From DOS/Windows to Linux HOWTO

Par Guido Gonzato, `REMOVE_MEguido@ibogfs.cineca.it` (enlevez «REMOVE_ME») ; adaptation française : Mark Hoebeke, `mh@jouy.inra.fr` .

v1.3.2, 22 Février 1999

Ce HOWTO est consacré à tous les (bientôt ex-?) utilisateurs de DOS et Windows qui ont décidé de basculer vers Linux, le clone gratuit d'UNIX. L'objectif de ce document est d'aider le lecteur à transposer sa connaissance de DOS et Windows à l'environnement Linux, et également de lui fournir des astuces sur la manière d'échanger des fichiers et des ressources entre les deux systèmes d'exploitation.

Contents

1 Introduction

1.1 Est-ce bien Linux qu'il vous faut ?

Vous voulez basculer du monde DOS (cette définition incluant Windows) vers Linux ? Bonne idée : Linux est techniquement supérieur au DOS, à Windows 9x et même à Windows NT. Mais prenez garde : cela ne vous sera peut-être pas si utile que ça. Voici les principales différences entre DOS/Windows et Linux :

- Windows peut faire tourner Microsoft Office et des tas de jeux ; est considéré comme facile à installer et configurer ; est notoirement instable ; est peu performant et sujet à des plantages fréquents.
- Linux peut faire tourner StarOffice, un flopée de logiciels techniques mais moins de jeux ; peut être difficile à installer et à configurer ; est d'une stabilité à toute épreuve ; offre des performances optimales et ne plante qu'extrêmement rarement.

C'est à vous de décider ce dont vous avez besoin. En outre, Linux vous donne la puissance mais nécessite un temps d'apprentissage pour la maîtriser. Ainsi, si vous voulez utiliser avant tout des logiciels du commerce, ou si vous ne vous sentez pas d'attaque pour apprendre quelques nouveaux concepts et commandes, vous feriez mieux de regarder ailleurs. Soyez conscient que de nombreux débutants abandonnent à cause des difficultés rencontrées au démarrage.

Des travaux sont en cours pour rendre Linux plus simple d'utilisation, mais *n'espérez pas vous y sentir à l'aise à moins de lire pas mal de documentation et de l'utiliser au moins pendant quelques mois*. Linux ne vous donnera pas de résultats immédiats. En dépit de ces avertissements, je suis confiant à 100% que si vous êtes le bon type d'utilisateur, vous trouverez en Linux votre Nirvana informatique. Au fait, Linux + DOS/Win peuvent coexister en toute harmonie sur la même machine.

Prérequis pour ce howto : je supposerai que

- vous connaissez les bases des commandes et des concepts du DOS ;
- Linux, et éventuellement le système X Window (X11 en abrégé), sont correctement installés sur votre PC ;
- votre interpréteur de commandes (ou *shell*, l'équivalent de `COMMAND.COM`) est `bash`.

Sauf indication contraire, toute l'information de ce document concerne ce vieux méchant DOS. On trouvera des informations à propos de Windows par-ci par-là, mais gardez à l'esprit que Windows et Linux sont totalement différents, contrairement au DOS qui est une espèce de parent pauvre d'UNIX.

Veuillez également noter que ce document n'est ni une introduction exhaustive ni un guide de configuration !

La version la plus récente de ce document est disponible en différents formats à l'adresse `ftp://sunsite.unc.edu:/pub/Linux/docs/howto` .

1.2 Oui, c'est bien Linux. Je veux en savoir plus.

Vous avez installé Linux et les programmes dont vous avez besoin sur le PC. Vous vous êtes créé un compte utilisateur (si ce n'est pas le cas, tapez `adduser votre_nom` *maintenant !*) et Linux est en train de tourner. Vous venez d'entrer votre nom et votre mot de passe, et, à présent, vous regardez l'écran en vous demandant : «Et maintenant, que faire ?»

Ce n'est pas le moment de désespérer. Vous êtes sur le point de faire les mêmes choses que celles que vous aviez l'habitude de faire avec DOS/Win, et beaucoup plus. Si vous étiez sous DOS/Win au lieu d'être sous Linux, vous seriez en train de faire quelques-unes des tâches suivantes :

- faire tourner des programmes et créer, copier, visualiser, effacer, imprimer et renommer des fichiers ;
- faire des CD, des MD, des RD et des DIR dans vos répertoires ;
- formater des disquettes et copier des fichiers vers ou à partir de celles-ci ;
- personnaliser le système ;
- surfer sur l'Internet ;
- écrire des fichiers .BAT et des programmes dans votre langage préféré ;
- le 1% restant.

Vous serez ravi de savoir que ces tâches peuvent être accomplies sous Linux de manière similaire au DOS. Sous DOS, l'utilisateur moyen n'utilise qu'une fraction de la bonne centaine de commandes disponibles : la même chose, jusqu'à un certain point, s'applique à Linux.

1.2.1 Concepts introductifs

La meilleure façon d'apprendre quelque chose de nouveau est de se jeter à l'eau. Il vous est chaudement recommandé d'expérimenter et de jouer avec Linux : à moins de vous connecter en tant que `root` cela ne peut en aucun cas endommager le système. Quelques points à noter :

- avant tout, comment quitter Linux en toute sécurité ? Si vous êtes devant un écran en mode texte, faites <CTRL-ALT-DEL>, attendez que le système commence à redémarrer ; à ce moment-là, éteignez le PC. Si vous travaillez sous X Windows, tapez <CTRL-ALT-BACKSPACE> d'abord, puis <CTRL-ALT-DEL>. Il ne faut *jamais* éteindre ou redémarrer le PC directement : cela pourrait endommager le système de fichiers ;
- contrairement au DOS, Linux est pourvu d'origine de mécanismes de sécurité. Des droits d'accès sont associés aux fichiers et aux répertoires ; par conséquent, l'utilisateur normal ne peut accéder à certains d'entre eux, cf. 4.3 (Droits d'accès et propriété). À l'inverse, DOS et Windows vous permettront de rayer de la carte l'intégralité du contenu de votre disque dur ;
- il existe un utilisateur particulier appelé «root» : c'est l'administrateur système qui dispose du pouvoir de vie et de mort sur la machine. Si vous travaillez sur votre propre PC, c'est vous qui êtes root. Travailler sous root est *dangereux* : toute erreur peut endommager sérieusement voire détruire le système, comme avec DOS/Win. Ne travaillez pas sous root sauf en cas de nécessité absolue ;
- une grande part de la complexité de Linux provient du fait qu'il est configurable à l'extrême : tout paramètre et toute application peuvent virtuellement être personnalisés grâce à un ou plusieurs fichiers de configuration. Cette complexité est le prix à payer pour la puissance ;
- les redirections et les *pipes* sont des capacités secondaires du DOS, mais ils sont beaucoup plus importants et plus puissants sous Linux. Des commandes simples peuvent être mises bout à bout pour accomplir des tâches complexes. Je vous encourage fortement à apprendre à vous en servir.

1.2.2 Obtenir de l'aide.

Il existe de nombreuses manières d'obtenir de l'aide avec Linux. Les plus importantes sont de :

- *lire la documentation* – J'insiste. Bien que le HOWTO que vous lisez en ce moment puisse servir d'introduction à Linux, il y a plusieurs livres que vous devriez vraiment lire : le «Linux Installation and Getting Started» par Matt Welsh (`http://sunsite.unc.edu/mdw/LDP/gs/gs.html`), le «Linux User Guide» de Larry Greenfield (`ftp://sunsite.unc.edu/pub/Linux/docs/linux-doc-project/users-guide`) , et la FAQ Linux (`http://sunsite.unc.edu/mdw/FAQ/Linux-FAQ.html`). Vous devriez vous sentir coupable jusqu'à ce que vous ayez lu au moins l'un d'entre eux.
- la documentation des paquetages installés sur votre machine se trouve souvent dans les sous-répertoires de `/usr/doc/` ;
- pour obtenir un peu d'aide sur les «commandes internes» du shell, tapez `help` ou mieux, `man bash` ou `info bash` ;
- pour avoir l'aide sur une commande, tapez `man commande` qui invoque la page du manuel (*man*) relative à `commande`. D'autre part, vous pouvez taper `info commande` qui invoque la page d'informations (*info*) traitant de `commande`. Info est un système de documentation à base d'hypertexte, dont l'utilisation n'est peut-être pas intuitive au début. Enfin, vous pouvez essayer `apropos commande` ou `whatis commande`. Pour toutes ces commandes, tapez «q» pour quitter.

- enfin, sur l'Internet : l'endroit approprié pour obtenir de l'aide est Usenet, par exemple dans news: comp.os.linux.setup . S'il vous plaît, ne m'envoyez pas de messages d'appel à l'aide car je suis quelque peu débordé.

1.3 Conventions

Tout au long de ce document, les exemples figureront souvent au format suivant : <...> est un argument obligatoire, alors que [...] est optionnel. Par exemple :

```
$ tar -tf <fic.tar> [> fic-redir]
```

fic.tar doit être spécifié, mais la redirection vers fic_redir est optionnelle.

