

Aide mémoire Unix

Michel Meynard

21 mars 2006

1 Introduction

Système d'exploitation multi-Utilisateurs, multi-tâches.

Ouverture de session :

Login *nomutilisateur* **Password** *motdepasse*

Un interpréteur de commandes (cmd) nommé "shell" parmi plusieurs possibles : Bourne shell "sh", C-shell "csh", Korn shell "ksh"... Les cmds internes sont incluses dans le shell, les cmds externes sont des fichiers exécutables recherchés dans des répertoires privilégiés listés dans la variable **PATH**. Chaque processus a 3 fichiers standards d'E/S et d'Erreurs associés au terminal. Toute cmd renvoie un code de retour (0 si pas de problème) dans la variable **status**. Le message d'invite, "prompt", est le % (csh) ou \$ (sh). Fin de session par menu, ou logout ou exit. Pour exécuter une cmd : % *nomcmd options params*. Les options sont introduites par - (signe moins). Le manuel "on-line" est obtenu par : % **man** *nomcmd [numsection]*. Avec Xman, les sections accessibles sont : sections 1 cmds, 2 appels systèmes, 3 fonctions de bibliothèque,...

2 Système de Gestion de Fichier

2.1 Désignation des fichiers, répertoires et périphériques

Les noms de fichier ou rép. sont composés de car. qcqs, minus≠MAJUS

/ rép. racine "slash"

. rép. courant "working directory" (wd)

.. rép. père du wd

/home/lic/dupont chemin d'accès *absolu*

mesprog/prg.c chemin d'accès *relatif* au wd

Expressions régulières

[coh] ou [0-9] : 1 car. parmi un ensemble

[^0-9] : 1 car. sauf un chiffre

* suite qcq de 0 à n car. saul le point (.) **initial**

? 1 car. qcq saul le point (.) **initial**

*list ?[5-9].c exemple "matchant" avec liste6.c ou alist.5.c

Périphériques (pilotes)

/dev/tty0 fichier lié à un périphérique caractère (terminal)

/dev/rz4 fichier lié à un périphérique bloc (partition disque)

/dev/null fichier de test lié à aucun périphérique

2.2 Commandes de base

pwd affiche rép. de travail

ls *rep* liste le contenu de *rep*

ls -Ralit liste du wd "Recursive" récursivement de ses sous rep.; "all" même ceux commençant par un point (.); "long" format long; "i-node" numéro d'"i-node"; "sort" trié par date de dernière modif.

cd *rep* change le wd en *rep*

cd revient au rep d'accueil (var HOME)

mkdir *rep* création de *rep*

rmdir *rep* destruction de *rep* s'il est vide

cat *fic1 fic2 ... ficn* concaténation et affichage sur la sortie standard

less *fic1 fic2 ...* affichage page par page avec commandes : <Espace> page suivante, b page préc., <Entrée> ligne suiv., /motif rechercher exp. régulière, n recherche suivante, ?motif recherche exp reg en arrière, 35g goto ligne 35, g goto début ...

more *fic1 fic2 ... ficn* affichage page par page sur la sortie standard

cp *source destination* copie du fichier source vers destination

cp -R *reps repd* copie récursive du répertoire *reps* vers *repd*

mv *anciennom nouvnom* renomme un lien (fic ou rep) en le déplaçant de répertoire éventuellement.

rm *fic* destruction du lien *fic* et si c'était le dernier lien, destruction du fichier

ln *anciennom nouvnom* création d'un nouveau lien *nouvnom*

ln -s *anciennom nouvnom* création d'un nouveau lien symbolique

du -k *rep* affiche le contenu total en "Ko" du répertoire récursivement (-s pour éviter la récursivité)

2.3 Droits d'accès

chmod [*ugoa*] [+|=] [*rwxt*] *lien* chgt droits d'accès du fic ou du rep : *user ou group ou other ou all*; ajout, suppression, affectation; *read ou write ou eXecute ou set-user-id ou sTicky*

chmod *o-r,i+x* *monscript* exemple de changements multiples

chown *utilisateur fic* chgt propriétaire fichier

chgrp *nouvgroupe fic* chgt groupe d'un fichier

groups [*utilisateur*] liste des groupes de l'utilisateur

2.4 Configuration de l'Utilisateur

passwd, **yppasswd** modification de son mot de passe

chsh -s /bin/bash *dupont* chgt du shell d'accueil de *Dupont*.

newgrp *groupe* chgt du groupe courant de l'utilisateur

chfn *dupont* chgt des informations (finger) concernant *Dupont* : nom, bureau, tel bureau, tel dom

