uncompress, gzip -d
.shar	archive shell	shar	unshar
.tar	archive	tar cvf	tar oxvf
.au	Son	**sox**	**sox**
.iff	Image ILBM/Amiga	**pbmplus**	**pbmplus**
.zoo	Archive compressée	zoo	zoo
.arj	Archive compressée	-	unarj
.arc	Archive compressée	arc	arc
.dvi	TEX	tex	xdvi, dvips
.ps	Postscript	divers	ghostview

FIG. 3 - Extensions des différents formats

7.3.1 Conversions de textes accentués

Le programme **accents** permet la conversion de formats divers (macintosh, ms-dos pc, ...) 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).

7.3.2 Conversions de sons et d'images: sox et pbmplus

Pour la conversion de sons ou d'images, les commandes **sox** et le sous-ensemble de commandes des logiciels **cjpeg**, **djpeg**, et **pbmplus** sont disponibles[16].

7.3.3 Conversions de fichiers multimédia

Le protocole d'encodage **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 14.6 page 49)*.

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**.

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,
- la source doit être disponible et portable,
- algorithme sans pertes,

---

[16] Les logiciels pbmplus sont disponibles dans `/share/bin/pbmplus`

Logiciel	**version**	famille	**taille**	**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	1.9	LZ77H	1021043	2 mn
hpack	0.75a	PPMC	1005367	85 mn
comp	2	arithm.	890976	27 mn

FIG. 4 - Comparaison d'algorithmes de compression

- le jeu de test doit être publié.

Les informations de la figure 4 en page 35 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.

7.5 Transferts de fichiers

Les commandes **sz** (et **rz**) permettent de transférer des fichiers jusqu'à votre terminal ou ordinateur personnel (ou vice-versa) *(voir section 22.3.3 page 72)*. La commande **ftp** permet de transférer des fichiers entre systèmes distants sur **Internet**. La commande **rcp** 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 14.6 page 49)*.

7.6 Cryptage 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 *(voir section 10.3.3 page 41)*.

7.7 Impression de fichiers

7.7.1 Soumission de travaux

La commande pour soumettre un fichier à l'imprimante est `lp`[17].

7.7.2 Etat de la queue

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

7.7.3 Destructions de travaux

La destruction de travaux (**jobs**) est possible avec la commande `cancel`[18], 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.

---

[17] parfois lpr

[18] parfois `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 Autres interfaces

Le **shell tcsh** présenté jusqu'ici n'est qu'une des nombreuses façon d'interagir avec notre système UNIX. Il en existe bien d'autres dont certaines sont indiquées ici.

8.1 Minishell

Pour les utilisateurs ne désirant pas utiliser le shell, il est possible d'utiliser une interface plus simple, nommée **Minishell**. Pour l'activer il faut écrire `ms.sh`. On peut aussi demander le remplacement du **shell** standard par Minishell.

8.2 BBS

Pour les utilisateurs désirant accéder au système d'une façon ressemblant aux BBS habituels, il est possible d'accéder à un BBS qui pour le moment ne supporte ni les **news** ni le **mail** avec `rlogin mips -l bbs`. On peut aussi entrer le login `bbs` sur le mips.

8.3 Autres shells

Les autres shells disponibles sont **sh**, aussi appelé le **Bourne shell**, **ksh** aussi appelé le **Korn shell** et **csh**, aussi appelé le **C-Shell**. Il y a aussi **tcsh**, une extension de **csh**, et **perl**.

Le shell **sh** est particulièrement conseillé pour la rédaction de courts programmes en langage shell nommés **scripts** *(voir section 9.3 page 38)*.

8.4 Interface graphique

Une interface graphique (ou plutôt un ensemble d'interfaces graphiques) nommée **X-11** est disponible pour autant que le logiciel serveur, affichant les fenêtres sur un terminal (p. ex. ordinateur personnel) soit disponible. Une description du protocole ainsi qu'une liste des logiciels disponibles pour différentes plates-formes se trouvent dans la section 20.

9 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.

9.1 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.

9.1.1 Variables

Le système utilise des variables comme

TERM type de terminal courant

PATH chemin de recherche des commandes

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`, `setenv` ou `set`. On modifie les variables comme suit:

`setenv TERM ansi` mettre le terminal à ANSI

`set correct=all` mode correction tout

`set prompt="hello>"` changer le prompt en hello>

Notez bien que ces modifications sont temporaires (jusqu'à la fin du **shell** courant). Pour les rendre permanentes, il faut les ajouter au fichier **.cshrc**.

9.1.2 Commandes de configuration

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

stty pour initialiser les paramètres du terminal,

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 catalogues. 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 catalogues au mode 755, on utilisera `umask 022`. Pour obtenir un peu plus de sécurité, on peut utiliser `umask 077`.

La commande **stty** est particulièrement difficile à manier et en général ce n'est pas nécessaire. En cas de besoin les **manpages** peuvent être d'un grand secours.

9.1.3 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`.

.cshrc Ce fichier contient des séquences d'instructions exécutées à chaque lancement de shell ou de sous-shell.

.login Ce fichier contient des séquences d'instructions exécutées au premier shell.

.logout Ce fichier contient des commandes exécutées en fin de connexion.

.plan Ce fichier contient des indications personnelles comme par exemple le projet sur lequel vous travaillez. Voir la commande **finger**. Pour que d'autres utilisateurs puissent y accéder, il faut que son mode (sa protection) soit `rwxr--r--`, soit avec une notation abrégée `chmod 744 .plan`. Cependant, cela ne fonctionnera que si votre catalogue est accessible (mode x). Cela se fait également avec une commande comme `chmod +x $HOME` *(voir section 5.6.9 page 19).*

.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.

.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.

.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**.

9.2 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** (**.login** ou **.cshrc**). Pour changer une variable, il faut utiliser l'ordre **setenv**, ou **set**. Une bonne idée est de regarder les exemples et de tester ses modifications avec un `rlogin localhost`. 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.

9.3 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.

9.3.1 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.

9.3.2 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**.

```
/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 (1ère ligne). Par exemple:

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

ou

```
#! /share/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 **csh**, ou l'opérateur `.` sous **sh**.

9.3.3 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**). De plus, certains le considèrent comme peu fiable.

csh A cause de ses nombreuses contradictions dans sa grammaire et les trop nombreuses mises à jour, ce shell ne peut être utilisé que pour la session interactive, mais il est avantageusement remplacé par **tcsh**.

tcsh Extrêmement conseillé pour les sessions interactives en raison de ses nombreuses possibilités d'édition comme de complétion, il est également utilisable comme interprète de scripts.

ksh Cette extension normalement compatible avec **sh** est suffisamment normalisée pour garantir la compatibilité **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 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**.

awk Ce langage 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.

9.3.4 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 catalogue courant qui finissent en .tgz en .tar.gz.

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

   if [ $# != 2 ]; then
         echo "$0 fromext toext"
         echo "$0: bad args"
         exit 1
   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).

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.

10 Aspects de sécurité

10.1 Piraterie

On entend par ce terme une des actions suivantes:

- s'introduire dans un système sans y être autorisé ou en ne respectant pas les limitations d'accès,
- échanger des logiciels sous licence.

10.1.1 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'autre 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 l'utilisateur final a intérêt à assurer un minimum de sécurité concernant son accès, par exemple:

- ne pas partager son compte avec plusieurs utilisateurs,
- choisir un mot de passe respectant les règles élémentaires *(voir section 10.2 page 40)*,
- 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 prison.

10.2 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.

10.2.1 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.

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,
- 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 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 10.4 page 41)*.

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 cryptage des sessions *(voir section 12.1.3 page 43)*.

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.

10.2.2 Changement du mot de passe

La commande passwd 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 faux. Il n'y aura pas d'écho.

10.3 Protection de vos données

10.3.1 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). Nous effectuons automatiquement des copies incrémentales des catalogues utilisateur de façon journalière, conservées sur le site. D'autres sites peuvent avoir des conventions différentes, renseignez-vous auprès de votre administrateur système.