«LPM» signifie «Lisez les Pages du Manuel pour de plus amples informations». Je ne soulignerai jamais assez combien il est important de lire la documentation. «DOSWin» signifie «DOS/Windows».

Lorsque l'invite (*prompt*) d'un exemple de commande est le symbole #, cette commande ne peut être lancée que par root.

2 Pour les impatients

Envie de vous y mettre tout de suite ? Examinez ce tableau :

```
DOS                     Linux                   Notes
------------------------------------------------------------------------------


ATTRIB (+-)attr fic     chmod <mode> fic        completement different
BACKUP                  tar -Mcvf device dir/   idem
CD repertoire\          cd repertoire/          presque la meme syntaxe
COPY fic1 fic2          cp fic1 fic2            idem
DEL fic                 rm fic                  attention, pas de undelete
DELTREE repertoire      rm -R repertoire/       idem
DIR                     ls                      pas exactement la meme syntaxe
DIR fic /S              find . -name fic        completement different
EDIT fic                vi fic                  Je ne pense pas que vous aimerez
                        jstar fic               un peu comme 'edit' sous DOS
EDLIN fic               ed fic                  n'y pensez meme pas
FORMAT                  fdformat,
                        mount, umount           syntaxe assez differente
HELP command            man command,            meme philosophie
                        info command
MD repertoire           mkdir repertoire/       presque la meme syntaxe
MORE < fic              less fic                bien mieux
MOVE fic1 fic2          mv fic1 fic2            idem
NUL                     /dev/null               idem
PRINT fic               lpr fic                 idem
PRN                     /dev/lp0,
                        /dev/lp1                idem
RD repertoire           rmdir repertoire/       presque la meme syntaxe
REN fic1 fic2           mv fic1 fic2            ne marche pas pour plusieurs fichiers
RESTORE                 tar -Mxpvf device       syntaxe differente
TYPE fic                less fic                beaucoup mieux
WIN                     startx                  a mille lieues !
```

S'il vous faut plus qu'un tableau de commandes, veuillez vous reporter aux sections suivantes.

3 À la rencontre du bash

Bonne nouvelle : avec Linux vous devez taper beaucoup moins de caractères à l'invite car l'interpréteur de commandes `bash` tape à votre place autant que possible, et dispose de fonctionnalités d'édition de ligne du tonnerre. Pour commencer, la flèche vers le haut rappelle les lignes de commandes précédentes ; mais il y a plus. L'appui sur <TAB> complète les noms de fichiers et de répertoires, ainsi, taper

```
$ls /uTABloTABbTAB
```

c'est comme taper

```
$ ls /usr/local/bin
```

S'il y a des ambiguïtés, comme lorsque vous tapez

```
$ ls /uTABloTABiTAB
```

`bash` s'arrête parce qu'il ne sait pas si vous voulez dire `/usr/local/info` ou `/usr/local/include`. Il faut fournir plus de caractères avant de taper <TAB> à nouveau.

D'autres combinaisons utiles sont <ESC-BACKSPACE> qui efface un mot sur la gauche, et <ESC-D> qui en efface un sur la droite ; <ESC-F> déplace le curseur d'un mot vers la gauche ; <CTRL-A> l'amène au début de la ligne, <CTRL-E> vers la fin. La touche <ALT> est équivalente à la touche <ESC>.

On s'arrête là pour le moment. Une fois habitué(e) à ces raccourcis, vous vous lasserez très rapidement de l'invite du DOS...

4 Fichiers et Programmes

4.1 Fichiers : notions préliminaires

La structure des répertoires et des fichiers sous Linux est très similaire à celle de DOSWin. Les fichiers ont des noms qui doivent obéir à certaines règles, ils sont stockés dans des répertoires, certains d'entre eux sont exécutables et parmi ceux-ci, la plupart disposent d'options. De plus, il est possible d'utiliser les caractères *joker*, la redirection et les pipes. Seules quelques différences mineures existent :

- sous DOS, les noms de fichier sont au format appelé 8.3 ; comme dans `TROPCOUR.TXT`. Linux peut faire mieux. Si vous avez installé Linux en utilisant un système de fichiers comme ext2 ou umsdos, vous avez le loisir d'utiliser des noms de fichier plus longs (jusqu'à 255 caractères), et avec plus d'un point : par exemple, `Ceci_est.un.TRES_long.nom.de.fichier`. Veuillez noter que j'ai utilisé aussi bien des majuscules que des minuscules : en fait...
- majuscules et minuscules dans les noms de fichier ou les commandes sont différenciées. Par conséquent, `NOMFIC.tar.tgz` et `nomfic.tar.gz` sont deux fichiers différents. De même, `ls` est une commande mais `LS` est une erreur ;
- utilisateurs de Windows, faites attention en utilisant les noms de fichier longs sous Linux. Si un nom de fichier contient des espaces (ce qui n'est pas recommandé mais néanmoins possible), vous devez l'entourer avec des guillemets à chaque fois que vous y faites référence. Par exemple :

```
$ # la commande suivante cree un repertoire appele "Mes vieux fichiers"
$ mkdir "Mes vieux fichiers"
$ ls
Mes vieux fichiers bin tmp
```

De plus, l'usage de certains caractères est à éviter : parmi eux, il y a `!*$&`.

- il n'existe pas d'extensions obligatoires comme .COM ou .EXE pour les programmes ou .BAT pour les fichiers de commandes. Les fichiers exécutables sont suivis d'un astérisque «*» lorsque vous faites la commande `ls -F`. Par exemple :

  ```
  $ ls -F
  Je_suis_un_rep/   cindy.jpg    cjpg*   lettre_a_Joe    mon_1er_script*  vieux
  ```

  Les fichiers `cjpg*` et `mon_1er_script*` sont des exécutables, autrement dit des «programmes». Sous DOS, les copies de sauvegarde se terminent en .BAK, alors que sous Linux elles se terminent par une tilde «~». De plus, un fichier commençant par un point est considéré comme étant caché. Exemple : le fichier `.Je.suis.un.fichier.cache` ne sera pas visible à la suite de la commande `ls` ;

- sous DOS, les options des commandes sont obtenues avec `/option`, Linux utilise `-option` ou `--option`. Exemple : `dir /s` devient `ls -R`. Remarquez que de nombreux programmes DOS, comme `PKZIP` ou `ARJ` utilisent les options dans le style d'UNIX.

À présent, vous pouvez aborder la section 4.4 (Traduire les commandes DOS en Linux), mais à votre place, je poursuivrais ici.

4.2 Liens symboliques

UNIX est pourvu d'un type de fichier sans équivalent sous DOS : le lien symbolique. On peut l'imaginer comme un pointeur sur un fichier ou un répertoire, et on peut l'utiliser en lieu et place du fichier ou du répertoire sur lequel il pointe ; cela ressemble aux raccourcis de Windows. Des exemples de liens symboliques sont `/usr/X11`, qui pointe sur `/usr/X11R6`; `/dev/modem`, qui pointe ou bien sur `/dev/ttyS0` ou bien sur `/dev/ttyS1`.

Pour fabriquer un lien symbolique :

```
$ ln -s <fic_ou_rep> <nomdulien>
```

Exemple :

```
$ ln -s /usr/doc/g77/DOC g77manual.txt
```

Il est alors possible de faire référence à `g77manual.txt` au lieu de `/usr/doc/g77/DOC`. Les liens apparaissent de la manière suivante dans l'affichage du contenu des répertoires :

```
$ ls -F
g77manual.txt@
$ ls -l
(des choses et d'autres...)          g77manual.txt -> /usr/doc/g77/DOC
```

4.3 Droits d'accès et propriété

Sous DOS, les fichiers et les répertoires disposent des attributs suivants : A (archive), H (caché : *hidden*), R (lecture seule : *read-only*) et S (système). Seuls H et R ont un sens sous Linux : les fichiers cachés sont ceux qui commencent par un point, et pour ce qui est de l'attribut R, lisez ce qui suit.

Sous UNIX un fichier possède des «droits d'accès» et un propriétaire, qui, à son tour, fait partie d'un «groupe». Regardez cet exemple :

```
$ ls -l /bin/ls
-rwxr-xr-x  1  root  bin  27281 Aug 15 1995 /bin/ls*
```

Le premier champ contient les droits d'accès au fichier `/bin/ls`, qui appartient à root et au groupe bin. En laissant de côté le reste de l'information, souvenez-vous que la signification de `-rwxr-xr-x`, de gauche à droite, est :

`-` est le type de fichier (`-` = fichier ordinaire, `d` = répertoire : *directory*, l = lien etc.) ; `rwx` sont les droits d'accès pour le propriétaire du fichier (lecture : *read*, écriture : *write*, exécution) ; `r-x` sont les droits d'accès pour le groupe auquel appartient le propriétaire du fichier (lecture, exécution) ; (je n'aborderai pas le concept de groupe, vous n'en avez pas besoin pour survivre, tant que vous êtes débutant ;-) `r-x` sont les droits d'accès pour tous les autres utilisateurs (lecture, exécution).