2.5 Filtres de fichiers

Il peuvent être utilisés comme commandes en les suffixant par un nom de fichier ou bien en série par l'intermédiaire de tube "pipe" |.

cat concaténation

more page à page
fgrep *chaîne* recherche du mot *chaîne*
grep '`^[0-9][0-9]*$`' recherche de motif défini par une expression régulière très limitée ...
egrep '`^[0-9]+(\.[0-9]*)?$`' recherche des lignes contenant un littéral numérique
wc [*lwc*] compte les lignes, mots, car.
uniq suppression lignes doubles
head *-10* copie les 10 premières lignes
tail *+10l* copie toutes les lignes à partir de la 10 ème
tail *-10c* copie les 10 derniers car.
paste - - concaténation verticale : 1ère ligne \t 2ème ligne \n 3ème ligne \t 4ème ligne \n ...
cut *-d' :'* *-f1,3,5* projection des champs 1,3,5 délimités par " : "
tr *a-z A-Z* traduit les minuscules en MAJUSCULES
tr *-s " " " "* supprime les espaces multiples
crypt cryptage
sort tri selon certains critères
dd *if=org of=dest conv=ebedic* copie et conversion (convertit org d'ascii en EBCDIC dans dest) ou *dd if=boot.img of=/dev/fd0 bs=1440k*
pr filtre de formattage ascii vers ascii (pagination, multi-colonne, ...)
psf *fic* filtre ascii vers postscript (options nombreuses)

2.6 Rechercher de fichiers et de commandes

locate fichier localise tous les endroits où se trouve ce fichier
which cmd indique où se trouve la cmd grâce au PATH
whereis cmd recherche la cmd dans certains répertoires
find *rep -name "*.c" -print* recherche récursive selon certains critères et exécution de commandes sur les fichiers trouvés
find *rep -name "*.c" -exec cat {} ";"* ou encore *find ~ \(-name '*.aux' -o -name '*~' \) -atime +3 -exec rm -f {} \;*

2.7 Compression et archivage

tar *cf backup rep* archiveur récursif du répertoire rep dans le fichier backup. *tar czf backup rep* archive et gzippe!
zip compresseur compatible PKZIP (extension .zip). Décomp. **unzip**. Ex : *tar cf - . | zip backup -* archive tous les fichiers de . dans backup.zip. *unzip -p backup | tar xf -* restaure.
gzip compresseur GNU. Pour zipper : *tar cf - . | gzip > backup.gz*. Pour dézipper : *gzip -dc backup | tar xf -*

2.8 Utilitaires

echo *message* affiche le message sur la sortie standard
basename */usr/vis/dupont* affiche dupont
dirname */usr/vis/dupont* affiche /usr/vis
split *-b 1400k fic* découpe régulière, par blocs de 1400 octets, de fic dans les fichiers xaa, xab, ...
split *-5 fic nouvfic* découpe horiz régulière, 5 lignes par 5 de fic dans les fichiers nouvficaa, nouvficab, ...
cmp *fic1 fic2* comparaison de 2 fichiers
diff *fic1 fic2* différences entre 2 fichiers

od *-x fic* dump fichier hexa (x) ou ascii (c)

2.9 Montages de systèmes de fichiers

Certains systèmes de fichiers peuvent être montés par l'utilisateur (disquette, CD-ROM, zip, ...). Pour voir lesquels : **more** /etc/fstab.

mount pour voir ce qui est monté.

mount *repmontage* permet de monter le système de fichier correspondant au répertoire de montage "repmontage". Par exemple, **mount** /mnt/floppy ou **mount** /mnt/cdrom.

umount *repmontage* permet de démonter le système de fichier.

2.9.1 Disquette et Mtools pour PC

Les Mtools sont des commandes permettant de manipuler des systèmes de fichiers non montés.

eject éjecter la disquette

mformat a : formate la disquette (boot, fat, dir)

mkdosfs /dev/fd0 formate la disquette à bas niveau

mdir ou **mdir a** : liste le répertoire

mcopy *.c a : copie les sources c

mdel fic détruit un fichier sur a :

3 Processus et jobs

Les pus sont identifiés par un numéro global à la machine (pid), les jobs par un numéro local au csh (job). On peut aussi désigner un job par %préfixe de la commande.

ps liste des pus contrôlés par ce terminal. Option l (Long).

ps faux liste hiérarchique (Forest) complète des pus contrôlés par un terminal (All) avec leurs noms d'utilisateur (User) y compris les processus sans terminal de contrôle (X).