10.3.2 Limitation d'accès à vos fichiers

Lors de l'installation de tout nouveau compte, le système crée automatiquement les catalogues `public` et `private` et autorise l'accès au catalogue par défaut. Pour les autres utilisateurs, les catalogues par défaut et `public` peuvent être accédés et lus. Le catalogue `private`, quant à lui, est réservé à son possesseur *(voir section 5.6.9 page 19)*.

Il peut être conseillé de protéger certains fichiers personnels en en limitant l'accès, avec la commande `chmod`, ou en déplaçant le fichier dans le sous-catalogue `private`.

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 **catalogue par défaut**

`chmod 711 .` n'importe qui peut pénétrer et lire vos fichiers non protégés spécifiquement dans votre catalogue, mais vous seul pourrez en lister le contenu[19]

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

`chmod +r .plan` Votre activité est donc accessible en lecture par tous, par exemple pour la commande `finger`.

`chmod +rx public` Vous autorisez l'accès au catalogue public. D'autres commandes de ce genre peuvent être utiles, par exemple, pour votre **homepage** *(voir section 21.6 page 67)*.

10.3.3 Cryptage 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. Evidemment, 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 cryptage 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 22.4 page 76)*.

10.4 Compte personnel

Votre compte est *PERSONNEL*. Vous êtes prié de ne jamais laisser quiconque d'autre que vous-même y accéder. Evitez de même d'entrer le mot de passe en présence d'autres personnes. Ne laissez jamais un terminal non bloqué (l'option LOCK SCREEN de `screen` *(voir section 5.6.20 page 22)* ou la commande `xlock` sur terminal X-11 *(voir section 20 page 59)*). 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 10.2 ainsi que celles de votre administrateur système est essentiel.

Sous UNIX, il est impossible à l'opérateur de connaître, à partir de la version encodée de votre mot de passe, le mot de passe en clair[20]

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**.

---

[19] Dans certains cas, ceci peut avoir plus d'inconvénients que d'avantages.

[20] Le système d'exploitation stocke les mots de passe dans un fichier nommé en général `/etc/passwd`. Parfois, et c'est le cas sur notre système, même les mots de passe encryptés ne sont pas visibles. Même en possédant les mots de passe encryptés, le décodage peut prendre des années sur un système rapide. Ce n'est donc pas strictement impossible, mais extrêmement long.

Deuxième partie
Réseaux informatiques

11 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. 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**.

12 Principaux réseaux UNIX

12.1 Internet

12.1.1 Introduction

Ce réseau est un réseau temps réel, où chaque ordinateur est relié aux autres par des liaisons fixes (par exemple des lignes louées) et via des passerelles entre réseaux locaux. Au départ réseau académique reliant les universités des Etats-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 facon 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).

Il existe cependant des protocoles permettant des connexions temporaires (via ligne téléphonique commutée à l'aide d'un **modem**, 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**[21] et aux **news**. Parmi elles, citons **UUCP** *(voir section 12.2 page 44)*.

12.1.2 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 figure 5 en page 43). Des **routers** permettent d'aiguiller les différents paquets entre sous-réseaux en fonction de leur adresse destination.

Chacune des machines connectées sur Internet possède une 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 est automatiquement effectuée par un serveur de nom *(voir section 19 page 58)*. Chacune des adresses textuelles (de la forme `machine.sousdomaine.domaine.topdomain`) correspond également à une ou plusieurs adresses numérique (ou **adresse IP**).

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, depuis assez longtemps on utilise le concept de sous-réseau (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).

Internet manquant 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 Classe 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. `128.b.c.d`) étaient toutes des Classes A (pour simplifier le routage). Désormais, on alloue souvent des Classe C à l'intérieur de Classe B ou A. Notez que certaines addresses sont réservées (par exemple `a.b.c.0` et `a.b.c.255`).

TCP/IP offre deux services de transport de bas niveau: TCP et UDP, et par dessus ces services de transport, d'autres services de plus haut niveau ont été définis (particulièrement dans les **RFC**s.). TCP/IP est un protocole ouvert et évolutif,

---

[21] Des passerelles mail pour de nombreux services normalement interactifs existent néanmoins.

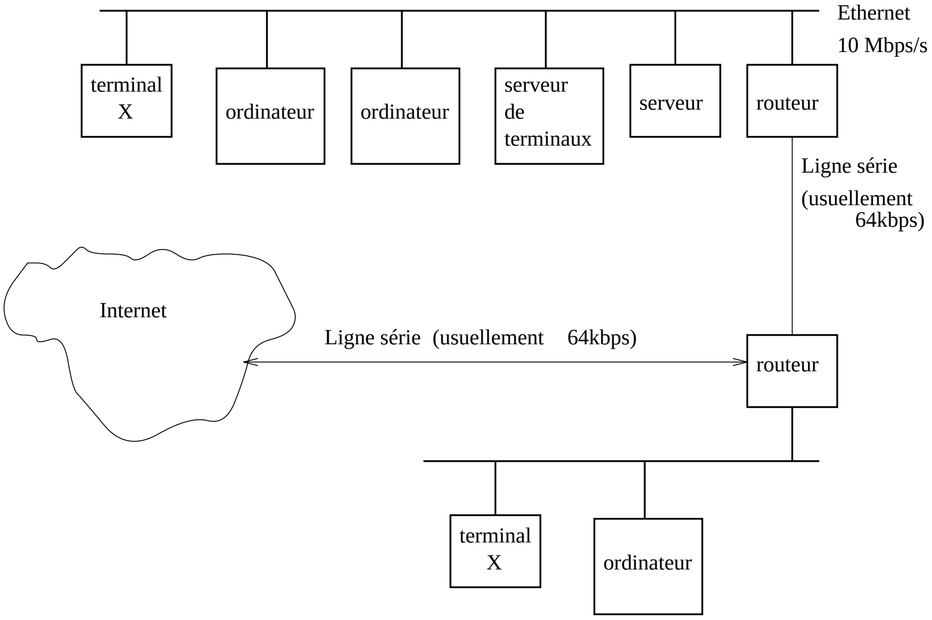


FIG. 5 - Exemple de configuration d'un réseau IP

de sorte que sans cesse de nouveaux protocoles 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*).

Au bas niveau, TCP/IP peut être transmis par des média variés, comme **Ethernet**, **Token Ring** ou des liaisons sérielles.

12.1.3 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 passe[22].

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**.

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 **FAQ**s du forum de news **comp.security.misc**.

Authentification On entend par ce mot le fait de vérifier et d'assurer 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 différents types d'identification qui sont décrits au prochain paragraphe. Certaines des méthodes sont acceptée devant les tribunaux. Souvent, les Etats interdisent certaines de ces méthodes.

Cryptage On entend par ce mot le fait d'encoder les données transmises de facon 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 22.4 page 76)*. D'autres sont très primitives

---

[22] Un 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].

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 des 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.

Avantages – le mot de passe ne circule pas.

Inconvénients – des méthodes de piratages cependant complexes[23] (anglais: *sequence number guessing*) rendent périlleux ce type d'identification sans une installation sûre (surtout au niveau du **router**).

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 cryptage 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 encryptée 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 22.4 page 76)*.

Conseils d'accès Les conseils suivant peuvent être utiles:

- pas de mot de passe en clair,
- 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` plutot que `xhost`, surtout à travers un grand réseau.

12.1.4 Evolution — 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 comme **videotex** (penser seulement à l'asymétrie des vitesses).

On peut espérer que des initiatives publiques ou privées permettront de rééquilibrer cet état de fait: c'est un des buts d'**ALPHANET NF**.

12.2 UUCP

12.2.1 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.

---

[23] Une annonce du **CERT**, un organisme de sécurité, en a parlé récemment. Cette méthode est cependant connue depuis 1985 quand elle a été publiée dans un journal de **Bell labs**.

12.2.2 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 défaut[24]. 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, rsmtp, rnews, ...) en fonction des indications dans les fichiers de contrôle.

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 associé de données et agissent sur celui-ci.

12.2.3 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.

13 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.

13.1 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 suffisante, 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 fichier interactif.

telnet Appel de système distant.

talk Discussion interactive.

nfs Partage de fichiers.

mud Jeux multi-utilisateurs distribués.

irc Discussion de type CB distribuée.

gopher Banques d'informations textuelles distribuées.