Le répertoire `/bin` dispose lui aussi de droits d'accès : cf. 5.2 (Droits d'accès aux répertoires) pour plus d'informations. C'est la raison pour laquelle vous ne pouvez pas effacer le fichier `/bin/ls` à moins d'être root : vous n'avez pas les droits pour le faire. Pour changer les droits d'accès d'un fichier, la commande est :

```
$ chmod <quiXperm> <fic>
```

où `qui` peut être `u` (utilisateur, ou propriétaire), `g` (groupe) `o` (autres, *other*), `X` vaut `+` ou `-`, et `perm` est à choisir parmi `r` (lecture, *read*), `w` (écriture, *write*) ou `x` (exécution). Les usages courants pour `chmod` sont les suivants :

```
$ chmod +x fichier
```

cela positionne le droit d'exécution pour le fichier.

```
$ chmod go-rw fichier
```

cela enlève les droits de lecture et d'écriture pour tous excepté le propriétaire.

```
$ chmod ugo+rwx fichier
```

cela donne les droits de lecture, d'écriture et d'exécution à tout le monde.

```
# chmod +s fichier
```

cela fabrique un fichier de type «setuid» ou «suid» – un fichier que tout un chacun peut exécuter avec les privilèges de son propriétaire. Typiquement, vous rencontrerez des fichiers suid appartenant à root ; souvent, il s'agit de fichiers système importants, comme le serveur X.

Un moyen plus court de faire référence aux droits d'accès est d'utiliser les chiffres : `rwxr-xr-x` peut être exprimé comme 755 (chaque lettre correspond à un bit : `---` vaut 0, `--x` vaut 1, `-w-` vaut 2, `-wx` vaut 3 etc.). Cela peut sembler difficile, mais avec un peu de pratique, vous comprendrez le concept. `root`, en tant que super-utilisateur, a la possibilité de changer les droits d'accès de quiconque. LPM.

4.4 Fichiers : traduction des commandes

À gauche, les commandes DOS, à droite, leurs correspondants Linux.

```
ATTRIB:         chmod
COPY:           cp
DEL:            rm
MOVE:           mv
REN:            mv
TYPE:           more, less, cat
```

Les redirections et la tuyauterie : < > >> |

Les jokers : `* ?`

`nul:` `/dev/null`

`prn, lpt1:` `/dev/lp0 ou /dev/lp1; lpr`

4.4.1 Exemples

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>ATTRIB +R FIC.TXT              $ chmod 400 fic.txt
C:\GUIDO>COPY JOE.TXT JOE.DOC           $ cp joe.txt joe.doc
C:\GUIDO>COPY *.* TOTAL                 $ cat * > total
C:\GUIDO>COPY FRACTALS.DOC PRN          $ lpr fractals.doc
C:\GUIDO>DEL TEMP                       $ rm temp
C:\GUIDO>DEL *.BAK                      $ rm *
C:\GUIDO>MOVE PAPIER.TXT TMP\           $ mv papier.txt tmp/
C:\GUIDO>REN PAPIER.TXT PAPIER.ASC      $ mv papier.txt papier.asc
C:\GUIDO>PRINT LETTRE.TXT               $ lpr lettre.txt
C:\GUIDO>TYPE LETTRE.TXT                $ more lettre.txt
C:\GUIDO>TYPE LETTRE.TXT                $ less lettre.txt
C:\GUIDO>TYPE LETTRE.TXT > NUL          $ cat lettre.txt > /dev/null
        n/a                             $ more *.txt *.asc
        n/a                             $ cat section*.txt | less
```

Notes:

- `*` est plus malin sous Linux : `*` désigne tous les fichiers, sauf ceux qui sont cachés ; `.*` désigne tous les fichiers cachés (mais également le répertoire courant «`.`» et son répertoire parent «`..`» : attention !) ; `*.*` ne désigne que les fichiers qui contiennent, ou qui se terminent par un «`.`» ; `p*r` désigne aussi bien «papier» que «plier» ; `*c*` désigne aussi bien «ici» que «pacha» ;
- lors de l'utilisation de `more`, tapez <ESPACE> pour parcourir le fichier, et «q» pour quitter. `less` est plus intuitif et vous permet d'employer les flèches ;
- `UNDELETE` n'existe pas, alors *pensez-y à deux fois* avant d'effacer quoi que ce soit ;
- en plus des < > >> du DOS, Linux dispose de 2> pour rediriger les messages d'erreur (stderr) ; de plus, `2>&1` redirige stderr vers stdout, alors que `1>&2` redirige stdout vers stderr ;
- Linux possède un autre joker : les `[]`. Utilisation : `[abc]*` désigne les fichiers commençant par a, b ou c ; `*[I-N1-3]` désigne les fichiers se terminant par I, J, K, L, M, N, 1, 2, ou 3 ;

- `lpr` <fic> imprime un fichier en tâche de fond. Pour vérifier l'état de la file d'impressions, utilisez `lpq` ; pour retirer un fichier de la file, utilisez `lprm` ;
- il n'existe pas de `RENAME` identique à celui du DOS ; en fait, `mv *.xxx *.yyy` ne marchera pas. Un commande du type REN est disponible à l'adresse `ftp://sunsite.unc.edu/pub/Linux/utils/file` ;
- utilisez `cp -i` et `mv -i` pour être averti lorsqu'un fichier est sur le point d'être écrasé.

4.5 Exécution de programmes : le multi-tâches et les sessions

Pour lancer un programme, entrez son nom comme vous le feriez sous DOS. Si le répertoire (cf. section 5 (Utilisation des répertoires)) où se trouve le programme est inclus dans le PATH (cf. section 8.1 (Fichiers d'initialisation du système)), le programme démarrera. Une exception : contrairement au DOS, sous Linux, un programme situé dans le répertoire courant ne s'exécutera pas à moins que ce répertoire ne soit inclus dans le PATH. *Escamotage*[1] : si `prog` est votre programme, tapez `./prog`.

Voici à quoi ressemble une ligne de commande typique :

```
$ commande [-o1 [-o2] ... [-on]] [arg1 [arg2] ... [argn]] [< entree] [> sortie]
```

où `-o1`,...,`-on` sont les options de la commande et `arg1`,...,`argn` sont les arguments de la commande. Il est possible de spécifier plusieurs commandes sur une seule ligne :

```
$ commande1 ; commande2 ; ... ; commanden
```

C'est tout au sujet de l'exécution des commandes, mais on peut facilement faire un pas de plus. L'une des principales motivations pour l'utilisation de Linux est qu'il s'agit d'un système d'exploitation multi-tâches – il est capable de faire tourner plusieurs programmes (qu'on appellera des processus) en même temps. Vous pouvez lancer des processus en arrière-plan et continuer à travailler dans la foulée. De plus, Linux vous permet d'avoir plusieurs sessions : c'est comme si on travaillait sur plusieurs ordinateurs à la fois !

- Pour passer à la session 1...6 parmi les consoles virtuelles, tapez <ALT-F1> ... <ALT-F6>
- Pour démarrer une nouvelle session dans la même console virtuelle, sans quitter la session courante, tapez `su - <nomdelogin>`. Par exemple : `su - root`. Cela peut être utile pour effectuer une tâche que seul root peut accomplir, par exemple.
- Pour clore une session, tapez `exit`. S'il reste des jobs arrêtés (cf. plus loin) vous en serez averti.
- Pour lancer un processus en arrière-plan, ajoutez une éperluette «&» à la fin de la ligne de commande :

  ```
  $ nomduprog [-options] [arguments] [< entree] [> sortie] &
  [1] 123
  ```

  Le shell associe un numéro de job au processus (ex. : `[1]` ; cf. plus loin), ainsi qu'un PID (*Process Identification Number*[2]) ; 123 dans notre exemple).
- Pour voir combien il y a de processus, tapez `ps ax`. Cela affichera la liste des processus qui tournent actuellement.

[1] NdT. : En français dans le texte.
[2] numéro d'identification du processus

- Pour tuer (*terminate*) un processus, tapez `kill` <PID>. Vous pouvez être obligé de tuer un processus lorsque vous ne savez pas comment le quitter proprement... À moins d'être root, il vous est impossible de tuer les processus d'autres personnes. Quelquefois, un processus ne pourra être tué qu'en faisant `kill -SIGKILL` <PID>.

  De plus, le shell vous permet d'arrêter ou de suspendre temporairement un processus, de l'envoyer en arrière-plan ou de le remettre au premier plan. Dans ce contexte, les processus sont appelés «jobs».

- Pour voir combien il y a de jobs, tapez `jobs`. Dans ce cas, les jobs sont identifiés par leur numéro de job, et non pas par leur PID.
- Pour arrêter un processus qui tourne au premier plan, tapez <CTRL-C> (cela peut ne pas marcher).
- Pour suspendre un processus qui tourne au premier plan, tapez <CTRL-Z> (idem).
- Pour envoyer un processus suspendu en arrière-plan, tapez `bg` <`%job`> (il devient alors un job).
- Pour amener un job au premier plan, tapez `fg` <`%job`>. Pour ramener au premier plan le dernier job envoyé en arrière-plan, il suffit de taper `fg`.
- Pour tuer un job, tapez `kill` <`%job`>, où <job>peut être 1, 2, 3,...

À l'aide de ces commandes, vous pouvez formater un disque, compresser une flopée de fichiers, compiler un programme, et décompresser et archiver le tout, tout cela en même temps, tout en gardant la main. Essayez-donc de faire cela avec Windows, rien que pour voir la différence de performance (si ça ne plante pas, bien sûr).

4.6 Exécution de programmes sur des ordinateurs distants

Pour lancer un programme sur une machine distante dont le nom est `machine.distante.edu`, il suffit de faire :