pstree liste des pus sous forme arborescente

top équivalent de ps itératif classant les pus par ordre décroissant de consommation de CPU.

kill pid envoi d'un signal de terminaison au pus

kill job envoi d'un signal de terminaison (TERM) au pus

kill -9 pid/%job envoi d'un signal de meurtre (KILL) au pus (ou au job)

kill -l liste des signaux

cmd1 & détachement de pus & en arrière-plan et création d'un job

jobs voir la section Shell

sleep 80 attente active pendant 80 secondes

3.1 Caractères de contrôle

<Ctrl><Z> arrêt (Suspend) du processus courant

<Ctrl><D> EOF caractère indiquant la fin de fichier

<Ctrl><C> signal d'interruption envoyé au pus courant

4 Formater et Imprimer

export PRINTER=reve variable d'environnement indiquant l'imprimante par défaut

lpr -Pimp fic impression de fic sur l'imprimante imp

lpq -Pimp examen du spool d'impression d'imp

lprm -Pimp [3] suppression de job du spool (éventuellement numéroté)

lpstat -a noms et états de “toutes” les imprimantes accessibles

lpstat -pimp état de imp

pr ou **atp** ou **a2ps** *fic1 fic2* ... filtre de mise en page et formattage d'un fichier ascii en ascii (options nombreuses)

dvips -c2 *fic* convertit *fic.dvi* en postscript et l'imprime en 2 exemplaires

dvips -o *fic.ps fic* convertit *fic.dvi* en postscript dans le fichier *fic.ps*

psf *fic* ascii vers postscript (options nombreuses).

enscript *fic* ascii vers postscript (options nombreuses).

fixmacps < *fmac.ps* > *fic.ps* traduit un postscript de Mac Word vers un postscript normal

fixwfwps < *fpc.ps* > *fic.ps* traduit un postscript de Word For Windows vers un postscript normal.

psnup -4 < *fic.ps* | *lpr* impression en 4 pages par feuille a4

psselect -p1-10 < *fic.ps* | *lpr*

ghostview & visualisateur postscript

xdvi & visualisateur dvi

5 Commandes diverses et configuration

date date et heure

datef date et heure en français

at *heure jour fic* lancement différé d'un script *fic*; ex : le 15/03 à 18h lancer *monscript* : *at 18 mar 15 monscript*

crontab -l liste des commandes à exécuter régulièrement

crontab -e édite les commandes à exécuter régulièrement

.Xdefaults configuration du terminal X

.xsession script de début de session xdm

.emacs,.exrc, ... configurations d'emacs, de vi, ...

6 Langages de commande : les Shells

bash Bourne Again SHell */bin/bash* est une modernisation de sh réalisé dans le cadre du projet GNU.

tcsh Tenex C-SHell */bin/tcsh*, du nom de son auteur est une version améliorée de csh dont le nom provient de sa syntaxe à la C.

ash, ksh sont d'autres interprètes de commandes.

6.1 Documentation

man bash 5371 lignes! Utiliser la recherche : */motif* puis n.

xman & l'aide en ligne avec un index.

emacs (manual-entry) l'aide dans un éditeur de texte.

/usr/doc/tcsh-xxx/FAQ Foire Aux Questions ...

bash en mode Info sous Emacs (<Ctrl>-h i)

6.2 Quelques trucs utiles

- La touche TAB ou ESC complète automatiquement le nom du fichier ou de la commande que l'on est en train de taper;

- Les touches flèches vers le haut ↑ et vers le bas permettent de faire défiler les commandes précédemment tapées. Il

reste à les éditer puis à taper Entrée n'importe où dans la ligne pour exécuter la nouvelle commande.

- Le nom de la commande (1er mot) est tout d'abord recherché dans les commandes internes de l'interpréteur puis dans les répertoires du PATH.

7 Généralités communes

7.1 Gestion des Processus et des jobs

cmd > *fic* redirection d'une cmd en sortie vers *fic*

cmd < *fic* redirection d'une cmd en entrée depuis *fic*

cmd >> *fic* redirection d'une cmd en sortie et concaténation à *fic*

cmd > & *fic* redirection d'une cmd en sortie et sortie d'erreur vers *fic*

cmd1 ; cmd2 séquence de cmds

cmd1 | cmd2 tube de la sortie standard de *cmd1* vers l'entrée standard de *cmd2*

cmd1 |& cmd2 tube de la sortie standard et d'erreur de *cmd1* vers l'entrée standard de *cmd2*