WWW Services multimédia interactifs distribués.

13.2 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.

news Forum de discussion. Permet d'échanger des points de vue dans des conférences à thèmes spécifiques avec d'autres utilisateurs.

uucp Transmission de fichiers.

---

[24] Sur notre système, il y a un utilisateur par machine se connectant, et un utilisateur anonyme **nuucp** avec moins de possibilités.

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 cinq serveurs, à savoir

newsserv@alphanet.ch permet des modifications de configuration **news** pour les sites appelant **mips**.

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**

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

14 Mail (courrier électronique privé)

14.1 Introduction

Le système **mail** sert à l'échange de courrier électronique privé entre correspondants connectés au réseau **ALPHANET**, **Internet**, **CBMNET**, **FidoNet**, **COMPUSERVE**, **X.400**, **UUCP** et divers autres réseaux. L'interface utilisateur comprend les commandes **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'accès[25].

14.2 Savoir-vivre

Echanger 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 facilités 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 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),
- marquer clairement les passages sarcarstiques[26],
- 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 conseillée est 70 caractères 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**,
- 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 12.1.3 page 43)*,
- 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 d'être sûr de l'identité de son correspondant.

Il faut également savoir que l'envoi de messages non sollicités à teneur publicitaire est normalement considéré comme un manquement grave aux règles élémentaires de comportement sur les réseaux.

---

[25] Sur ALPHANET on transforme les mailing-lists en forums et on diminue ainsi la charge mail totale en centralisant les requêtes à *newsadm@alphanet.ch*.

[26] Le symbole `:-)` peut être utilisé à cet effet. Si vous ne comprenez pas, tournez la tête de 90 degrés: 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.

14.3 Interface utilisateur

De nombreuses interfaces existent. Elles diffèrent surtout par la facilité d'emploi et la présentation (terminal **VT100** ou interface **X-11** *(voir section 20.7 page 61)*. 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*).

14.3.1 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 extrêment 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.

Exemple: Le professeur Falken envoie un mail à Bill CLINTON. On voit que sur le mips, on termine les messages par *CTRL–D*. De plus, 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.

```
<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>
```

14.3.2 elm: 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 catalogue `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 9.1.1 page 36)*.

14.4 Personnes de contact

Le tableau en figure 6 en page 48 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**.

14.5 Aspects techniques

14.5.1 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, en cas de 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**, les 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.

Adresse	**description**
sysadm	administration système générique
acctadm	comptes
uucpadm	système UUCP
fileadm	archives de fichiers
fnetadm	système FidoNet
sladm	SLIP
usenet	
newsadm	système de news
postmaster	
mailadm	système de mail
admin	club ALPHANET NF

FIG. 6 - Adresses **mail** des administrateurs pour alphanet.ch

14.5.2 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 tracer 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 12.1.3 page 44)*. 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 22.4 page 76)* (pour l'authentification aussi bien qu'au chiffrement du contenu).

14.5.3 Spécification d'adresses

Le routage mail sur **ALPHANET** est très simple. 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 très mal définie. Remarquons qu'elle est souvent utilisée par les mécanismes de transfert de messages, cela est évidemment normal. Par contre l'utilisateur ne doit pas s'en servir.

% Dans quelques rares cas ce symbole peut être utilisé pour spécifier une passerelle.

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 [11] 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`).

14.5.4 TOP domains

Les TOP domains sont les domaines terminant une adresse mail. Il existe énormément de TOP domains, dont par exemple:

edu Education aux Etats-Unis (universités).

gov Gouvernement des Etats-Unis.

com Organisations à but lucratif ou général.

org Organisations à buts non lucratifs.

net Organisations de réseaux.

codes ISO Les codes ISO à deux lettres correspondent à des pays. Par exemple:

  ch Suisse.

  us Etats-Unis.

  de Allemagne.

fr France.
at Autriche.
au Australie.
jp Japon.

14.5.5 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 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 `2:301/315.50`
- sinon, votre correspondant doit envoyer un mail à

  `UUCP` sur `2:301/315.50`

  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 catalogue local `~admin/info/news.announce.newusers` et dans le forum de **news alt.internet.services**.

14.6 Transfert de fichiers via mail

Il est possible de transférer des fichiers via courrier électronique privé avec **uuencode** *(voir section 5.6.32 page 23)* ou grâce au standard multi-formats **MIME** *(voir section 7.3.3 page 34)*. 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 découragé: 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 à votre site.

15 News (messages publics)

15.1 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. 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*). Le système automatique d'archivage des FAQs des **forums** disponibles localement met à jour les FAQs par thème (dans `/share/archives/autofaq/faq`) ou par hiérarchie (`/share/archives/autofaq/faq-by-hierarchy`).

L'interface utilisateur est composée des commandes **postnews** (écriture) et **tin** (lecture et écriture *(voir section 15.5.1 page 51)*).

15.2 Catégories

Les différentes catégories de forums disponibles se différencient par leur nom. Leur structure est similaire à un **système de fichiers**[27], si l'on remplace les points par ce qui sépare les catalogues (sous UNIX: `/`).

comp Forums concernant l'informatique.

misc Sujets divers.

rec Activités récréationnelles.

gnu Etat du projet GNU.

alphanet Réseau alphanet.

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.

fido Réseau (partiel) **FidoNet**.

xgp Réseau XGP.

francom Réseau (partiel) **FrancoMedia**.

15.3 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 catalogue

`/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 visualisés avec **vi**.

Voici donc une liste de quelques forums importants, dont au moins les deux premiers devraient être lus par tous les utilisateurs du réseau ALPHANET. Le troisième est également intéressant car il contient des périodiques d'information sur les news en général, mais en anglais. Le quatrième (alphanet.local) peut être utilisé pour votre premier article de news afin de voir si cela marche.

alphanet.general Informations et annonces importantes ALPHANET.

alphanet.misc Discussions entre utilisateurs d'ALPHANET.

news.announce.newusers Explanatory postings for new users. (Moderated).

alphanet.local Forum local de test.

alphanet.aminet.weekly Nouveautés sur **Aminet** par semaine.

---

[27] d'ailleurs les news, sous UNIX sont disponibles traditionnellement dans `/usr/spool/news`

alphanet.infotech.edupage Information Technology — EDUPAGE.

alphanet.infotech.hotlist Information Technology — HOTLIST.

alphanet.ml.self The SELF object-oriented language.

alphanet.fido.chf.322.test Aire de test FidoNet.

alphanet.fido.chf.amiga Aire Amiga FidoNet.

alphanet.fido.chf.pgmrs Aire Programmeurs FidoNet.

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.

15.4 Savoir-vivre

Les règles de savoir-vivre du courrier électronique privé *(voir section 14.2 page 46)* 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 les suivants:

- l'accès aux news n'est pas un droit fondamental,
- ne pas poster un message reçu en courrier privé (**mail**) sur les **news**, car le correspondant a le droit à sa sphère privée,
- n'utiliser d'accents que sous les hiérarchies alphanet et fr, et toujours des accents **ISO Latin 1**,
- ne poster un message qu'une seule fois. Il faut mieux **cross-poster** que de poster un message plusieurs fois.
- citer, ne pas reposter en entier un article juste pour y ajouter quelques lignes de commentaires. Particulièrement, les signatures n'ont aucune raison, en général, d'être citées,
- en général, le caractère de citation sera > afin d'obtenir quelque chose du genre:

  > la transmission série à 2400 bps à une vitesse de 2400/8 cps

  non, plutôt 2400/10, soit 240 cps.

  l'usage de séquences de citations plus longues prétérite 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),
- lire toutes les réponses et éviter de redire ce qui a déjà été dit. 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:,
- ne pas critiquer autrui sur la forme, mais sur le fond,
- ne pas hésiter à encoder un message dont le contenu peut froisser certains lecteurs (**rot-13**),
- 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,
- ne jamais écrire de message lorsqu'on est excédé, énervé, ou sous l'influence d'un breuvage (solide, liquide ou autre) quelconque :-;

15.5 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 **X-11** *(voir section 20.7 page 61)*. 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*).

