

Général

man veut dire « manuel », exemple : man <commande>

man permet de visualiser les options. Exemple : man grep donne les options de grep.

Il permet de visualiser les options.

Pour chercher une commande : man -k + sujet

man x /-x permet d'aller directement à l'option de commande recherchée.

Regarder aussi dans le man : « voir aussi... »

Caractères « joker » :

remplace une suite de n'importe quel caractère

ex. : abcde = *e = a* = a*e

ls a*

ls /home /u*

? remplace 1 caractère quelconque non vide

ex. : a ? ? ?

<< ? >> a ?* ? ?* * ?

[abcde] = n'importe quel caractère de la liste

[abcdef]

[a-f]

[0--9]

ls a* b* c*

ls [abc]*

ls [a-k,y,z]*

ls [a-j,l-z]*

ls résultat.2000.0[1-9]

ls résultat.2000.1[012] *ou* [0-2]

pour lister tous les mois de l'année.

Pour compter : wc

wc = word count

wc -c : compte les caractères avec espaces

wc -w : compte les mots

wc -l : compte les lignes

Sortie après Ctrl D

La commande **set -x** montre ce que va exécuter le système.

Pour revenir à l'état initial, set +x

Quand apparaît en début de ligne « > », c'est que la commande n'est pas terminée, le système attend.

Les caractères *apostrophes* neutralisent tout mécanisme de substitution.

Les caractères *guillemets* neutralisent uniquement les mécanismes de type « fichier ».

Le mécanisme `` (Alt GR + 7) montre le résultat d'une commande.

sort : réaffiche par ordre alphabétique ce que l'on tape.

En Korn shell (**ksh**), on peut remplacer \$HOME par ~ (Alt Gr + 2). ~ se lit Tilda.
Le fait de taper ksh lance un nouveau shell que l'on supprime avec exit.

Pour aller chercher l'adresse d'une commande x, faire **which** x.

Mettre une commande en arrière / avant-plan :

fg = foreground

bg = background

Pour remettre une commande en avant-plan : fg si une seule, sinon :
fg %n ou n est le n° entre [] que l'on voit dans ps -l

Pour mettre en arrière-plan : bg %n
Ctrl + z met en arrière-plan et fige l'application.

cd fait par défaut un cd \$HOME

Variables d'environnement 7.4 :

PATH indique les répertoires à consulter pour trouver les commandes. Les répertoires sont séparés par des « : ».

more permet de voir les commandes, écran par écran.

Ex. : ls /usr/bin | more

banner

banner fich1 affiche fich1 en bannière.

Contenu d'un répertoire 3.7

ls

Pour voir le répertoire bin page par page, faire « ls -l | more » et à l'inverse : « ls -l | less »

ls /usr/bin indique toutes les commandes

ls /usr/bin | more, donne la même chose, mais écran par écran.

Ls a* b* c*

ls [abc]*

ls [a-k,y,z]*

ls [a-j,l-z]*

ls résultat.2000.0[1-9]

ls résultat.2000.1[012] *ou [0-2]*

pour lister tous les mois de l'année.

Quand on fait un ls -l :

- s'il s'agit d'un fichier, le premier caractère des droits est un «-«
- s'il s'agit d'un répertoire ou directory, le premier caractère des droits est un «d »

ls *f donne tous les fichiers commençant par f

ls fic* : fait apparaître tout ce qui commence par fic + n'importe quoi.

ls fic ? : Idem mais plus un seul caractère.

ls fic[...] : Idem avec en plus un des caractères entre crochets.

Ce sont des caractères pour indiquer des motifs de types « fichier ».

4.4 cat

Concaténer (cat)= mettre bout à bout

cat fichier1 = cat <fichier1 = prendre fichier1 comme entrée standard = concaténer à partir de ce fichier là.

cat fichier1 >fichier2 = fichier 1 recopié dans fichier 2

\$cat (seul) :

Le système attend un texte que l'on termine en faisant Ctrl D

cat .profile pour voir son compte perso.