cmd1 && cmd2 *cmd2* conditionnée par la réussite de *cmd1*

cmd1 || cmd2 *cmd2* conditionnée par l'échec de *cmd1*

cmd1 & détachement de *pus* en arrière-plan et création d'un job

jobs liste des jobs. Option -l pour avoir les pid

suspend arrêt du job courant. Possible avec stop %job

fg mise en avant-plan du job courant et remise en route. Ou plus simplement %job

fg %2 mise en avant-plan du job 2 et remise en route

bg %2 mise en arrière-plan du job 2 et remise en route

wait attendre la termin. des *pus* fils en arrière-plan

<Ctrl><Z> arrêt (Suspend) du processus courant

<Ctrl><D> EOF caractère indiquant la fin de fichier

<Ctrl><C> signal d'interruption envoyé au *pus* courant

7.2 Les variables

Chaque shell a un jeu de variables locales (minusc.), chaque processus a un jeu de variables d'environnement, exportables (MAJUSC.). Un nom de variable commence par une lettre. La substitution du nom de variable par sa valeur est obtenue en le préfixant par dollar : **\$var** ou mieux **\${var}**. Les commandes de base communes à tous les shells :

env liste des variables d'environnement

set liste des var locales

echo \${var} affiche la valeur de var

7.2.1 Variables d'environnement prédéfinies

PATH liste des répertoires de recherche des commandes (fichiers exécutables). Liste séparée par des “:”. Le répertoire courant “.” ne fait pas toujours partie du PATH!

HOSTNAME nom de la machine

DISPLAY référence du terminal X de connection (ncdd16 :0.0 ou :0.0)

TERM type de terminal (xterm, vt100, ...)

HOME répertoire d'accueil

USER nom de l'utilisateur (login)

SHELL shell d'accueil

TERMCAP configuration du terminal courant (voir stty)

7.2.2 Variables locales prédéfinies

~ répertoire d'accueil
~ *nomutilisateur* répertoire d'accueil de ...
\$\$ pid du pus courant
\$! pid du dernier pus détaché
\$? code de retour de la dernière commande (0 : OK)

7.3 Substitutions dans la ligne de commande

Un certain nombre de substitutions sont possibles :

*?[] [^] substitution d'exp reg par des noms de fichiers
<|&;\n\t\r redirections et caractères spéciaux
\$var valeur de var
'cmd' substitution par la sortie standard de cmd
alias un alias est un diminutif d'une commande complexe
mot1 mot2 ... \n commentaire (ignoré)

Pour éviter leur interprétation, ils doivent être précédés par \ ou dans une chaîne encadrée par ' (quote, apostrophe) ou " (double quote, guillemets).

* est le caractère ASCII étoile

"mot1 mot2" annule les substitutions sauf \, \$, '
'mot1 mot2' annule toute substitution

7.4 Scripts

Un script est un fichier lisible contenant une suite de commandes. S'il est exécutable et commence par #!*interpréteur*, il peut être directement lancé par son nom sinon il pourra être exécuté par *interpréteur* script.

#!/bin/tcsh première ligne d'un script
\$0 nom du premier mot de ligne de commande (fichier exécutable)
\$1 ... \$9 arguments du script
shift supprime \$1 et décale les autres argument à gauche
\$# nombre d'arguments

7.5 Commandes internes utiles

source fic exécute le script fic dans le shell courant
exit quitte le shell courant
cd, echo, exec, eval ...

8 Tcsh

8.1 Variables

Toutes les variables sont des chaînes, ou des tableaux de chaînes. Le caractère @ indique que l'on veut faire des calculs numériques.

8.1.1 Variables locales tcsh

Attention, la syntaxe est très stricte : notamment en ce qui concerne les séparateurs, espace(s) et tabulation(s) notés ci-dessous par <S> et les retours à la ligne notés \n.

set<S>var<S>=<S>motbonjour définition et affectation d'une valeur mot à une variable var
set<S>var<S>=<S>'mot1<S>mot2 ...' suite de mots
set var = "mot1 mot2 ..." suite de mots

set tab = (mot1 mot2 ...) tableau de mots (indices de 1 à n)

echo \$tab[2] affichage du deuxième mot du tableau

set tab[2] = mot affectation à l'indice 2

set tab[2-4] = (mot1 mot2 mot3) affectations

shift var décalage gauche de la variable tableau

@ var = exp variable numérique : += et -= sont également possibles. L'expression exp doit être espacée voire parenthésée.