15.5.1 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 messages **cross-posted** 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 distance[28] via **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 au message suivant non lu (ce qui permet une lecture de tous les nouveaux messages dans tous les forums par appui répétitif sur *TAB*) et la sortie du niveau courant. A chaque fois qu'une liste de forums, de messages 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 souscrire 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 message d'une thread avec *RETURN* (en se déplaçant bien sûr préalablement sur le message 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 message non lu avec **TAB**, et visualiser l'ensemble des messages d'une discussion avec *l*.

liste des messages dans un sujet particulier (thread)

affichage d'un message proprement dit On tape *ESPACE* pour passer à la page suivante, *b* pour revenir à la page précédente, *s* pour sauver un message (par défaut dans le catalogue `News/`), *m* pour envoyer une copie d'un message via **mail** à un tiers, *r* pour répondre en **mail** (courrier privé) à un message, ou encore *f* pour répondre en **news** (public) au message. On sort évidemment de ce niveau avec *q*. On passe au message non lu suivant avec *TAB*. On peut détruire un message (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 5.6.1 page 17)*.

15.6 Aspects techniques

15.6.1 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. 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. Un système efficace de détection de doublons (anglais: *duplicates*) permet d'éviter l'apparition en plusieurs exemplaires de messages. La propagation d'un article de **news** se fait donc par inondation successives de divers sites. La figure 7 en page 53 montre les liens mondiaux entre serveurs de news.

15.6.2 Sécurité

Les **news** n'assurent pas l'identité du rédacteur. Cependant, il est parfois possible de tracer la provenance d'un message 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 12.1.3 page 44)*. De plus, les messages transmis peuvent être modifiés par un pirate (anglais: *cracker*). Une certaine authentification peut être obtenue à l'aide de logiciels comme **PGP** *(voir section 22.4 page 76)* (les fameuses signatures PGP de certains article s de news, considérées par certains comme du *gaspillage de bande passante*).

16 Communication entre utilisateurs

16.1 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.

16.2 `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.

---

[28] 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.

Fig. 7 - Graphe de connectivité mondiale Usenet

Exemples:

`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.

16.3 `who`

Permet de connaître les utilisateurs en ligne.

16.4 `w`

Donne des informations comparables à who.

16.5 `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 le nom du terminal où se trouve l'utilisateur s'il en existe plusieurs. Si cette dernière information n'est pas précisée, le premier trouvé sera sélectionné.

16.6 `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.

16.7 `irc`

Cette commande permet d'accéder à un système de téléconférence distribué et en temps réel. Dans certains cas, particulièrement si votre émulation VT100 est déficiente, il est conseillé d'entrer `irc -d` pour simplifier l'interface.

17 Communication à distance

17.1 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/`.

17.2 `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.

17.3 `ftp`

Cette commande permet de transférer des fichiers depuis vers un site distant. Une utilisation intéressante de cette commande est dans les accès anonymes (en indiquant ftp ou anonymous comme nom d'utilisateur distant). Il faut évidemment que le site distant autorise l'accès. Dans ce cas, il faut entrer `ftp` comme nom d'utilisateur et votre adresse **mail** comme mot de passe. Il est possible d'exécuter des commandes **shell** depuis ftp en les précédant de `!`. Evidemment, pour changer de catalogue local cela ne fonctionne pas (un sous-shell est crée, change de catalogue et quitte, donc perd le nouveau catalogue). On utilisera alors `lcd`.

Notez que l'accès à un site **FTP** distant n'est pas un droit: en particulier, il faut limiter les transferts de masse en dehors des heures de bureau *locales*, choisir le site le plus près, et donner son adresse **mail** à la place du mot de passe.

En général, il vaut mieux chercher un peu plutôt que de se jeter sur le premier site. Le système **archie** permet de rechercher des fichiers.

Exemple: Voici un exemple d'utilisation du FTP anonyme (anglais: *anonymous 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 catalogues 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 catalogues 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
jungfrau.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.
150 Opening data connection for jungfrau.jpg (193.72.186.3,1278) (58705 bytes).
226 Transfer complete.
local: jungfrau.jpg remote: jungfrau.jpg
58705 bytes received in 0.22 seconds (2.6e+02 Kbytes/s)
```

Similairement, on envoie des fichiers avec put/mput et on peut aussi afficher des donnees 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)
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.
```

17.4 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 sont répartis sur un réseau) de ne pas donner de mot de passe.

17.5 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).

17.6 rcp

On peut copier des fichiers à distance avec cette commande, avec les même restrictions vues à la commande **rsh**.

17.7 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.

17.8 traceroute

Cette commande permet de tracer le parcours de paquets IP jusqu'à un site distant. Son utilisation est similaire à celle de la commande **ping**.

17.9 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.

17.10 whois

Cette commande permet d'obtenir des informations sur des **domain**es 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: Battieux 6c
address: CH-2013 Colombier
phone: +41 38 41 29 29
e-mail: schaefer@alphanet.ch
nic-hdl: MS193
changed: fm@eunet.ch 940311
source: RIPE
```

18 Information sur le protocole SLIP

18.1 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 des 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.

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.).

18.2 Information sur le protocole

Sur la liaison série s'écoulent des paquets et des confirmations de réception (en effet SLIP corrige[29] 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.

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.

18.3 Configuration d'un réseau SLIP

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

---

[29] En fait, ce n'est pas tout à fait cela: SLIP ne corrige (ni ne détecte) rien du tout, mais TCP/IP s'en charge.

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**. Du côté UNIX, la session SLIP s'effectue à l'aide d'un **login** spécial.

18.4 Problèmes courants

18.4.1 CSLIP

Cette ariante 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.

18.4.2 PPP

Le protocole **PPP** n'est pas disponible.

18.4.3 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.

18.5 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 vient à demander au **modem** à 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.

19 Information sur le système DNS (Domain Name Server)

19.1 Introduction

Le système DNS permet d'effectuer la correspondance entre les adresses textuelles, comme ftp.alphanet.ch et les adresses en format Internet (comme 193.72.186.3), et vice-versa. On appelle la conversion nom vers adresse **resolving** et la conversion inverse **reverse mapping**.

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 compilées de manière à accéder aux services du serveur de nom de la machine principale, et éventuellement d'une machine secondaire en cas de panne. L'ensemble de ces services se nomme le **resolver**.

19.2 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
```

20 Système de multifenêtrage X-11

20.1 Introduction

Le système de multi-fenêtrage X-11, aussi appelé X-Window, a été développé au MIT et permet de rendre totalement indépendants plusieurs aspects d'un système de fenêtrage. Ce système permet de développer des applications clientes des services de l'interface, appelées clients, indépendamment du matériel, du système d'exploitation, de la localisation de l'interface et de l'aspect de celle-ci (**look-and-feel**).

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 X-11 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, comme par exemple X-Window, Motif, OpenWindows, etc.

Un autre intérêt du protocole X-Window, à part qu'il est supporté sur un nombre imposant de plates-formes, est aussi que ses sources sont disponibles sous 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 X-11 se trouve dans [14].

La plupart des systèmes implémentent au moins X-11 Release 4, certains X-11 Release 5, et la dernière version (disponible début mai 1994) est la version X-11 Release 6.

20.2 Terminologie

La terminologie X-11 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**).

20.3 Protocole

X-11 fonctionne normalement à l'aide du service **TCP** de **TCP/IP** dans la région du port 6000[30]. Les clients se connectent donc sur le serveur grâce à **TCP/IP** ou un médium local.

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

20.4 Configuration

La configuration d'une session se fait à travers divers fichiers de configuration et commandes UNIX.

20.4.1 Fichiers de configuration

La plupart des applications sous **X-11** utilisent des entrées dans un fichier de configuration standard `.xresources`. Le démarrage de la session s'effectue en général depuis `.xsession` ou encore `.xinitrc`. De plus, la configuration du **window-manager** se trouve dans un fichier du type `.twmrc`, `.mwmrc` ou encore `.tvtwmrc` suivant le type de gestionnaire (`twm`, `mwm` ou `tvtwm` qui sont respectivement le gestionnaire X standard, le gestionnaire **Motif** et le gestionnaire d'écrans virtuel.)