On met dans le .profile ce qui est perso et on y fait les modifications.

head et tail 4.6

Voir les x premières lignes d'un fichier : `head -x`

Voir les x dernières lignes d'un fichier : `tail -x`

Voir tout sauf les x premières lignes d'un fichier : `tail -x`

Exemple : ne pas voir les 5 premières lignes, ni les 5 dernières sur un fichier « secret » de 20 lignes :

```
tail +5 secret | head -10
```

Voir alors, le nombre de lignes, rajouter `| wc -l`, donc cela fait la commande :

```
tail +5 secret | head -10 | wc -l
```

Résultat = 10

Find 4.8

find permet de trouver des noms de fichiers. Avec :

- name
- size
- atime
- ctime
- mtime
- print
- exec
- ok
- type

Recherche d'un fichier commençant par m ou f :

```
find $HOME \( -name 'm' -o -name 'f*' \) -print
```

↳ o = or (ou)

ou : `find $HOME \(-name 'm*' -o -name 'f*' \) -print`

ou : `find ~ -name «fich4» - exec ls -l {} \;`

Les accolades représentent ce qui a répondu dans la première partie. Ici, fich4.

Manipulations de fichiers Chapitre 4

Motifs de type « expression régulière »

L'interpréteur de commande fait l'interprétation. Il examine d'abord s'il y a des mobiles de type « fichier » que l'on peut remplacer. C'est systématique.
Pour le neutraliser, il faut délimiter par des « « ou des ' '.

grep 4.11

grep = global result expression

* est un facteur de répétition, exemple : b*

[...] est l'indétermination sur 1 caractère.

a[123]*b peut-être égal à ab, puisqu'il y a l'étoile.

Ex. : grep uni* secret
 ↙ ↘
grep unique1 unique2 secret

Comme il trouve deux fichiers commençant par uni, il comprend qu'il doit aller chercher les lignes où l'on trouve *unique1* dans *unique2* et *secret*.

C'est pourquoi il faut noter comme suit : 'uni*'

Délimiter la commande grep avec des ' '.

« grep » recherche une chaîne de caractères dans un fichier.

ex. : grep <chaîne> <fichier>...

ex. : cat telephone | grep Durand

wc = word count (l ligne)

wc -l telephone donne le nombre de lignes de téléphone.

cat telephone | grep Durand | wc -l —————> donne le nombre de lignes de téléphones avec Durand.

cat telephone | grep Durand | grep 04 | wc -l —————> donnerait les abonnés commençant par 04, mais ce n'est pas exact, car cela va donner plusieurs lignes avec un abonné qui aurait plusieurs fois 04 dans son n° de téléphone.

Les options peuvent être jointes. Exemple :

grep -n secret

grep -v secret

sont égales à : grep -nv secret

ou : grep -vn secret

ou : grep -v -n secret

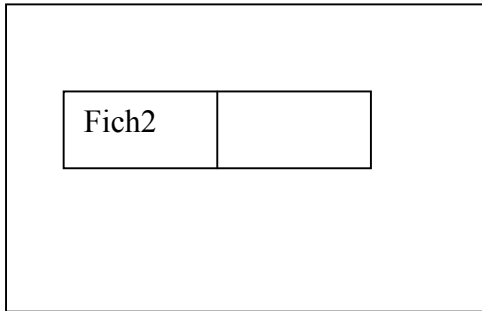
ou : grep -n -v secret

man permet de visualiser les options. Exemple : man grep donne les options de grep.

grep	expression recherchée	fichier
grep	abc	toto
grep	^abc	toto (début de ligne)
grep	^aa*	toto
	(a en premier + un nombre quelconque de a derrière, ou autre, ou rien)	
grep	^aa*\$	toto
	(les lignes avec que des a ou un seul)	
grep	^a.b\$	toto
	(commence par a, n'importe quoi derrière, et fini par b)	