@ tab[2] += 5 - 4 ajoute 1 au deuxième composant du tableau : il faut avoir initialisé cet élément avec un nombre!

set varbool variable pseudo-booléenne définie (bascule)

set var = \$< entrée d'une var au clavier (Input)

unset var annulation de définition d'une variable quelconque : mot, tableau, bool.

set liste des variables locales : @ est également possible

8.1.2 Variables d'environnement tcsh

setenv<S>var<S>mot définition et affectation d'une valeur mot à une variable var dans l'environnement.

setenv var 'mot1 mot2 ...' chaîne qcq

setenv var "mot1 mot2 ..." chaîne de caractères

setenv varbool variable pseudo-booléenne définie (bascule)

unsetenv var annulation de définition

setenv liste des variables d'environnement (ou env)

8.1.3 Variables locales et d'ENV. prédéfinies dans tcsh

Certaines variables locales sont couplées aux variables d'environnement de même nom en majuscules : path, home, term, user, shell.

argv liste des mots de la ligne de commande indicé de 0 (nom de la cmd) à n (nième paramètre). Equivalent à \$0, \$1, ...

status code de retour de la dernière cmd. Equivalent à \$?
cwd "current working directory"

prompt chaîne de caractères de l'invite. Peut contenir les caractères spéciaux suivants : %/ cwd complet; %C2 seulement les 2 derniers sous-répertoires; %! numéro dans l'historique; %m nom de l'hôte; %n user name;

set prompt = "%n@%m :%C2 %! > "
dupont@cerise :C++/Tableur 24 >

history nombre de lignes de cmd à conserver dans l'historique

ignoreeof évite les exit par <Ctrl> D

filec permet l'expansion automatique ("completion") par <Tab>

tcsh version de l'interprète

8.2 Historique, alias

Capacités locales au tcsh courant.

!4 historique : rappel 4ème cmd de l'historique

!pref rappel cmd par abbréviation : préfixe de cmd

!! rappel dernière cmd

alias *nouvnom cmd* permet d'abrèger une cmd complexe

unalias *nouvnom* supprime l'alias

alias liste des alias courants

\!0 premier paramètre d'un alias (nom de l'alias)

\!* tous les paramètres de l'alias

8.3 Expressions tcsh

Attention chaque membre de l'expression doit être espacé du suivant et du précédent : @ x = 12 + (34 * 2). Le parenthésage est plus que fortement conseillé notamment si des opérateurs ayant d'autres significations sont utilisés (& | <).

= += -= *= /= %= affectations
\$#{var} taille d'une variable tableau (1 à n)
\${var} vrai (1) si var définie, faux (0) sinon
0 1 faux vrai logique
~ & | opérateurs binaires non et ou
! && || opérateurs logiques non et ou
nombre << n décalages à gauche (>> à droite)
mot =~ motif "pattern-matching" motif composé de cars, et de : * ? []
mot !~ motif pas de correspondance de motif
-e fic vrai (1) si fic existe, faux (0) sinon
-d -f -o -r -w -x -z fic répertoire?, fichier?, propriétaire?, lecture?, écriture?, exécution?, taille nulle?

8.4 Scripts de configuration

Fichiers systèmes : /etc/...

/etc/csh.cshrc script générique appelant les scripts /etc/profile.d/*.csh. Pour chaque csh ou tcsh
/etc/csh.login Uniquement pour shell de login

Fichiers Personnels : ~/...

.cshrc script lancé en tout début de session puis pour chaque nouveau processus csh
.login script lancé après .cshrc en début de session. 1 fois à la connexion.
.logout script lancé en fin de session

8.5 Structures de contrôle tcsh

if (exp) cmd \n alternative simple
if (exp1) then \n cmds1 \n else if (exp2) then \n cmds2 \n endif alternative
foreach var (mot1 mot2 ...) \n cmds \n end itérative : \$var prenant les valeurs successives de la liste; foreach...et end seuls sur leur ligne
while (exp) \n cmds \n end itérative : tant que expr ≠ 0 ...
break sortie d'une itérative
continue passage à l'itération suivante
repeat 5 cmd 5 répétitions de cmd
switch (mot) \n case mot1 : cmds1 breaksw \n case mot2 : cmds2 breaksw \n default : cmds3 breaksw \n endsw alternative multiples : case(s) et default en début de ligne : utilisation des * ? et [] dans les mots 1, 2, ...
onintr lab branchement au label "lab :" en cas d'interruption
goto lab branchement au label "lab :"

9 Bash

9.1 Variables

Toutes les variables sont d'abord locales, puis on peut les exporter dans l'environnement.