20.4.2 Commandes

A part les commandes de sécurité *(voir section 20.5 page 60)*, 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.

20.5 Sécurité

20.5.1 Introduction

Le système X-11 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 X-11 a pour but de n'autoriser l'accès qu'à des machines ou 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 n'est valable que sur des systèmes ne permettant pas l'utilisation de la machine simultanément par plus d'un utilisateur (typiquement des ordinateurs personnels).

xauth l'autorisation se fait sur la base d'une clé (qui peut être combinée avec un protocole de partage de fichiers réseau comme **NFS** et est donc relativement sûre). L'autorisation est donc au niveau de l'utilisateur lui-même, ou du moins de la connaissance d'une clé d'accès. Il faut d'ailleurs noter que la plupart des protocoles dits 'sûrs' modernes utilisent ce concept de clé, mais sous une forme bien plus complexe (voir **PGP**, **Kerberos**, **Secure-RPC**).

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 (dans notre cas votre ordinateur personnel).

20.5.2 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. L'accès ne vaut que pour la session courante.

---

[30] Le port 6000 est le port de base de l'écran 0. Les écrans suivants se trouvent à 6000 + x.

20.5.3 xauth

Cette commande est légèrement plus complexe en ce sens qu'il faut l'exécuter à la fois sur le serveur et sur chacun des clients 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 génèrent automatiquement une clé.

20.6 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** et au système de **pipe** si les deux systèmes concernés tournent sous UNIX. Notez également qu'il existe un script permettant de lancer l'interface en précisant le nom de la machine serveur qui affichera. Cela se fait par la commande:

```
~schaefer/x11_start.sh amiga1
```

On supposera que le serveur X est `amiga1.alphanet.ch` et les clients seront lancés depuis `mips.alphanet.ch`.

On peut également lancer un **login** graphique en demandant la configuration de l'outil **xdm**.

20.6.1 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.

Voici un exemple d'utilisation de **xauth** pour autoriser l'accès à votre écran.

```
<mips> schaefer:/d2/users/schaefer> xauth
    Using authority file /users/schaefer/.Xauthority
    xauth> add amiga1.alphanet.ch:0 MIT-MAGIC-COOKIE 1122334455
    xauth> exit
```

20.6.2 De votre côté (affichage, serveur)

Il suffit de lancer le serveur X et d'avoir les mêmes paramètres **xauth** pour le site distant. Il est évident que la méthode d'accès (interface graphique, messages) sera différente, mais le but identique, autoriser l'accès.

```
<amiga1> schaefer:/d2/users/schaefer> xauth
    Using authority file /users/schaefer/.Xauthority
    xauth> add mips.alphanet.ch:0 MIT-MAGIC-COOKIE 1122334455
    xauth> exit
```

20.7 Applications sous X

Les catalogues X-11 où se trouvent des exécutables standards sont entre autres dans `/usr/bin/X11`. Déplacez-vous dans ce catalogue avec **cd** car il s'agit d'un **link**. En remontant ensuite, vous vous trouverez dans la base du système X-11.

De nombreuses applications existent pour l'interface **X-11**, 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 **mips**. 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. 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 **cut-and-paste**).

Voici cependant une courte liste:

Mosaic interface graphique permettant l'accès au système hypertexte distribué **WWW**.

emacs éditeur multifonction.

xv afficheur d'images.

gnuplot génération de graphes.

ghostview visualisation postscript.
xterm support de terminal virtuel.
xarchie recherche de fichiers sur les serveurs **FTP**.
xedit éditeur de texte.
xdvi visualisation de fichiers **DVI** produits par le logiciel **TEX**.
xpaint logiciel de dessin.
xfig logiciel de génération de schémas.

20.8 Logiciels TCP/IP et X-11 du côté serveur (affichage)

20.8.1 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 **X-11**. 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.

20.8.2 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**, **mwm** ou encore **fvwm** et des applications sont également disponibles.

Par exemple, le système **Linux**, disponible pour PC, permet également de faire tourner XFree86.

20.8.3 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, 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 (GfxBase). 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 viennent 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.

20.8.4 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 7.5. C'est un logiciel commercial.

20.8.5 MS-DOS

Les produits suivants sont disponibles[31]:

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

---

[31] voir la FAQ **pcnfs**.

Je ne sais pas si des logiciels supplémentaires (par exemple de couche TCP/IP) sont nécessaires.

Une rumeur non vérifiée prétend que Microsoft distribue une version redistribuable d'un serveur X dans les milieux académiques.

20.8.6 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

20.8.7 MS-Windows NT

Il semble que X11R6 a été compilé pour Windows NT.

20.8.8 OS/2

Produit	**coût**	**compagnie**	**version**
X Windows	$150	IBM	X11R5

20.8.9 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

21 WWW (World-Wide Web)

21.1 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 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 bibliothèque, 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), au contraire de **Mosaic**, qui intègre 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.

21.2 Terminologie

La terminologie WWW est relativement fournie, mais n'intéresse probablement que les concepteurs de **homepages** ou 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 donnéees 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 LaTeX (il existe d'ailleurs des traducteurs automatiques).

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, en domaine public ou en commercial. 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 dans 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**, `http://www.alphanet.ch/~schaefer`.

21.3 Protocole

Le protocole HTTP utilise des connexions **TCP**. Au niveau utilisateur, on donne une spécification de document (aussi appelée **URL** pour Universal[32] Resource Locator) qui permet de retrouver le document concerné. Par exemple, on a les URL `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 `http://www.alphanet.ch:80/members.html`.

21.4 Sécurité

La version actuelle du protocole prévoit des extensions futures pour un protocole réputé sûr de communication entre serveur et client via clé publiques/privées (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.

21.5 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 **URL** du document. En général, les logiciels permettent de se déplacer vers un document quelconque avec une commande du type *go*, et s'ouvrent souvent avec une page par défaut utile si l'utilisateur ne spécifie pas de document sur la ligne de commande.

21.5.1 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'images. Un exemple est donné à la figure 8 en page 65. Notez que **lynx** ne fonctionne pas tout seul sous **X-11**[33]: ce logiciel nécessite un environnement terminal **xterm** (sans X-11, un login normal suffit).

21.5.2 Mosaic

Cette interface est graphique et est disponible sur la plupart des plates-formes. Une copie d'écran se trouve à la figure 9 en page 66. Il s'agit d'un logiciel librement redistribuable.