```
$ telnet machine.distante.edu
```

Après vous être connecté, démarrez votre programme préféré. Cela va sans dire, vous devez avoir un compte utilisateur sur la machine distante.

Si vous disposez de X11, il vous est même possible de faire tourner une application X sur une machine distante dont l'affichage se fera sur votre écran X. Soit `machine.distante.edu` la machine distante disposant de X, et `machine.locale.linux` votre machine Linux. Pour faire tourner à partir de `machine.locale.linux` une application X résidant sur `machine.distante.edu`, il faut faire ce qui suit :

- démarrez X11, lancez un émulateur de terminal comme `xterm` ou ses équivalents, et tapez :

  ```
  $ xhost +machine.distante.edu
  $ telnet machine.distante.edu
  ```

- après vous être connecté, tapez :

  ```
  remote:$ DISPLAY=machine.locale.linux:0.0
  remote:$ nomduprogramme &
  ```

  (au lieu de `DISPLAY...`, il se peut que vous ayez à taper `setenv DISPLAY machine.locale.linux:0.0`. Cela dépend du shell distant.)

Et voila ! `nomduprogramme` va maintenant démarrer sur `machine.distante.edu` et ses affichages se feront sur votre machine. Toutefois, il est inutile d'essayer de le faire par modem, ce serait trop lent pour être utilisable. De plus il s'agit d'une méthode fruste et non sécurisée : veuillez vous reporter au «Remote X Apps mini-HOWTO» disponible à l'adresse `http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps` .

5 Utilisation des répertoires

5.1 Répertoires : notions préliminaires

Nous avons déjà vu les différences entre les fichiers sous DOSWin et Linux. Pour ce qui est des répertoires, sous DOSWin, le répertoire racine est \, sous Linux c'est `/`. De même, les répertoires imbriqués sont séparés par \ sous DOSWin, et par `/` sous Linux. Un exemple de chemin d'accès :

```
DOS:     C:\ARTICLES\GEOLOGIE\MI_EOC.TEX
Linux:   /home/guide/articles/geologie/mi_eocene.tex
```

Comme d'habitude, `..` est le répertoire parent et `.` est le répertoire courant. Rappelez-vous que le système ne vous autorisera pas à faire des `cd`, `rd` ou des `md` n'importe où à votre convenance. Chaque utilisateur place ses affaires dans son propre répertoire appelé répertoire d'accueil (*home directory*), attribué par l'administrateur système ; par exemple mon répertoire d'accueil est `/home/guido`.

5.2 Droits d'accès aux répertoires

Les répertoires ont également des droits d'accès. Ce que nous avons vu dans la section 4.3 (Droits d'accès et propriété) (propriétaire, groupe et autres), s'applique également aux répertoires. Pour un répertoire, `rx` signifie que vous pouvez vous rendre dans ce répertoire à l'aide de `cd`, et `w` que vous pouvez effacer un fichier de ce répertoire, ou le répertoire lui-même.

Par exemple, pour empêcher d'autres utilisateurs de venir fouiner dans `/home/guido/text` :

```
$ chmod o-rwx /home/guido/text
```

5.3 Répertoires: traduction de commandes

```
DIR:          ls, find, du
CD:           cd, pwd
MD:           mkdir
RD:           rmdir
DELTREE:      rm -rf
MOVE:         mv
```

5.3.1 Exemples

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>DIR                            $ ls
C:\GUIDO>DIR  FIC.TXT                   $ ls fic.txt
C:\GUIDO>DIR *.H *.C                    $ ls *.h *.c
C:\GUIDO>DIR/P                          $ ls | more
```

```
C:\GUIDO>DIR/A                        $ ls -l
C:\GUIDO>DIR *.TMP /S                 $ find / -name "*.tmp"
C:\GUIDO>CD                           $ pwd
        inapplicable, cf. notes       $ cd
        idem                          $ cd
        idem                          $ cd  /temp
C:\GUIDO>CD \AUTRES                   $ cd /autres
C:\GUIDO>CD ..\TEMP\CORBEIL           $ cd ../temp/corbeille
C:\GUIDO>MD PROGSNVX                  $ mkdir progsnvx
C:\GUIDO>MOVE PROG ..                 $ mv prog ..
C:\GUIDO>MD \PROGS\TURBO              $ mkdir /progs/turbo
C:\GUIDO>DELTREE TEMP\CORBEIL         $ rm -rf temp/corbeille
C:\GUIDO>RD PROGSNVX                  $ rmdir progsnvx
C:\GUIDO>RD \PROGS\TURBO              $ rmdir /progs/turbo
```

Notes :

- L'utilisation de `rmdir` ne peut se faire que sur des répertoires vides. Pour effacer un répertoire et tout son contenu, utilisez `rm -rf` (à vos risques et périls).
- Le caractère «˜» est un raccourci pour le nom de votre répertoire d'accueil. Les commandes `cd` ou `cd` vous emmèneront dans votre répertoire d'accueil, quel que soit l'endroit où vous êtes ; la commande `cd /tmp/` vous emmènera dans `/home/votre_rep_d_accueil/tmp`.
- `cd -` «défait» le dernier `cd`.

6 Disquettes, disques durs et consorts

Il existe deux manières de gérer les périphériques sous Linux : à la manière du DOS, ou à celle d'UNIX. Faites votre choix.

6.1 Gestion des périphériques «à la DOS»

La plupart des distributions Linux sont accompagnées de la suite Mtools, un ensemble de commandes parfaitement équivalentes à leurs correspondants DOS, mais commençant par un «m» : `mformat`, `mdir`, `mdel`, `mmd`, etc. Ils sont même capables de conserver les noms de fichiers longs, mais pas les droits d'accès aux fichiers. En configurant Mtools, par l'édition d'un fichier appelé `/etc/mtools.conf` (dont un exemple est fourni), vous pouvez également accéder à la partition DOS/Windows, aux lecteurs de CD–ROM et de Zip. Toutefois, pour formater une disquette, la commande `format` ne fait pas l'affaire. Il vous faudra préalablement exécuter, en tant que root, la commande : `fdformat /dev/fd0H1440`

Note : il ne vous est pas possible d'accéder aux fichiers d'une disquette avec une commande de type `less a:fic.txt`! C'est l'inconvénient d'accéder aux disques à la manière du DOS.

6.2 Gestion des périphériques « à la UNIX»

UNIX aborde la gestion des périphériques d'une manière totalement différente. Il n'existe pas de volumes comme A: ou C: ; un disque, que ce soit une disquette ou n'importe quoi d'autre, devient une partie du système de fichiers local par une opération que l'on appelle le «montage». Lorsque vous n'avez plus besoin du disque, avant de l'éjecter il vous faut le «démonter».

Le formatage physique d'un disque est une chose, y créer un système de fichiers en est une autre. La commande DOS `FORMAT A:` fait les deux à la fois, mais Linux fournit des commandes séparées. Pour formater une disquette, voyez ci-dessus ; pour y créer un système de fichiers :

```
# mkfs -t ext2 -c /dev/fd0H1440
```

Vous pouvez utiliser `dos`, `vfat` (ce qui est conseillé) ou d'autres formats à la place de `ext2`. Une fois le disque préparé, montez-le avec la commande :

```
# mount -t ext2 /dev/fd0 /mnt
```

en spécifiant le type adapté pour le système de fichiers si vous n'utilisez pas `ext2`. À présent, vous pouvez accéder aux fichiers de la disquette à l'aide de `/mnt` au lieu de A: ou B:. Quelques exemples :

```
DOS                                       Linux
---------------------------------------------------------------------

C:\GUIDO>DIR A:                           $ ls /mnt
C:\GUIDO>COPY A:*.*                       $ cp /mnt/* .
C:\GUIDO>COPY *.ZIP A:                    $ cp *.zip /mnt
C:\GUIDO>EDIT A:FIC.TXT                   $ jstar /mnt/fic.txt
C:\GUIDO>A:                               $ cd /mnt
A:> _                                     /mnt/$ _
```

Lorsque vous avez terminé, et avant d'éjecter la disquette, vous *devez* la démonter avec la commande :

```
# umount /mnt
```

Bien évidemment, vous ne devez utiliser `fdformat` et `mkfs` qu'avec des disquettes non-formatées, et non pas avec celles déjà utilisées. Si vous voulez utiliser le lecteur B:, faites référence à `fd1H1440` et `fd1` au lieu de `fd0H1440` et `fd0` dans les exemples ci-dessus.