Les droits d'accès aux fichiers selon les entités concernées 5.1

Mon rep



I – nodes

UID	GID	Droits	3	Type	Nombre	Emplacements
Propriétaire	Groupe	accès	dates		de liens	physiques
Description du fichier						

Droits d'accès :

Pour modifier les droits : `chmod`

Exemple : le fichier 1 est à 644 avec `rw-r--r--`

Pour le passer à `rw-rw-r--`, c'est à dire à 664, faire : `chmod 664 fich1`

On peut également :

- pour ajouter w à group : `chmod g + w fich1`
- pour ajouter x à tous : `chmod ugo + x fich1`

u = user
g = group
o = others

User 6			Group 4			Others 4		
1	1	0	1	0	0	1	0	0
R	W	-	R	-	-	R	-	-

Calcul des droits :

$$\begin{array}{|c|c|c|} \hline 1 & 1 & 0 \\ \hline \end{array}$$

$2^2 \quad 2^1 \quad 2^0$

$$\text{Répertoire} = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline \end{array} = 777$$

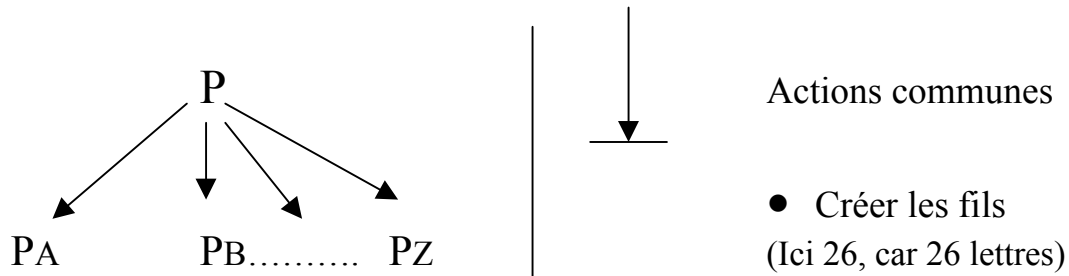
$2^2 \quad 2^1 \quad 2^0$

Les processus 6.3

Pour créer un nouveau processus, il faut avoir de la place dans la table des processus, même s'il reste de la RAM.

Le processus est le programme + les données associées. Une application est un ensemble de processus qui s'enchaînent ou qui travaillent en même temps, de façon interactive ou non.

Les processus père et fils :



Etat des processus :

Peuvent être sleeping (s)

Peuvent être Running (r)

pp = Processus père

pri = Plus le nombre est faible, plus il est prioritaire. Plus il dure dans le temps, moins il le devient.

Le scheduler alloue du temps CPU aux processus.

La table des processus contient les données dynamiques du processus. Peut demander plus de mémoire pour ses calculs au système. Elle est donc de taille variable.

Redirection d'un fichier standard 6.5

Commande 1 > toto 2 > toto

Equivalent à :

Commande 2 > &1 1 > toto

Si l'on met juste « commande », c'est sous-entendu la commande 0

Exemple de programme

```
cat > prog02                    ( > = redirection de sortie )  
while true    ───────────▶ toujours vrai  
do  
echo bonjour ───────────▶ écrire bonjour  
sleep 1      ───────────▶ arrêter une seconde  
done        ───────────▶ et recommencer  
^D
```

Pour l'exécuter : \$ prog02

En arrière plan : \$ prog02 &

Commande 2>&1 : le « et » est là pour indiquer au shell qu'il s'agit de la sortie 1, et non d'un fichier nommé 1.

kill 6.10

La commande `kill + n°` de processus autorise le processus à refuser la terminaison pour s'arrêter proprement. `kill` seul ou `kill -15` est la même chose.

`kill -9 pd`
`kill -kill pd`

Avec ces commandes, le noyau fait terminer le processus sans ménagement.

7.3 Les alias

C'est la gestion des variables locales à un shell ou variables d'environnement, communes à plusieurs shell.

set permet de voir toutes les variables locales à un shell.