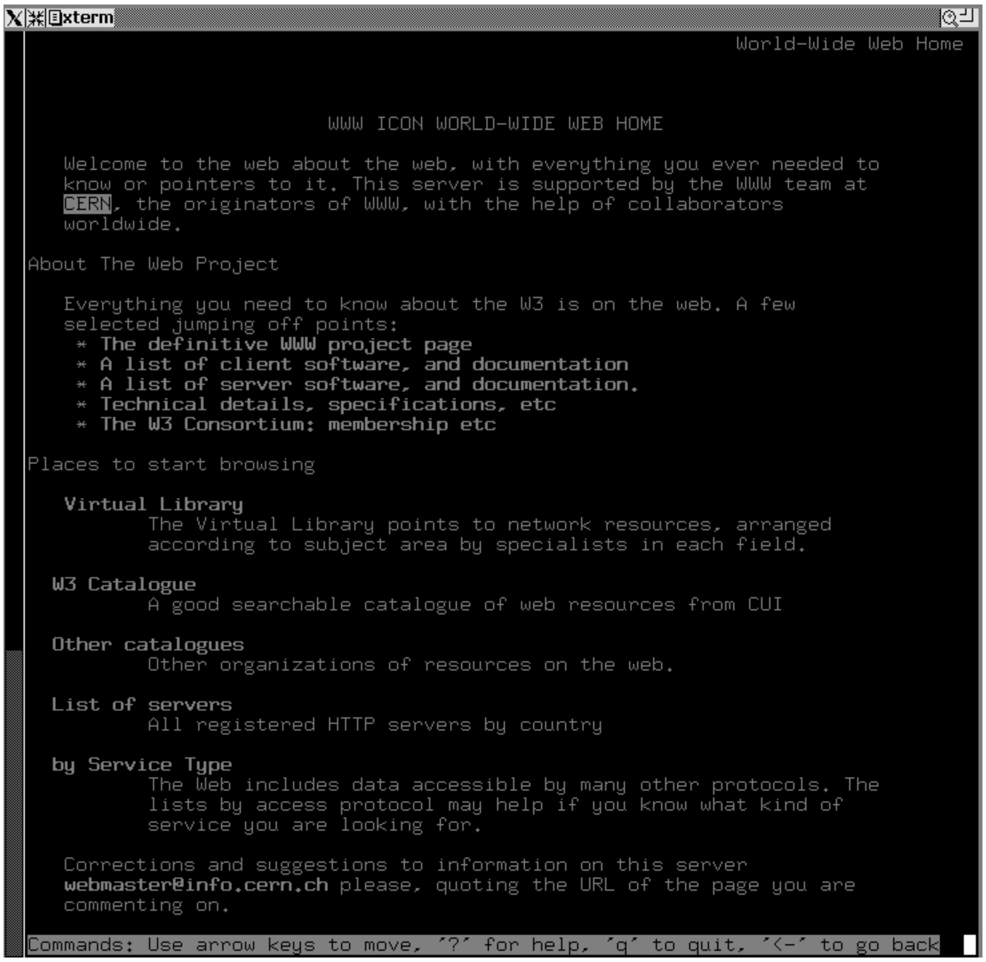
21.5.3 Chimera

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 redistribuables.

---

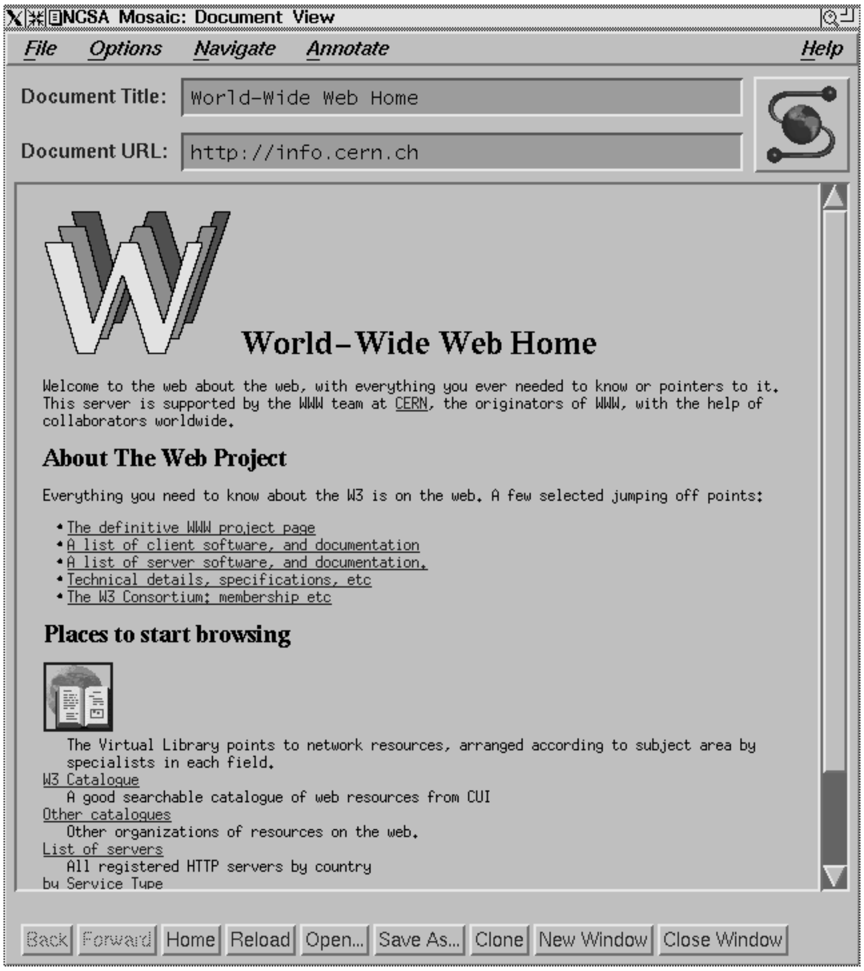
[32] Certains disent Uniform

[33] D'ailleurs le seul intérêt de lancer lynx sous X-11 est de limiter les données transférées, ce qu'on peut aussi demander sous **Mosaic** avec l'option *Delay image loading*.



```
                                                           World-Wide Web Home



                         WWW ICON WORLD-WIDE WEB HOME

   Welcome to the web about the web, with everything you ever needed to
   know or pointers to it. This server is supported by the WWW team at
   CERN, the originators of WWW, with the help of collaborators
   worldwide.

About The Web Project

   Everything you need to know about the W3 is on the web. A few
   selected jumping off points:
    * The definitive WWW project page
    * A list of client software, and documentation
    * A list of server software, and documentation.
    * Technical details, specifications, etc
    * The W3 Consortium: membership etc

Places to start browsing

   Virtual Library
          The Virtual Library points to network resources, arranged
          according to subject area by specialists in each field.

  W3 Catalogue
          A good searchable catalogue of web resources from CUI

  Other catalogues
          Other organizations of resources on the web.

  List of servers
          All registered HTTP servers by country

  by Service Type
          The Web includes data accessible by many other protocols. The
          lists by access protocol may help if you know what kind of
          service you are looking for.

   Corrections and suggestions to information on this server
   webmaster@info.cern.ch please, quoting the URL of the page you are
   commenting on.

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back
```

FIG. 8 - Fenêtre xterm contenant lynx sous X-11



Document Title: World-Wide Web Home

Document URL: http://info.cern.ch

World-Wide Web Home

Welcome to the web about the web, with everything you ever needed to know or pointers to it. This server is supported by the WWW team at CERN, the originators of WWW, with the help of collaborators worldwide.

About The Web Project

Everything you need to know about the W3 is on the web. A few selected jumping off points:

- The definitive WWW project page
- A list of client software, and documentation
- A list of server software, and documentation.
- Technical details, specifications, etc
- The W3 Consortium; membership etc

Places to start browsing

The Virtual Library points to network resources, arranged according to subject area by specialists in each field.

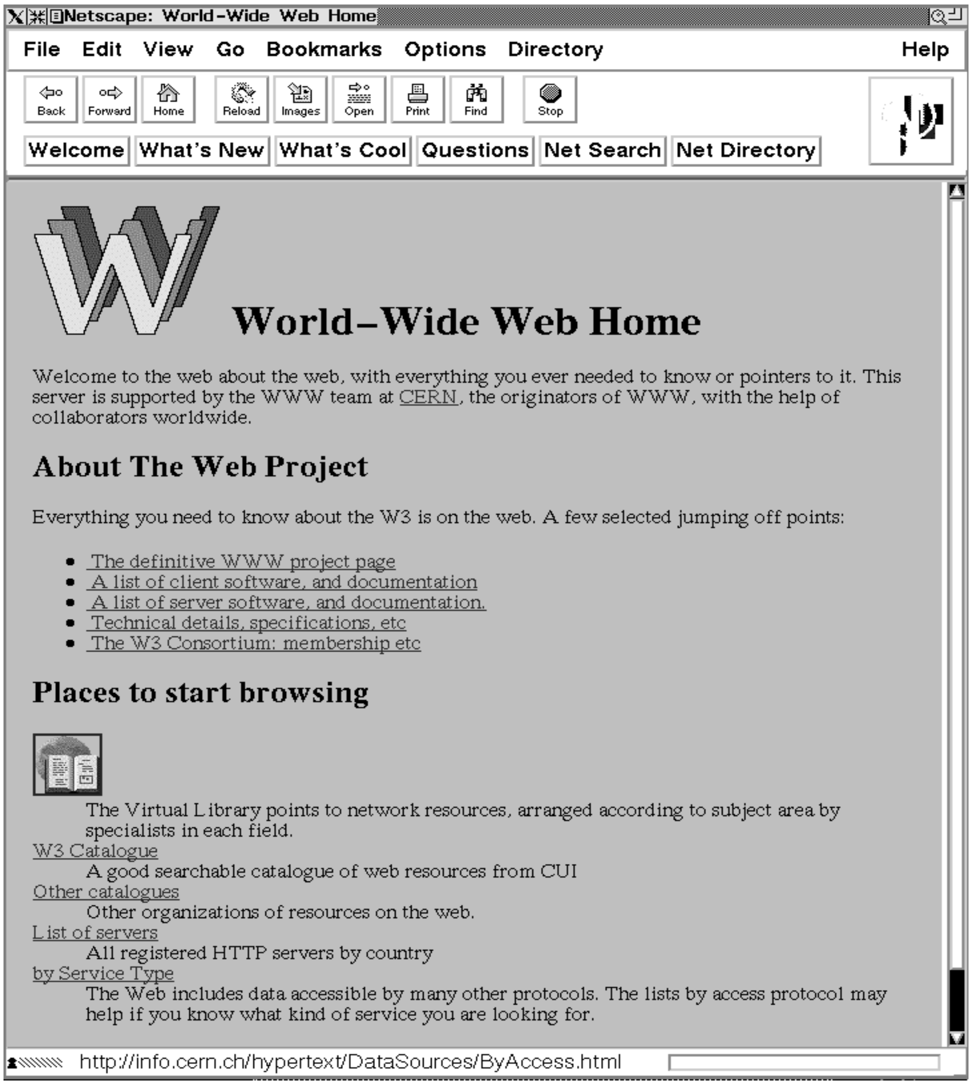
W3 Catalogue
A good searchable catalogue of web resources from CUI