9.1.1 Variables locales bash

Attention, la syntaxe est très stricte : peu d'espaces par rapport à tcsh. Une variable non définie a comme valeur la chaîne vide.

var=motbonjour définition et affectation d'une valeur mot à une variable var
var='mot1 mot2 ...' suite de mots
var="mot1 mot2 ..." suite de mots
read var1 var2 ... entrée de var au clavier (Input). 1 mot par variable.
unset var annulation de définition d'une variable quelconque
set liste des variables locales

9.1.2 Variables d'environnement bash

export var exportation dans l'environnement de la variable var
export var=valeur exportation et affectation
export -n var var n'est plus dans l'environnement
env liste des var d'environnement (exportées)

9.1.3 Variables locales et d'ENV. prédéfinies dans bash

\$* variable non nommée par défaut affectée par set mot1 mot2 ... ou liste des arguments dans un script.
PS1 chaîne de caractères de l'invite.

9.2 Expressions bash

. fic équivaut à source fic
test exp retourne un résultat (status) nul (0) si l'expression est vraie : 0 pour vrai ; 1 pour faux! (contraire du C, tcsh)
[<S> exp <S>] même chose que test
! -a -o opérateurs logiques non et ou entre expressions
num1 -eq num2 égalité numérique (resp. -neq, -gt, -ge, -lt, -le)
-e fic vrai (0) si fic existe, faux (1) sinon
-d -f -c -b -L -r -w -x -s fic répertoire?, fichier?, périph car?, périph bloc?, lien symbolique?, lecture?, écriture?, exécution?, taille non nulle?
fic1 -nt fic2 fic1 plus récent que fic2?
chaîne teste si chaîne non vide
-z chaîne chaîne vide? (resp -n : non vide?)
chaîne1=chaîne2 égalité de chaînes; (resp !=)

9.3 Scripts de configuration bash

Fichiers systèmes : /etc/...

/etc/bashrc Pour chaque bash
/etc/profile script générique appelant les scripts /etc/profile.d/*.sh Uniquement pour shell de login.

Fichiers Personnels : ~/...

.bashrc script lancé en tout début de session puis pour chaque nouveau processus bash
.bash_profile ou .bash_login ou .profile script lancé en début d'un shell de login. 1 fois à la connexion.
.bash_logout script lancé en fin de session

9.4 Structures de contrôle bash

Dans une liste de commandes, c'est le résultat de la dernière qui compte. Les ";" séparant les commandes sont très importants.

```
if cmds ...; then cmds ...; else cmds ...; fi
    alternative simple
if [ $x ]; then echo toto ;fi exemple: si var x existe
...
if cmds ...; then cmds ...; elif cmds ...; then ...; fi
    altern. complexe
for var in mot1 mot2 ...; do cmds ...; done
    itérative : $var prenant les valeurs successives de
    la liste
for var in 1 2 3; do echo toto; done exemple: affiche
    3 fois toto
while cmds ...; do cmds ...; done itérative: tant que
until cmds ...; do cmds ...; done itérative: jusqu'à
break sortie d'une itérative
continue passage à l'itération suivante
case chaîne in motif1) cmds ...;; motif2) cmds
...;; *) ...;; esac alternative multiples; utilisation
    des *?[] dans les motifs
case $x in 1) echo un ;; [0-9]*) echo nombre ;; esac
    exemple
```

10 Informations de Session

id {-[ug]{n}} identité de l'U ou du Groupe, numéro ou Nom
logname nom d'utilisateur
hostname nom de la machine
tty nom du fichier périphérique associé au terminal (/dev/tty...)
users liste des utilisateurs loggés sur cette machine
who liste des utilisateurs, des tty, des heures de connection et des DISPLAY loggés
who am i ligne correspondant à l'utilisateur
finger liste des utilisateurs loggés

11 Réseaux et Communication

11.1 Protocoles de bas niveau

arp -a liste la table de résolution : {(nomHôte adrIP adr-sEth)...}
arp -d hostname supprime l'hôte de la table
rarp -a liste la table de résolution inverse
route liste la table de routage locale (options permettant de modifier)
netstat -rn liste la table de routage locale (ipDest, ipRouteur, ... interface)
netstat -t liste des connexions tcp
netstat -p liste des programmes associés aux sockets (root)

11.2 Interface réseau

ifconfig liste les interfaces réseaux actives et leur état
ifconfig eth0 état de l'interface eth0
ifconfig ppp0 up active l'interface ppp0 (**down** pour désactiver)
ifup ppp0 active l'interface ppp0 (ifdown désactive)

11.3 Configuration réseau

linuxconf partie réseau
netconf même chose que linuxconf (partie de)
netcfg outil graphique de configuration

11.4 Test réseau

ping 127.0.0.1 tente la communication avec soi-même (loopback)
ping localhost idem avec un serveur de nom (DNS)