Il va sans dire que ce qui s'applique aux disquettes s'applique également à d'autres périphériques ; par exemple, vous pouvez avoir envie de monter un autre disque dur ou un lecteur de CD–ROM. Voici comment monter le CD–ROM :

```
# mount -t iso9660 /dev/cdrom /mnt
```

Ce qui précède était la manière «officielle» de monter les disques, mais il y a une astuce. Comme c'est assez pénible de passer root pour monter un disque ou un CD–ROM, chaque utilisateur peut être autorisé à les monter de la façon suivante :

- sous root, faites ce qui suit :

  ```
  # mkdir /mnt/floppy ; mkdir /mnt/cdrom
  # chmod 777 /mnt/floppy /mnt/cd*
  # # assurez-vous que le peripherique correspondant au CD-ROM est correct
  # chmod 666 /dev/hdb ; chmod 666 /dev/fd*
  ```

- ajoutez les lignes suivantes à `/etc/fstab` :

  ```
  /dev/cdrom      /mnt/cdrom  iso9660 ro,user,noauto         0       0
  /dev/fd0        /mnt/floppy vfat    user,noauto            0       0
  ```

À présent, pour monter un disquette DOS et un CD–ROM vous pouvez faire :

```
$ mount /mnt/floppy
$ mount /mnt/cdrom
```

`/mnt/floppy`, et `/mnt/cdrom` sont à présent accessibles à tous les utilisateurs. Souvenez-vous que d'autoriser tout le monde à monter des disques de cette manière est un trou de sécurité béant, si cela vous préoccupe.

Deux commandes utiles sont `df`, qui donne des informations sur les systèmes de fichiers montés, et `du nomderepertoire` qui renseigne sur l'espace disque utilisé par le répertoire.

6.3 Les sauvegardes

Il existe plusieurs paquetages pour vous aider, mais le moins que vous puissiez faire pour obtenir une sauvegarde sur plusieurs volumes est (sous root) :

```
# tar -M -cvf /dev/fd0H1440 rep_a_sauvegarder/
```

Assurez-vous d'avoir une disquette formatée dans le lecteur, et un paquet d'autres sous la main. Pour restaurer vos affaires, insérez la première disquette dans le lecteur et faites :

```
# tar -M -xpvf /dev/fd0H1440
```

7 Et Windows alors ?

Le système X Window est «l'équivalent» de Windows. À l'opposé de Windows ou du Mac, X11 n'a pas été conçu pour être facile d'utilisation ni esthétique, mais uniquement pour munir les stations de travail sous UNIX de capacités graphiques[3]. Voici les principales différences :

- Alors que Windows adopte le même *look and feel* partout dans le monde, ce n'est pas la cas pour X11 qui est beaucoup plus configurable. Son apparence générale est définie par un composant-clé appelé le «gestionnaire de fenêtres» (*window manager*) dont il existe une large gamme : `fvwm`, élémentaire mais agréable et peu gourmand en mémoire, `fvwm2-95`, `Afterstep`, `WindowMaker`, `Enlightenment` et beaucoup d'autres. Ce gestionnaire est habituellement invoqué dans un fichier nommé `.xinitrc`.
- Votre gestionnaire peut être configuré de manière à ce que les fenêtres agissent comme dans, hum, Windows : vous cliquez dessus pour les faire venir au premier plan. Une autre possibilité est de l'amener au premier plan lorsque la souris se déplace dessus (la fenêtre obtient le «focus»). De plus, le placement des fenêtres sur l'écran peut être automatique ou interactif : si, au lieu de votre programme, il apparaît un cadre étrange, cliquez avec le bouton gauche à l'endroit où vous voulez le faire apparaître.
- la plupart des fonctionnalités peuvent être personnalisées en éditant un ou plusieurs fichiers de configuration. Lisez la documentation de votre gestionnaire de fenêtres ; ces fichiers de configuration peuvent être `.fvwmrc`, `.fvwm2rc95`, `.steprc`, etc. Un fichier de configuration type se trouve généralement dans : `/etc/X11/window-manager-name/system.window-manager-name`;
- Les applications X11 sont écrites à l'aide de bibliothèques spéciales (les «widget sets») ; comme il en existe plusieurs, les applications ont des apparences différentes. Les plus élémentaires sont celles utilisant les widgets Athena (apparence 2–D; `xdvi`, `xman`, `xcalc`) ; d'autres utilisent Motif (`netscape`), d'autres encore utilisent Tcl/Tk, XForms, Qt, Gtk et en veux-tu en voilà. Presque toutes ces bibliothèques fournissent en gros le même look & feel que Windows.

[3]NdT. : Et surtout de la possibilité de déporter l'affichage sur d'autres machines au travers du réseau

- Malheureusement, le mode d'interaction peut se révéler incohérent. En l'espèce, lorsque vous sélectionnez une ligne de texte à la souris et tapez <BACKSPACE>, vous vous attendez à ce que la ligne disparaisse, pas vrai ? Cela ne fonctionne pas avec les applis basées sur Athena, mais avec celles utilisant d'autres bibliothèques de widgets.
- Le mode de fonctionnement des ascenseurs (*scrollbars*) et le redimensionnement dépendent du gestionnaire de fenêtres et du widget set. Astuce : si les ascenseurs ne se comportent pas à votre goût, essayez d'utiliser le bouton du milieu, ou les deux boutons en même temps pour les déplacer.
- Les applications n'ont pas d'icône par défaut, mais elles peuvent en avoir un certain nombre. La plupart des gestionnaires de fenêtres proposent un menu que l'on appelle en cliquant sur le fond de l'écran (*root window*) ; il va sans dire que ce menu peut être personnalisé. Pour changer l'apparence du fond de l'écran, utilisez `xsetroot` ou `xloadimage`.
- Le presse-papiers ne peut contenir que du texte, et se comporte étrangement. Une fois un texte sélectionné, il est déjà copié dans le presse-papiers : déplacez vous ailleurs et cliquez sur le bouton du milieu pour le coller. Il existe une application `xclipboard` fournissant de multiples tampons pour le presse-papiers.
- Le glisser-déplacer est une option, et n'est disponible qu'avec les applications et/ou les gestionnaires de fenêtres qui l'intègrent.

Pour économiser la mémoire, il est préférable d'utiliser des applications utilisant les mêmes bibliothèques, mais cela peut se révéler difficile à appliquer.

Il existe des projets visant à doter X11 d'une apparence et d'un comportement aussi cohérents que Windows. Gnome, `http://www.gnome.org` et KDE, `http://www.kde.org` , sont épatants. Donnez-leur une chance : vous ne regretterez plus votre bureau Windows.

8 Personnalisation du système

8.1 Fichiers d'initialisation du système

Sous DOS, deux fichiers importants sont `AUTOEXEC.BAT` et `CONFIG.SYS`, utilisés lors du démarrage pour initialiser le système, positionner quelques variables d'environnement comme PATH et FILES, et, le cas échéant lancer un programme ou un fichier de commandes. Sous Linux, il y a une flopée de fichiers d'initialisation, qu'il vaut mieux ne pas bidouiller pour certains d'entre eux à moins de savoir exactement ce que vous faites. Je vous dirai quels sont les plus importants de toute façon :

```
FICHIERS                              NOTES

/etc/inittab                          pas touche pour le moment !
/etc/rc.d/*                           idem
```

Si vous vous contentez de positionner le PATH et d'autres variables d'environnement, ou désirez changer les messages de login, ou souhaitez lancer automatiquement un programme après vous être connecté, considérez les fichiers suivants :

```
FICHIERS                              NOTES

/etc/issue                            positionne le message de pre-connexion
/etc/motd                             positionne le message de post-connexion
```

```
/etc/profile                        positionne $PATH, d'autres variables etc.
/etc/bashrc                         positionne alias et fonctions etc.
/home/votre_home/.bashrc            positionne vos alias et fonctions
/home/votre_home/.bash_profile   ou
/home/votre_home/.profile           definit l'environnement et lance vos programmes
```

Si ce dernier fichier existe (remarquez que c'est un fichier caché), il sera lu après que vous vous soyez connecté et les commandes qu'il contient seront exécutées.

Un exemple – examinez ce fichier `.bash_profile`:

```
# Je suis un commentaire
echo Environnement:
printenv | less   # equivalent de la commande SET sous DOS
alias d='ls -l'   # pour comprendre facilement ce qu'est un alias
alias up='cd ..'
echo "Je vous rappelle que le chemin est "$PATH
echo "Aujourd'hui nous sommes le `date`"  # utilise la sortie de la commande 'date'
echo "Bonjour, "$LOGNAME
# Ce qui suit est une "fonction" shell
ctgz() # Affiche le contenu d'une archive .tar.gz.
{
  for file in $*
  do
    gzip -dc ${file} | tar tf -
  done
}
# fin du .profile
```

`$PATH` et `$LOGNAME`, vous l'aurez deviné, sont des variables d'environnement. Il s'en trouve beaucoup d'autres avec lesquels on peut s'amuser ; LPM pour des applis comme `less` ou `bash` par exemple.