Other catalogues
Other organizations of resources on the web.

List of servers
All registered HTTP servers by country

by Service Type

Fig. 9 - Fenêtre typique de Mosaic sous X-11



World–Wide Web Home

Welcome to the web about the web, with everything you ever needed to know or pointers to it. This server is supported by the WWW team at CERN, the originators of WWW, with the help of collaborators worldwide.

About The Web Project

Everything you need to know about the W3 is on the web. A few selected jumping off points:

- The definitive WWW project page
- A list of client software, and documentation
- A list of server software, and documentation.
- Technical details, specifications, etc
- The W3 Consortium: membership etc

Places to start browsing

The Virtual Library points to network resources, arranged according to subject area by specialists in each field.

W3 Catalogue
A good searchable catalogue of web resources from CUI

Other catalogues
Other organizations of resources on the web.

List of servers
All registered HTTP servers by country

by Service Type
The Web includes data accessible by many other protocols. The lists by access protocol may help if you know what kind of service you are looking for.

http://info.cern.ch/hypertext/DataSources/ByAccess.html

FIG. 10 - Fenêtre typique de Netscape sous X-11

21.5.4 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 destiné à devenir commercial. Son aspect est également légèrement plus attractif que Mosaic (voir figure 10 en page 67). Des versions pour la plupart des plates-formes sont disponibles.

21.6 Comment créer votre propre page WWW

Pour ce faire, il faut un serveur WWW (souvent appelé **httpd**). Sur le **mips**, ce serveur tourne en permanence. Il existe des serveurs pour la plupart des plates-formes. 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/`[34] et correspond au fichier `home.html` ou `index.html` se trouvant dans le catalogue `public_html` situé dans votre **homedir**. Le format de ce fichier est expliqué dans les documents à propos de **HTML**. Vous pouvez aussi prendre une page existante 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.

---

[34] Le / final est normalement optionnel, mais certains serveurs WWW ont un comportement étrange si celui-ci n'est pas ajouté.

21.7 Exemple d'un fichier source HTML

Le langage de description d'une page WWW se nomme **HTML**. De la documentation est disponible dans le catalogue `/users/admin/info/www`. En résumé, une source HTML est composée de lignes de texte comportant des séquences déchappement 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 `<B>` permet de mettre en gras le texte jusqu'au prochain `</B>`. Notez qu'un document **HTML** peut comporter des accents selon la convention du langage ou les accents standards.

Exemple: Page HTML

```
<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 1er" SRC="http://www.alphanet.ch/~schaefer/marc.gif">
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 client, avec documentation (anglais)</a>
<li><a href="http://info.cern.ch/hypertext/WWW/Daemon/Overview.html">
Liste de logiciels serveur, 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://www.w3.org/hypertext/WWW/Organization/Consortium/Prospectus/Overview.html">
Le consortium W3 (anglais)</a>
</ul>
</BODY>
</HTML>
```

22 Problèmes de télécommunication

22.1 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 **FAQ**s sur le sujet.

22.2 Les modems

Les **modem**s (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 22.4 page 72)*.

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 oeuvre pour assurer l'intégrité des données: par exemple **RTS/CTS**, appelé également contrôle de flux matériel (**hardware handshaking**) ou **XON/XOFF**[35]. 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 de diminuer le temps de connexion en compressant (comprimant) les données pendant la transmission.

Une très bonne source de renseignement sur les modems est la FAQ de **comp.dcom.modems**.

22.2.1 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.

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.

Le **CCITT** (nouvellement **ITU-T**) s'occupe de la normalisation en télécommunications. Les normes **Vxx** concernent plus particulièrement les modems.

22.2.2 Vitesse et modulation

La vitesse de transmission se mesure en général en **bits par seconde** ou **bps**. 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 la force du signal.

en phase On décale un signal dans le temps par rapport à une référence.

en fréquence On change la hauteur du "son" émis.

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 (**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`.

22.2.3 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 figure 11 en page 70 définit quelques-uns de ces standards.

---

[35] Notez 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. On recommande donc le contrôle de flux matériel.

Norme	**émission [kbps]**	**réception [kbps]**	**notes**
V34	28.8	28.8	sur réseaux fiables
V.FC	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	1.2	0.075	asymétrique figé à l'appel
V22bis	2.4	2.4	
V22	1.2	1.2	
V21	0.3	0.3	

FIG. 11 - Normes ITU-T et standards propriétaires

Norme	**type**	**notes**
MNP 1 à 4	correction	propriétaire, gain de l'ordre de 20% dus à la transmission par paquet (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.

FIG. 12 - Standards de correction d'erreur et de compression de données

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). La figure 12 en page 70 montre les quelques standards couramment utilisés.

22.2.4 Aspects techniques

La mise en oeuvre 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 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 on utilise un câble **null modem**.

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 recevoir.

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.

`ATDT038414081` Composition en **mode vocal** (**DTMF**) du numéro `038 41 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 générale utiles lors de perturbation 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 pourcents. 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 18 page 57)*.

Notez bien que la liaison entre ordinateur et 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 34)*.

22.2.5 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**).

22.3 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 **X-11**.

On ne traitera cependant dans cette section que des logiciels de communication classiques.

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**.

FIG. 13 - Protocoles de transfert courants

22.3.1 Terminologie

download Transférer un fichier du serveur distant au système local.

upload Transférer un fichier du système local au serveur distant.

22.3.2 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.

22.3.3 Protocoles de transfert de fichiers

Il existe une pléthore de protocoles de transfert offrant divers avantages. La figure 13 en page 72 résume ceux-ci.

Des protocoles bidirectionnels existent également, mais la plupart du temps ils sont propriétaires.

22.4 RNIS

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.

En Europe, la liaison RNIS de base se compose de deux canaux à 64 kbps 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 80 kbps 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**.

Glossaire

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 fichier 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 fichier 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 fichier 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 4.2 page 13)* 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 **X-11** 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 rendus.

ASCII American Standard Code for Information Interchange. Jeu standard de caractères usuels, sans accents.

AT&T Un des vendeurs d'un des systèmes d'exploitation les plus connus, AT&T System V Release 4 (en sous-licence chez presque tous les constructeurs, de Commodore à IBM en passant par Bull, Apple, Silicon Graphics, Sun, HP, Microsoft, etc).

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**.

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.

C Langage de programmation très commun sous environnement UNIX.

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 lecture article, par exemple).

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.

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**.

core Fichier généré par une défaillance logicielle d'une application ou du système d'exploitation lui-même, pour un debugging ultérieur (debugging post-crash).

cracker Un cracker est un utilisateur d'une machine qui essaie de franchir des systèmes de sécurité divers (protection logicielle, matérielle, cryptage, 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).

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 24/24 et listés dans la **nodelist** avec un argument spécial.