11.5 Applications de communication synchrone et asynchrone

write *nomutilisateur* ouvre une session de communication on line (téléphone). Quitter par <CTRL><D>
mesg n décrocher le téléphone. **mesg y** pour redevenir accessible
talk *nomutilisateur* ouvre une session de communication interactive (on line)
talk *nomutilisateur@machine* possibilité de parler avec une autre machine
mail session mail pour lire ses messages; commandes: h entêtes, 3 3ième msg, m dupont mail, r reply, p print, + ou n next, - précédent, s fic save, d détruit, q quitte
mail *nomutilisateur* pour envoyer un message (off line). Quitter par <CTRL><D>
mail *nomutilisateur* < fic pour envoyer un fichier
mh système de gestion du mél (mail): ensembles de commandes (inc,next,show,...) utilisées par **xmh** et **exmh**
ftp *numero-ip* session de transfert de fichier (prompt pour éviter l'interactif en cas de transfert multiple mput mget) ...

11.6 Réseaux

NIS (yellow pages) pages jaunes permettant à un groupe de machines de partager des fichiers de configuration (passwd, group, hosts,...)
nisdomainname donne le nom du domaine NIS ou affecte
dnsdomainname donne le nom du domaine DNS ou affecte
hostname donne le nom de la machine (DNS) ou affecte
yppasswd change le mot de passe NIS
ypchsh pour changer son shell
ypchfn pour changer ses info
ypcat -x liste des fichiers accessibles par ypcat
ypcat -k *passwd, hosts, aliases, group, networks, services* affiche le contenu d'un fichier des pages jaunes (key, value)
ypmatch *machine hosts* affiche la ligne du fichier des pages jaunes "hosts" correspondant à machine.
ypmatch meynard passwd informations de passwd sur meynard
ypmatch lic aliases nom des étudiants de licence
ypwhich donne le nom du serveur NIS
nslookup *machine* donne le numéro ip de la machine (DNS)

12 Divers

pwd affiche le wd
echo *liste de mots* echo sur sortie standard

date date et heure
datef date et heure en français

12.1 L'éditeur vi

vi *fic* appelle l'éditeur vi en mode commande : 2 modes à bascule commande/frappe. Mode frappe : texte et <backspace>;
i passe en mode frappe : "insertion"
a passe en mode frappe : "append"
<Esc> passe en mode commande
:w *fic* sauve le texte courant sous le nom *fic* : "write"
:e *fic* édite un nouveau fichier
:q quitte vi
:wq pour sauver et quitter
:set all pour voir toutes les options
:set number pour numéroter les lignes
6x supprime 6 cars courants
2dd supprime 2 lignes courantes
D supprime jusqu'à la fin de la ligne
Y,p Copier / Coller 1 ligne : n lignes : nY, np
u undo
flèches déplacements ou j,k,l,m
Ctrl-B Ctrl-F déplacements 1 page : backward, forward
.exrc fichier de configuration de vi (set ...)

13 Gestion des terminaux

<Ctrl> <Alt> <F1> ... <F6> 6 consoles textes (Linux),
<Ctrl> <Alt> <F7> la console graphique.
<Ctrl> <Alt> <Backspace> termine la session X (Linux)
<Ctrl> <Alt> <Suppr> en console txt, reboot.
<Alt> <Setup> config. terminal X pour la connexion
bt *type ip_Tal ip_serveurGal ip_passerelle masque* boots-trap du Tal X ou plus simplement **bt**
stty -a liste la configuration notamment : erase, intr, eof, ...
stty erase ^? configure la touche backspace pour la fonction d'effacement "erase"
stty intr ^C susp ^Z <Ctrl> c (resp <Ctrl> z) pour interrompre (resp suspendre) un pus
xmodmap -e "keycode 102=Delete" configure la table de conversion touche physique, touche logique du **terminal X**
xmodmap -pk liste de la table de conversion
xev & visualisation des évènements X (codes) du clavier et de la souris
ypcat hosts pour connaître les numéros d'ip des machines et terminaux
setenv DISPLAY bidon :0 permet de travailler sur l'écran d'un PC distant dont le nom est connu par le serveur de nom du domaine de l'hôte
setenv DISPLAY 193.49.106.146 :0.0 permet de travailler sur un terminal X distant (numéro ip)