Ajouter la ligne suivante à votre `/etc/profile` vous fournira approximativement un équivalent de `PROMPT $P$G` :

```
export PS1="\w\\$ "
```

8.2 Fichiers d'initialisation de programmes

Sous Linux, tout peut potentiellement être personnalisé pour convenir à vos besoins. La plupart des programmes ont un ou plusieurs fichiers d'initialisation que vous pouvez triturer, souvent appelés `.nomdeprogrammerc` et situés dans votre répertoire d'accueil. Les premiers que vous aurez envie de modifier sont :

- `.inputrc` : utilisé par `bash` pour définir les touches du clavier ;
- `.xinitrc` : utilisé par `startx` pour initaliser le système X Window ;
- `.fvwmrc` : utilisé par le gestionnaire de fenêtres `fvwm` ;
- `.joerc, .jstarrc` : utilisé par l'éditeur `joe` ;
- `.jedrc` : utilisé par l'éditeur `jed` ;

- `.pinerc` : utilisé par l'outil de traitement de courrier électronique `pine` ;
- `.Xdefault` : utilisé par de nombreux programmes X Window.

Pour tous ceux-ci et d'autres que vous rencontrerez tôt ou tard, LPM. Au cas où, je vous signale l'existence du *Configuration HOWTO* disponible à l'adresse `http://sunsite.unc.edu/mdw/HOWTO/Config-HOWTO.html` .

9 Le réseau : concepts

Non seulement la connexion au réseau par modem est disponible sous Linux, mais elle est également plus stable et plus rapide. Ce dont il s'agit, c'est «PPP», le protocole employé pour se connecter à l'Internet à l'aide de modems. Vous avez besoin d'un noyau intégrant le support PPP et d'un outil pour composer les numéros et établir la connexion.

Pour récupérer votre courrier chez votre fournisseur d'accès Internet (*FAI*), vous aurez besoin d'un outil appelé «récupérateur de mail» utilisant le protocole POP ; lorsque le courrier est récupéré, tout se passera comme s'il avait été livré directement sur votre babasse Linux. Vous utiliserez ensuite un MUA (*Mail User Agent* ou agent utilisateur de courrier) comme `pine`, `mutt`, `elm` ou l'un des nombreux autres pour gérer votre courrier.

Alors que sous Windows, le programme de composition de numéros est invoqué automatiquement au lancement d'une application Internet, sous Linux, on procède dans l'autre sens : on compose d'abord le numéro, puis on lance l'application. Un truc du nom de `diald` fournit le comportement habituel. L'installation et la configuration d'un réseau avec connexion par modem était habituellement l'une des choses les plus difficiles à faire sous Linux, mais ce n'est plus le cas : veuillez vous reporter au Configuration HOWTO.

Enfin, un mot du «Voisinage réseau» : il vous est possible de faire croire aux machines Windows du réseau local que votre station Linux est un serveur Windows NT/9x. La formule magique est Samba : il ne s'agit pas de la danse brésilienne endiablée, mais d'une implémentation du protocole SMB pour Linux. Rendez-vous à `http://samba.anu.edu.au/samba` .

10 Un peu de programmation

10.1 Les *shell-scripts* : des fichiers .BAT dopés aux anabolisants

Si vous utilisiez des fichiers .BAT pour créer des raccourcis pour de longues lignes de commandes (moi-même j'en ai fait pas mal), vous pouvez atteindre ce but en insérant des lignes d'*alias* appropriés (cf. l'exemple ci-dessus) dans le `profile` ou le `.bash_profile`. Mais si vos .BAT étaient plus compliqués, alors vous adorerez le langage de *script* que vous offre le *shell* : il est aussi puissant que ce bon vieux QBasic, si ce n'est plus. Il possède des variables, des structures comme `while, for, case, if... then... else` et des tas d'autres fonctionnalités : il peut être une bonne alternative à un «vrai» langage de programmation.

Pour écrire un script – l'équivalent d'un fichier .BAT sous DOS – tout ce que vous avez à faire c'est d'écrire un fichier ASCII contenant les instructions, de le sauver, et de le rendre exécutable à l'aide de la commande `chmod +x` <`fichierscript`>. Pour l'exécuter, tapez son nom.

Un mot d'avertissement. L'éditeur du système s'appelle `vi`, et, d'après mon expérience, la plupart des nouveaux utilisateurs le trouvent très ardu. Je ne m'étendrai pas sur son utilisation, veuillez consulter le livre de Matt Welsh ou faire une recherche sur le Net. Je me contenterai de dire que :

- pour insérer du texte, tapez `i` puis votre texte ;

- pour effacer des caractères; tapez <ESC> puis `x` ;
- pour quitter `vi` sans enregistrer, tapez <ESC> puis `:q!` ;
- pour enregistrer et quitter, tapez <ESC> puis `:wq`.

Un bon éditeur pour débutants est `joe` : en le lançant avec la commande `jstar`, vous obtiendrez les mêmes raccourcis-clavier que ceux de l'éditeur de DOSWin. `jed` en mode WordStar ou IDE est même mieux. Veuillez consulter la section 11.4 (Où trouver les applications) pour savoir où obtenir ces éditeurs.

L'écriture de scripts sous `bash` est un sujet tellement vaste qu'il nécessite un livre en lui-même, et je ne creuserai pas le sujet plus avant. Je me contenterai de vous donner un exemple de shell-script duquel vous pourrez extraire quelques règles de base :

```
#!/bin/sh
# exemple.sh
# Je suis un commentaire.
# Ne changez pas la premiere ligne, elle doit se trouver a cet endroit.
echo "Ce systeme est : `uname -a`" # utilise la sortie de la commande
echo "Mon nom est $0" # variables predefinies
echo "Vous m'avez transmis les $# parametres suivants : "$*
echo "Le premier parametre est : "$1
echo -n "Quel est votre nom ? " ; read votre_nom
echo notez la difference : "salut $votre_nom" # citation avec "
echo notez la difference : 'salut $votre_nom' # citation avec '
REPS=0 ; FICS=0
for fic in `ls .` ; do
  if [ -d ${fic} ] ; then # si fic est un repertoire
    REPS=`expr $REPS + 1`  # REPS = REPS + 1
  elif [ -f ${fic} ] ; then
    FICS=`expr $FICS + 1`
  fi
  case ${fic} in
    *.gif|*jpg) echo "${fic}: image" ;;
    *.txt|*.tex) echo "${fic}: fichier texte" ;;
    *.c|*.f|*.for) echo "${fic}: fichier source" ;;
    *) echo "${fic}: fichier generique" ;;
  esac
done
echo "il y a ${REPS} repertoires et ${FICS} fichiers"
ls | grep "ZxY--!!!WKW"
if [ $? != 0 ] ; then # code de retour de la derniere commande
  echo "ZxY--!!!WKW introuvable"
fi
echo "ca suffit... tapez 'man bash' pour en savoir plus."
```

10.2 «E-C-iez» par vous-même

Sous UNIX, le langage du système est le C, que vous l'aimiez ou non. Des douzaines d'autres langages (Java, FORTRAN, Pascal, Lisp, Basic, Perl, awk...) sont également disponibles.

En partant du principe que vous connaissez le C, voici quelques lignes directrices pour ceux d'entre vous qui ont été trop gâtés par le Turbo C++ ou l'un de ses cousins sous DOS. Le compilateur C de Linux s'appelle `gcc` et n'est pourvu d'aucun des gadgets qui accompagnent en général ses équivalents DOS : pas de IDE (*Integrated Development Environment* ou environnement de développement intégré), ni d'aide en ligne ou de débogueur intégré etc. Il s'agit juste d'un compilateur en ligne de commande brut de décoffrage, très puissant et efficace. Pour compiler votre `hello.c` de base, vous ferez :

```
$ gcc hello.c
```

ce qui créera un fichier exécutable appelé `a.out`. Pour donner un nom différent à l'exécutable, faites :

```
$ gcc -o hola hello.c
```

Pour intégrer une bibliothèque lors de l'édition de liens, ajoutez l'option -l<nomdebibli>. Par exemple, pour intégrer la bibliothèque mathématique :

```
$ gcc -o progmath progmath.c -lm
```

(L'option `-l<nomdebibli>` oblige `gcc` à intégrer la bibliothèque `/usr/lib/lib<nomdebibli>.so`; ainsi, `-lm` intègre `/usr/lib/libm.so`).

Jusque-là, tout va bien. Mais lorsque votre programme se compose de plusieurs fichiers source, vous aurez besoin de l'utilitaire `make`. Supposons que vous ayez écrit un analyseur d'expresssions : son fichier source s'appelle `parser.c` et il inclut (par `#include`) deux fichiers d'en-tête, `parser.h` et `xy.h`. Ensuite, vous souhaitez utiliser les routines de `parser.c` dans un programme, disons `calc.c`, qui à son tour inclut `parser.h`. Quel bazar ! Que faut-il faire pour compiler `calc.c` ?

Vous devrez écrire un `Makefile`, c'est ainsi qu'on l'appelle, qui apprenne au compilateur quelles sont les dépendances entre fichiers source et fichiers objet. Dans notre exemple :

```
# Voici Makefile, utilise pour compiler calc.c
# Appuyez sur <TAB> aux endroits indiques !

calc: calc.o parser.o
<TAB>gcc -o calc calc.o parser.o -lm
# calc depend de deux fichiers objet : calc.o et parser.o

calc.o: calc.c parser.h
<TAB>gcc -c calc.c
# calc.o depend de deux fichiers source

parser.o:  parser.c parser.h xy.h
<TAB>gcc -c parser.c
# parser.o depend de trois fichiers source

# fin du Makefile.
```

Enregistrez ce fichier sous le nom `Makefile` et faites `make` pour compiler votre programme ; une alternative est de l'enregistrer sous `calc.mak` et de taper `make -f calc.mak`, et, bien sûr, LPM. Vous pouvez accéder à quelques aides sur les fonctions C, qui se trouvent dans les pages du `man`, à la section 3, par exemple,

```
$ man 3 printf
```

Pour déboguer vos programmes, utilisez `gdb`. Faites `info gdb` pour apprendre à vous en servir.