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.

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.

debugging On entend souvent par debugging la mise au point de programmes. Des termes équivalents existent en français: déverminage, déboguage, mise au point.

DFS Ce système de fichier distribué (Distributed File System) implémente les services habituels de **NFS**, avec en plus des mécanismes supplémentaires d'authentification et de cryptage. 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.

DNS Domain Name Server. *(voir section 19 page 58)*

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 **TEX**. Il décrit un document d'une facon similaire à **PostScript**.

Ethernet Un réseau rapide (10 megabits/s) disponible pour une grande variété de systèmes et développé au fameux XEROX PARC.

FAQ Une FAQ est en anglais, Frequently Asked Questions, et dénote un résumé de réponses aux questions courantes. La plupart des forums ont leur FAQ postée une fois par mois ou de façon régulière: il est très important de les consulter avant de poster. Ces FAQs sont archivées automatiquement dans `/share/archives/autofaq`.

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)

FTP File Transfer Protocol. 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 facon 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.

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 9)*.

Gopher Système d'information précurseur de WWW.

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. Une extension de ce système permet l'inclusion de multimédia et de références à des documents distants. *(voir section 21 page 63)*

Internet Un vaste réseau, au départ militaire et gouvernemental, puis académique (universités), et qui s'ouvre maintenant au 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 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.

IRC Internet Relay Chat. Un système de conférence texte 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 Jeu de caractères étendu comportant des accents standardisés. C'est le jeu standard sur **Amiga** et UNIX.

Jargon file Ce texte est une compilation de définitions de jargon UNIX et hackers. Il est disponible sous plusieurs formats, comme **ASCII**, **$\TeX$** 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 5.3 page 15)*.

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).

link Un lien est une référence à un autre fichier ou à un catalogue. Par exemple, sur mips, le catalogue `/users/admin/info` est en fait situé sur /share/archives/info. De même, le catalogue /scratch est lui situé sur /d3/scratch. Ou encore l'exécutable /usr/lib/sendmail est un lien à l'exécutable /usr/smail/bin/smail. Il existe plusieurs types de liens. 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.

Linux Implémentation UNIX **POSIX** (orientée System V) gratuite et du domaine public.

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. 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 14 page 46)*.

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 13.2 page 46)*.

MIME Multipurpose Internet Mail Extension. Un standard permettant l'envoi d'accents, images, sons, etc. via courrier électronique.

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 crashmail, passerelle FTPMAIL, serveur de clés publiques, passerelle **FidoNet** et de connexion SLIP. De plus, des connexions interactives sont également possibles grâce aux **login**s guest et bbs, 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), 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 X-11. 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**.

nameserver Un serveur de correspondance entre adresse IP et nom de machine. On parle aussi de serveur **DNS**.

netmask Un netmask permet de définir un sous-réseau afin de faciliter le routage ou le **broadcast**.

netlag Mauvaise interactivité d'un réseau, bien connue des joueurs de **mud** et des utilisateurs d'**IRC**.

news Messages publics (expéditeur dans un forum pour des lecteurs, qui peuvent commenter ensuite le message et les réponses.)

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`. Egalement 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**.

PGP Pretty Good Privacy. Programme implémentant une cryptographie à clé révélée (**public-key cryptography**) permettant 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 décryptage 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 cryptage classiques). Il implémente également d'autres fonctionnalités très pratiques, par exemple, pour le test d'un logiciel par un groupe fermé d'usagers. Ce programme, du domaine public, repose sur des algorithmes cryptographiquement sûrs, comme IDEA et RSA. 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. Il est encore illégal dans certains pays comme la France, sans annoncement préalable. A cause de cela, des versions plus ou moins pirates (dans le sens pirates d'une réglementation, le programme étant libre de licence) circulent.

pipe Un mécanisme puissant de composition de commandes complexes basé sur les pipes UNIX, des tubes de données permettant de connecter entre eux des processus évitant l'utilisation de fichiers temporaires et améliorant le parallélisme.

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, est généré par des programmes de traitement de texte ou de dessin.

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 plus souple néanmoins.

regexp Regular Expression. Désigne les motifs de remplacement permettant de retrouver des fichiers ou des phrases.

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.

RNIS Réseau numérique à intégration de services, ISDN en anglais.

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 optimisé pour cela.

RPC Remote Procedure Call. Il s'agit d'un protocole 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 Fine[36] Manual).

satcom Ordinateur sous UNIX servant anciennement de passerelle mail et news pour ALPHANET. Actuellement n'est plus utilisé par le club. Voir **mips**.

serveur Logiciel s'exécutant souvent en arrière plan et traitant les requêtes de **clients**.

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`.

SLIP Protocole permettant de raccorder un ordinateur (ou un réseau Ethernet, SLIP, ...) à un autre ordinateur (ou réseau). Nous exploitons depuis mars un tel protocole sur mips.alphanet.ch qui servira de passerelle entre notre réseau local (satcom, amiga1, mips, ix) et Internet pendant les courts instants où nous serons réellement connectés à Internet.

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**.

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**).

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 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.

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, etc), 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 [10] et [9].

Teletex Extension du protocole telex pour l'utilisation avec des terminaux informatiques.

Teletexte Système monodirectionnel d'information ayant comme support les lignes inutilisées d'une émission de télévision.

---

[36] Il y a d'autres choix possibles.

TEX Un logiciel de mise en page de documents, utilisé couramment dans les publications scientifiques et techniques, développé par Donald KNUTH et décrit dans [12]. Ce document a été mis en page grâce à une extension de TEX nommée LATEX.

Token Ring Un réseau en anneau principalement utilisé sur systèmes IBM.

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**).

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.

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 les réseaux, le travail partagé et les systèmes de fenêtres distribués. Plusieurs variantes d' UNIX existent cependant, comme System V, BSD et OSF/1, maintenant une saine émulation entre les différents produits. Des standards existent afin de spécifier les interfaces nécessaires: **POSIX**, X/OPEN, etc.

Usenet Réseau informel recevant les **news**.

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, 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 CEPTEL 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 ANSI VT100, permettant l'affichage du jeu de caractère 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.

window-manager Un programme gérant le **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 **X-11** est justement d'offrir une indépendance importante de l'environnement.

WWW World-Wide Web. Système d'information distribué et multimédia basé sur le protocole **HTTP**. Plusieurs clients implémentent ce protocole, comme **Mosaic** et **lynx**.

X-11 X-11, ou X-Window, 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 X-11, 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 extrêmement bien défini, mais 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.

XDMCP Protocole permettant à un serveur X (par exemple un terminal) de rechercher tous les systèmes sur lesquels un **login** graphique est possible. Un serveur XDMCP tourne sur **mips.alphanet.ch**.

Plus d'informations

Le catalogue `/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 `/share/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.

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`).

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 dans `/users/admin/info/unix` et `/users/admin/info/RFC`.

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] G. SPAFFORD, *Practical UNIX security*, 1992, O'Reilly.

[9] J. DAVIDSON, *An Introduction to TCP/IP*, 1988, Springer-Verlag.

[10] D. E. CORNER, *Internetworking with TCP/IP*, 1991, Prentice-Hall.

[11] D. FREY & R. ADAMS, *!@%:: A directory of Electronic Mail Addressing and Networks*, 1989, O'Reilly.

[12] D. KNUTH, *The TEXbook*, 1989, Addison-Wesley.

[13] Oliver JONES, *Introduction to the X Window system, updated for X11R4*, 1987, Prentice-Hall.

[14] Tim O'REILLY, Valerie QUERCIA & Linda LAMB, *X Window system user's guide*, 1989, O'Reilly.

[15] Olivier RICOU, *Le RéZo — Histoire et guide pour le novice*, 1992, publié électroniquement sur Internet.

[16] Mark BUTLER, *Internet ... tout de suite (traduction)*, 1994, Dunod.

[17] John LEVINE & Carol BAROUDI, *Internet pour les nuls (traduction)*, 1994, Sybex.

[18] Christian HUITEMA, *Le routage dans Internet*, 1994, Eyrolles.

[19] Ed KROL, *The Whole Internet User's Guide and Catalog*, 1994, O'Reilly.

Index