14 Applications X

Lancer ces applications de façon détachée (&). Souvent <Ctrl> <W> permet de supprimer la sélection.
<Ctrl> <Alt> <Backspace> termine la session X (Linux)
xfontsel visualiser les polices, puis en sélectionner une, puis coller dans une application X : à droite de *appli*font* : dans .Xdefault

xrdb .Xdefault relancer .Xdefault
xmh gestionnaire de mail
xfig création de figures
xwd -frame -out fic.xwd sauve la fenêtre cliquée dans fic.xwd
xwud -in fic.xwd visualise la fenêtre sauvee
xtwd fic.xwd crée un fichier pour tex (ne fonctionne pas)
xpr -device ps -portrait -output fic.ps < fic.xwd créeun fichier postscript de la fenêtre
xwininfo informations sur une fenêtre
ghostview visualisateur postscript
editres arbre des widgets d'une appli. X
/usr/lib/X11/app-defaults/ appli ressources par défaut d'une appli. X

15 KDE

kfm, kedit, kmail, konsole, kvt, knotes, ...

16 Administration Linux

16.1 Démarrage/Arrêt/Messages

shutdown +10 redémarrage du système dans 10 minutes après avoir envoyé un msg aux Util. **shutdown now** arrête de suite.
/sbin/reboot ou **/sbin/halt** redémarrage du système sans avertissement.
dmesg | tail messages du noyau (démarrage)
tail /var/log/messages affiche les 10 dernières entrées du journal

16.2 Services (démons)

Les démons sont lancés au démarrage du système par des liens situés dans un répertoire dépendant du niveau de démarrage, souvent : **/etc/rc5.d**. Ces liens référencent des scripts dans **/etc/init.d**, par exemple : **network, sshd, ypbind, xinetd**. **xinetd** est un démon particulier puisqu'il permet de lancer d'autres démons (telnet, ftp, cvs,...) à la volée lorsqu'un client envoie une requête. Enfin, les répertoires **/etc/xinetd/**, **/etc/httpd/** contiennent les fichiers de configuration de ces démons.

Commandes utiles :

ps aux | grep sshd pour voir si le démon tourne
kill -9 6869 tue un démon
/etc/init.d/sshd lance le démon sshd
/etc/init.d/xinetd restart relance les services de xinetd

16.3 Internet

Fichiers :

/etc/hosts adrsIp nom1 nom2 ... associations locales entre ip et noms symboliques
host.conf ordre dans lequel on cherche la conversion des noms symboliques en adrs ip (order hosts,bind,dns)
hosts.allow service : machine ... indique les machines autorisées à se connecter aux services xinet. Par exemple : **cvs : ALL**

16.4 Disques et fichiers

mount/umount montage/démontage des SF
df disk free : pourcentages libres des SF montés
du -sk rep1 (disk usage) taille totale utilisée par le répertoire rep1 et sa descendance
dd if=fic1 of=fic2 bs=512 count=1 (disk dump) copie le premier bloc de 512 octets du fichier fic1, dans le fichier fic2

16.5 Utilisateurs, groupes

Machine isolée : fichiers de config, réseau : NIS (commandes yp...)

/etc/passwd fichiers des Users :
meynard :tv34ze :500 :500 :michel
Meynard,,,: /home/meynard :/bin/tcsh (login, passwd crypté, UID, GID par défaut, infos, HOME dir, shell)
/etc/group fichier des Groupes :
users : :100 :meynard, dupont (Group name, passwd, GID, membres)
usercfg permet de configurer graphiquement les U et les groupes (ajout, maj, sup)

16.6 Paquetages RPM

Archives compressées gérant les dépendances et organisées en hiérarchie.

gnorpm & gestion des paquetages rpm

rpm -ivh <fic.rpm> installe le package, **-Uvh <package>** upgrade, **-qa** query all (liste des packages), **-qi auctex** infos sur auctex installé, **-qip /mnt/cdrom/RedHat/RPMS/toto.rpm** info sur le paquetage du cdrom, **-qf /bin/ls** : indique le paquetage auquel appartient le fichier ls, **-qg Applications/Editors** liste des paquetages de ce groupe, **-qR kdbase** liste des paquetages Requis, **-ql kdbase** liste des fichiers, **-qp ftp://ftp.lip6.fr/pub/linux/distributions/redhat/redhat-6.2/i386/RedHat/RPMS/bison-1.28-2.i386.rpm** nom du paquetage distant (ftp ou http), **-v auctex** vérifie la présence des fichiers, **-e auctex** désinstalle.

Pb rpm ivvh instal. très Verbeux, **-rebuilddb** reconstruit la Bd des paquetages.

rpm2html crée une BD HTML à partir de paquetages rpm, y compris les dépendances

rpmfind client pour rpm2html : cherche des paquetages sur un site web.