De nombreuses bibliothèques sont disponibles ; les premières d'entre elles qui vous seront utiles sont `ncurses` (effets en mode texte), et `svgalib` (graphismes en mode console). Si vous vous sentez assez enhardi pour aborder la programmation X11 (ce n'est pas si difficile que cela), il existe des bibliothèques qui transforment l'écriture de programmes X11 en une promenade de santé. Regardez donc du côté de `http://www.xnet.com/blatura/linapp6.html` en n'oubliant pas que Gtk devient un standard Linux.

Beaucoup d'éditeurs peuvent faire office d'IDE ; `emacs` et `jed`, par exemple, sont capables de mettre en évidence la syntaxe, faire de l'indentation automatique etc. Autrement, rapatriez le paquetage `rhide` à partir de

`ftp://sunsite.unc.edu:/pub/Linux/devel/debuggers/` . C'est un clone de l'IDE de Borland, et vous devriez l'apprécier.

11 Le 1% restant

En fait, il reste beaucoup plus que 1%...

11.1 Utilisation de tar et gzip

Sous UNIX, il y a quelques applications très répandues pour l'archivage et la compression de fichiers. `tar` sert à fabriquer des archives – comme `PKZIP` ou `Winzip` mais il ne fait qu'archiver, sans compression. Pour fabriquer une nouvelle archive :

```
$ tar -cvf <nom_d_archive.tar> <fic> [fic...]
```

Pour extraire des fichiers d'une archive :

```
$ tar -xpvf <nom_d_archive.tar> [fic...]
```

Pour examiner le contenu d'une archive :

```
$ tar -tf <nom_d_archive.tar> | less
```

Vous pouvez compresser les fichiers à l'aide de `compress`, mais il est obsolète et ne devrait plus être utilisé, ou `gzip` :

```
$ compress <fic>
$ gzip <fic>
```

qui créent un fichier compressé portant l'extension `.Z` (`compress`) ou `.gz` (`gzip`). Ces programmes ne peuvent compresser qu'un fichier à la fois. Pour décompresser :

```
$ compress -d <fic.Z>
$ gzip -d <fic.gz>
```

LPM.

Il existe également les utilitaires `unarj`, `zip` et `unzip` (compatibles avec PK??ZIP). Les fichiers portant l'extension `.tar.gz` ou `.tgz` (archivés avec `tar`, puis compressés avec `gzip`) sont aussi communs dans le mode UNIX que les fichiers .ZIP sous DOS. Voici comment examiner le contenu d'une archive au format `.tar.gz` :

```
$ tar -ztf <fic.tar.gz> | less
```

11.2 Installation d'applications

Tout d'abord : l'installation de paquetages est le travail de root. La plupart des applications Linux sont distribuées sous forme d'archives `.tar.gz` qui contiennent typiquement un répertoire portant un nom approprié où se trouvent tous les fichiers et/ou sous-répertoires. Un bonne habitude est d'installer ces paquetages à partir de `/usr/local` à l'aide de la commande :

```
# tar -zxf <archive.tar.gz>
```

puis de lire les fichiers README ou INSTALL. La plupart du temps, l'application est distribuée sous forme de code source, que vous devrez compiler pour créer les binaires ; en général, en vous contentant de taper `make` puis `make install`. Si l'archive contient un script `configure` lancez celui-ci d'abord. Bien évidemment, vous devez avoir les compilateurs `gcc` ou `g++`.

D'autres archives doivent être déballées à partir de `/` ; c'est notamment le cas pour les archives `.tgz` de Slackware. D'autres archives contiennent des fichiers mais pas de sous-répertoires – attention à ne pas mettre le souk ! Il faut toujours examiner le contenu d'une archive avant de l'installer.

Debian et Red Hat ont leurs propres formats d'archives, respectivement `.deb` et `.rpm`. Ce dernier bénéficie d'une reconnaissance de plus en plus large ; pour installer un paquetage `rpm`, tapez :

```
# rpm -i paquetage.rpm
```

11.3 Astuces indispensables

Le défilement en arrière : l'appui sur <SHIFT + PAG UP> (la touche grise) vous permet de remonter en arrière de quelques pages, en fonction de la quantité de mémoire vidéo dont vous disposez.

Le nettoyage de l'écran : s'il vous arrive de faire `more` ou `cat` sur un fichier binaire, votre écran peut se retrouver rempli de caractères bizarres. Pour y remedier, tapez `reset` même si vous n'y voyez plus rien, ou entrez cete séquence de caractères : `echo CTRL-V ESC c RETURN`.

Le collage de texte : dans la console, voyez plus loin ; sous X, cliquez et déplacez la souris en maintenant le bouton enfoncé pour sélectionner le texte dans une fenêtre `xterm`, puis cliquez avec le bouton central (ou les deux boutons à la fois si vous avez une souris à deux boutons) pour coller. Il existe également un `xclipboard`[4] (qui ne marche qu'avec du texte, hélas) ; ne vous laissez pas perturber par son très long temps de réponse.

L'utilisation de la souris : si vous avez installé `gpm`, un pilote de souris pour la console, alors vous pouvez cliquer et déplacer la souris sans relâcher le bouton pour sélectionner du texte, puis cliquer avec le bouton droit pour coller le texte sélectionné. Cela marche d'une console virtuelle à l'autre.

Messages émis par le noyau : examinez `/var/adm/messages`, ou `/var/log/messages` en étant root pour savoir ce que le noyau a à vous dire, y compris les messages de démarrage. La commande `dmesg` est également bien pratique.

11.4 Où trouver les applications

Si vous vous demandez si vous pouvez ou non remplacer votre bonne vieille application DOS/Win par un équivalent Linux, je vous propose de parcourir les principaux «entrepôts» de logiciels pour Linux : `ftp://sunsite.unc.edu/pub/Linux` , `ftp://tsx-11.mit.edu/pub/linux` , et `ftp://ftp.funet.fi/pub/Linux` . D'autres endroits idéaux pour commencer sont la «Linux Applications and Utilities Page» située à `http://www.xnet.com/blatura/linapps.shtml` , et la page «officielle» de Linux à `http://www.linux.org` .

[4] NdT. : Presse-papiers sous X.

11.5 Quelques trucs impossibles sous DOS

Linux peut faire tout un tas de choses qui étaient pénibles, difficiles ou impossibles sous DOS/Win. Voici une petite liste qui devrait vous ouvrir l'appétit :

- at vous permet de lancer des programmes à des heures spécifiques ;
- awk est un langages simple mais néanmoins puissant pour manipuler des fichiers de données (mais pas seulement). Par exemple, si data.dat est votre fichier contenant des données sur plusieurs champs,

  ```
  $ awk '$2   "abc" {print $1, "\t", $4}' data.dat
  ```

  affiche tous les premiers et quatrièmes champs de chaque ligne de data.dat dont le second champ contient «abc».
- cron est utile pour effectuer des tâches de manière périodique, à des dates et des heures spécifiques. Tapez man 5 crontab.
- file <nomdefichier> vous indique le type de nomdefichier (texte ASCII, exécutable, archive, etc.) ;
- find (cf. également la section 5.3 (Répertoires : traduction de commandes)) est l'une des commandes les plus puissantes et les plus utiles. On l'utilise pour trouver des fichiers répondant à certaines caractéristiques, et pour effectuer des actions sur ceux-ci. L'utilisation de base de find est :

  ```
  $ find <repertoire> <expression>
  ```

  où <expression> inclut les critères de recherche et les actions. Exemples :

  ```
  $ find . -type l -exec ls -l {} \;
  ```

  trouve tous les fichiers qui sont des liens symboliques et montrent ce sur quoi ils pointent ;

  ```
  $ find / -name "*.old" -ok rm {} \;
  ```

  trouve tous les fichiers correspondant au motif donné et les efface, après vous avoir demandé confirmation ;

  ```
  $ find . -perm +111
  ```

  trouve tous les fichiers dont les droits d'accès sont 111 (exécutables) ;

  ```
  $ find . -user root
  ```

  trouve tous les fichiers appartenant à root. Il y a plein d'autres possibilités – LPM ;
- grep trouve des motifs de caractères dans des fichiers. Par exemple,

  ```
  $ grep -l "geologie" *.tex
  ```

  affiche tous les fichiers *.tex contenant le mot «geologie». La variante zgrep marche avec les fichiers «gzippés». LPM ;
- **Les expressions régulières** constituent un mécanisme complexe mais fichtrement puissant pour effectuer des opérations de recherche dans un texte. Par exemple, ^a[^a-m]X{4,}txt$ correspond aux lignes commençant par un «a», suivi d'un autre caractère ne se trouvant pas dans l'intervalle a-m, suivi de de 4 ou plus «X» et finissant en «txt». Les expressions régulières sont utilisées avec des éditeurs avancés, less, et de nombreuses autres applications. man grep pour avoir une introduction ;

- script <fichier_script> envoie le contenu de l'écran dans le fichier `fichier_script` jusqu'à ce que vous lanciez la commande `exit`. Utile pour le débogage ;
- `sudo` autorise les utilisateurs à effectuer certaines tâches de root (par ex. : le formatage et le montage des disques. LPM) ;
- `uname -a` vous renvoie de l'info sur votre système ;
- `zcat` et `zless` sont utiles pour parcourir et rediriger des fichiers gzippés sans les décompresser. Par exemple :

```
$ zless fictexte.gz
$ zcat fictexte.gz | lpr
```

- les commandes suivantes tombent souvent à pic : `bc, cal, chsh, cmp, cut, fmt, head, hexdump, nl, passwd, printf, sort, split, strings, tac, tail, tee, touch, uniq, w, wall, wc, whereis, write, xargs, znew.` LPM.