La commande **set -x** montre ce que va exécuter le système. Pour revenir à l'état initial, **set +x**

echo \$toto indique à quoi est égal la variable toto.

unset toto supprime la variable toto.

env donne les variables d'environnement, communes à plusieurs shell.

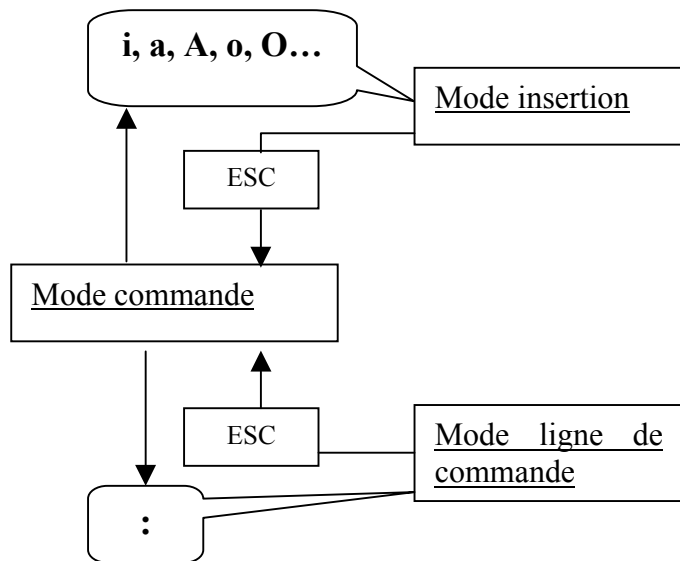
export envoie une variable locale dans un environnement qui sera valable pour tous les processus fils.

exit pour sortir du dernier shell.

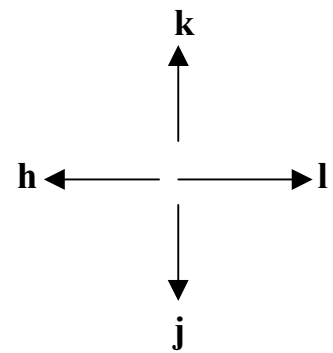
term = terminal

8.2 L'éditeur de textes vi

vi signifie video input ou vidéo inverse.



Pour circuler, touches :



h, j, k, l = déplacements curseur

`$HOME/.exec` peut-être défini pour rentrer des options de vi.

9.4 La communication en ligne

Donner un message d'invitation à la discussion

Talk utilisateur [@ machine]



login



Nom adresse IP

Pour sortir, Ctrl + c ou clear

12.3 Sauvegarde et restauration

tar 12.3

Périphérique de création d'archive de sauvegarde sur /der/rst16

```
cd /usr/local
tar cvf /der/rst16 .          ( . = pour tous )
tar cvf $HOME/archive.tar .
tar uvf $HOME/archive.tar .
```

Conseil : Faire les liens symboliques avec un chemin absolu. Partir de : « / ».

Sinon, utiliser tar -h

Crée une copie du fichier. Suit d'abord le lien jusqu'au bout, et le met dans archive.

Inconvénient : création de copie supplémentaire. Quand on modifiera dans ce cas, on risque d'oublier de modifier l'un ou l'autre.

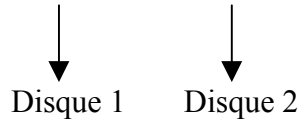
dd 12.7

Copie de disquette sous Linux:

Sur le disque : dd if = /dev/fd0 of = image

Sur la nouvelle disquette : dd if = image of = /dev/fd0

Idem d'un disque sur l'autre : dd if = /dev/hd0 of = /dev/hdb



13.2 Sauvegarde et restauration

at 13.2

Soumission différée:

at tomorrow –1hour

at > commande (éventuellement plusieurs, une par ligne)

Ctrl + D

at est utilisée pour opération ponctuelle.

cron 13.5

Utilisée pour opérations habituelles.

Exemple : sauvegarde tous les lundis du mois :

15 0 1-7 * 1

1^{er} et 3^{ème} lundis : idem, sauf 3^{ème} champ : 1-7, 15-21