11.6 Entraînement à Unix sous DOS/Windows

Le croirez-vous ? Il existe de chouettes outils fournissant un environnement de type UNIX sous DOS/Windows. L'un d'entre eux est la suite Djgpp (`http://www.delorie.com/djgpp`) pour DOS, alors que Cygnus (`http://www.cygnus.com`) est un portage plus complexe pour Win32. Les deux comprennent les mêmes outils de développement et utilitaires GNU que sous Linux ; vous n'aurez cependant ni la même stabilité ni les mêmes performances.

Si vous voulez avoir un avant-goût de Linux, essayez Djgpp. Téléchargez et installez les fichiers suivants (au moment de la rédaction de ce document, la version la plus récente est la 2.02) : `djdev202.zip, bnu281b.zip, bsh1147b.zip, fil316b.zip, find41b.zip, grep22b.zip, gwk303b.zip, lss332b.zip, shl112b.zip.`. Les instructions pour l'installation sont fournies, et vous obtiendrez de l'aide sur `news:comp.os.msdos.djgpp` .

En particulier, l'utilisation de `bash` sous DOSWin constitue une bouffée d'air frais. Pour le configurer correctement, éditez le fichier `BOOT.BAT` fourni pour l'adapter à votre configuration, mettez ensuite ces fichiers dans votre répertoire d'accueil (dans la partition Windows) en remplacement de ceux qui sont fournis :

```
# voici le  _bashrc

LS_OPTIONS="-F -s --color=yes"
alias cp='cp -i'
alias d='ls -l'
alias l=less
alias ls="ls $LS_OPTIONS"
alias mv='mv -i'
alias rm='rm -i'
alias u='cd ..'


# voici le  _bprof
if [ -f  /_bashrc ]; then
  .  /_bashrc
fi
PS1='\w\$ '
PS2='> '
CDPATH="$CDPATH: "
# trucs pour less(1)
```

```
LESS="-M-Q"                    # invite longue, silencieuse
LESSEDIT="%E ?lt+%lt. %f"      # edition de la ligne superieure
VISUAL="jed"                   # editeur
LESSCHARSET=latin1             # visualisation des lettre accentuees
export PS1 PS2 CDPATH LS_OPTIONS LESS LESSEDIT LESSOPEN VISUAL LESSCHARSET
```

11.7 Les extensions courantes et les programmes correspondants

Vous pouvez rencontrer d'innombrables extensions aux noms de fichiers. En excluant les plus exotiques (en l'occurrence les polices etc.), voici une liste de qui est qui :

- `1 ... 8` : les pages du manuel (`man`). Lisez-les avec `groff -Tascii -man <file.1>`.
- `arj` : archive fabriquée avec `arj`.
- `dvi` : fichier de sortie produit par TeX (cf. ci-dessous). `xdvi` permet de le visualiser et `dvips` le transforme en fichier Postscript `.ps`.
- `gz` : archive fabriquée avec `gzip`.
- `info` : fichier info (espèce d'alternative aux pages man). Récupérez `info`.
- `lsm` : fichier Linux Software Map. Il s'agit d'un fichier ASCII ordinaire contenant la description d'un paquetage.
- `ps` : fichier PostScript. Pour le visualiser, réupérez `gs` et, optionnellement `ghostview` ou `gv`.
- `rpm` : paquetage Red Hat. Il est possible de l'installer sur n'importe quel système à l'aide du gestionnaire de paquetages `rpm`.
- `taz, tar.Z` : archive fabriquée avec `tar` et compressée avec `compress`.
- `tgz, tar.gz` : archive fabriquée avec `tar` et compressée avec `gzip`.
- `tex` : fichier texte à fournir en entrée à TeX, un puissant système de composition de documents. Récupérez le paquetage `tex` disponible pour de nombreuses distributions.
- `texi` : fichier texinfo, peut produire aussi bien des fichiers TeX et des fichiers info (cf. `info`). Récupérez `texinfo`.
- `xbm, xpm, xwd` : fichiers graphiques.
- `Z` : archive fabriquée avec `compress`.

11.8 Conversion de fichiers

Si vous devez échanger des fichiers entre DOS/Windows et Linux, méfiez-vous du problème des «sauts de ligne». Sous DOS, chaque ligne se termine par CR/LF (plus précisément, ASCII 13 + ASCII 10), alors que Linux se contente de LF. Si vous essayez d'éditer un fichier DOS sous Linux, il est probable que chaque ligne se termine par un étrange caractère «M» ; un texte Linux apparaîtra sous DOS comme une ligne unique extrêmement longue et aucun paragraphe ne sera présent. Il existe quelques outils, `dos2unix` et `unix2dos`, pour convertir les fichiers.

Si vos fichiers texte contiennent des caractères accentués, assurez-vous qu'ils ont été saisis sous Windows (avec Notepad) et non sous DOS ; sinon tous les caractères accentués seront joyeusement mélangés.

Pour convertir des fichiers Word ou WordPerfect vers des fichiers texte pur, le sujet est un peu plus délicat mais c'est possible. Vous aurez besoin de l'un des outils qui se trouvent sur les sites CTAN ; l'un d'entre eux étant `ftp://ftp.tex.ac.uk` . Récupérez le paquetage `word2x` du répertoire `/pub/tex/tools/`, ou essayez l'un des paquetages disponibles dans le répertoire `/pub/tex/support/`. En particulier, `word2x` convertit les fichiers Word 6, alors que pour Word 7, vous aurez besoin de `mswordview` (`http://www.csn.ul.ie/caolan/docs/MSWordView.html`) qui les transforme en HTML.

11.9 Suites bureautiques gratuites

Si la conversion de fichiers ne suffit pas, vous avez le choix de vous faire les dents sur un paquetage (gratuit !) du type Microsoft Office.

La suite StarOffice est gratuite pour une utilisation personnelle. Elle est volumineuse, un peu lente, mais très bonne quoi qu'il en soit : elle offre de nombreuses fonctionnalités absentes de Microsoft Office. Elle est également capable de lire et d'écrire des fichiers Word et Excel, bien que la conversion ne soit pas toujours parfaite. Page Web : `http://www.stardivision.com` .

Un autre paquetage de bonne qualité est Corel WordPerfect, dont une édition gratuite est disponible en téléchargement. Dois-je en dire plus ? Allez la récupérer sur : `http://www.corel.com` .

12 La fin, du moins pour l'instant

Félicitations ! À présent, vous avez quelques notions d'UNIX et êtes prêt pour commencer à travailler. Souvenez-vous que votre connaissance du système est encore limitée, et qu'il est préférable d'avoir un peu plus de pratique de Linux pour l'utiliser confortablement. Mais si tout ce que vous aviez à faire était de récupérer un paquet d'applications pour commencer à travailler, ce que j'ai inclus dans ce document est suffisant.

Je suis persuadé qu'il vous plaira d'utiliser Linux et que vous continuerez votre apprentissage – comme tout le monde. Je prends également le pari que vous ne retournerez plus jamais sous DOSWin ! J'espère avoir été compréhensible et avoir rendu service à mes 3 ou 4 lecteurs.

12.1 Droits d'auteur

`linux-howto@sunsite.unc.edu`

via email.

12.2 Responsabilités

Le «From DOS/Windows to Linux HOWTO» a été écrit par Guido Gonzato, `REMOVE_MEguido@ibogfs.cineca.it` . Un grand merci à Matt Welsh, l'auteur de «Linux Installation and Getting Started» à Ian Jackson, l'auteur de «Linux frequently asked questions with answers», à Guiseppe Zanetti, l'auteur de «Linux» à toutes les personnes qui m'ont envoyé des suggestions par courrier électronique, et des remerciements particuliers à Linux Thorvalds et GNU qui nous ont apporté Linux.

Ce document est fourni «en l'état». Je me suis efforcé de le rédiger avec toute l'exactitude possible, mais l'utilisation que vous faites de l'information qu'il contient est à vos risques et périls. Je ne pourrai en aucun cas être tenu pour responsable de dommages résultant de l'utilisation de ce travail.

Tout retour est le bienvenu. N'hésitez pas à me contacter pour toute question, suggestion, critique incendiaire etc.

Profitez de la vie et de Linux,

Guido `=8-)`