EM8BTEM - INTRODUCTION A LA MODELISATION MOLECULAIRE

PRESENTATION DU SYSTEME UNIX/LINUX

Georges Czaplicki
IPBS-CNRS, 205, route de Narbonne, 31077 Toulouse, France
Tél. : 05 61 17 54 04, email : cgeorge@ipbs.fr

UNIX (Linux)

Pour :

I. Système multitâches

II. Système multiutilisateurs

III. Système fiable

IV. Système sécurisé

V. Système optimisé

Contre :

I. Système complexe (?)

…

A QUOI ÇA SERT ?

- **Système flexible**
  - travail multitâche et multiutilisateur
- **Développement de logiciels**
  - écriture de programmes (traitement de texte)
  - compilation (C, Fortran, …)
  - exécution (nouvelles commandes)
  - *debugging*
- **Utilisation du réseau**
  - travail à distance (ssh, rlogin)
  - récupération de fichiers (sftp)
  - consultation du courrier électronique (elm)
  - recherche de données sur Web (Firefox)
- **Autres applications…**

HISTOIRE

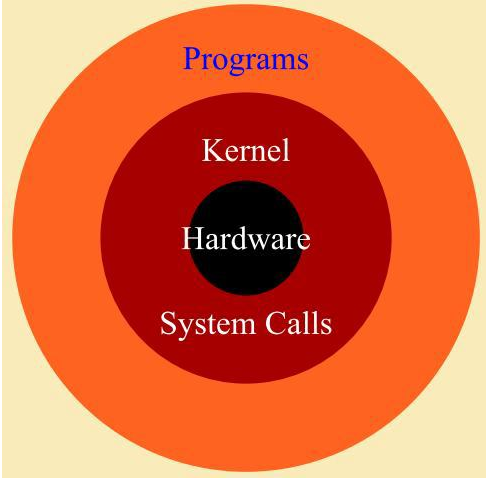
1969	Naissance du système **Unix** (***Uniplexed Information and Computer Service***) dans les laboratoires de Bell Labs, filiale d'AT&T. Les auteurs, **Kenneth Thompson** et **Dennis Ritchie**, l'ont écrit en assembleur.
1971	La date officielle de la première version du système Unix, dont nom est dérivé de **Multics** (***Multiplexed Information and Computer Service***), un système interactif.
1973	La troisième version du système Unix, complètement réécrit en langage C, apparaît sur le marché. Unix devient portable.
1974	L'université de Berkeley en Californie devient un centre du développement Unix (éditeur vi, la gestion de la mémoire virtuelle, etc.).
1977	Le début de la version Berkeley d'Unix : **BSD** (***Berkeley Software Distribution***).
1981	Le début de l'ère commerciale d'Unix : **Système III**, version **4.1 de BSD**, **Xenix** de Microsoft et SCO (***Santa Cruz Operation***).
1983	**Système V** : version intégrale et finale.
1990-	Création du système **Linux** par Linus Torvalds

LES SYSTEMES UNIX

Provenance	Version
Sun Microsystems	SunOS / Solaris
Digital / Compaq	Digital Unix (Tru64)
Hewlett Packard	HP-UX
Silicon Graphics, Inc.	Irix
Cray	UNICOS
UC Berkeley	BSD
Linus Torvalds	Linux

LES CARACTERISTIQUES GENERALES

Le **noyau** (***kernel***) :	contient toutes les informations pour le fonctionnement de base d'Unix.
Un **coquillage** (***shell***) :	se charge des relations entre le noyau et l'utilisateur.
Les **commandes** :	peuvent correspondre au langage propre d'un ***shell***, ou représenter les programmes externes, fournis par l'utilisateur.



PRESENTATION

ENTREE ET SORTIE DU SYSTEME

L'ORGANISATION DES FICHIERS ET DES REPERTOIRES

LES COMMANDES UTILES

GESTION DE REPERTOIRES ET DES FICHIERS

LES ENTREES ET SORTIES DE DONNEES

TRAVAIL EN MODE MULTITACHES

INTERACTION AVEC LES AUTRES ORDINATEURS

ENCHAINEMENTS, REDIRECTIONS ET CONTROLE DE TACHES

ECRITURE DE SCRIPTS SHELL

ENTRER ET SORTIR DU SYSTEME

Unix est un système complexe, susceptible d'être utilisé par un grand nombre d'utilisateurs, sur divers terminaux, parfois très éloignes les uns des autres. Chacun souhaite protéger son travail pour en assurer la confidentialité ou simplement pour être sûr de retrouver le lendemain le travail de la journée. Dans ce contexte, la **sécurité** est donc un aspect important.

Pour utiliser le système Unix, un individu doit d'abord obtenir un **compte** (***account***) auprès de **l'administrateur du système** (***super-user ; root***) qui lui accorde un **nom d'utilisateur** (***login*****)** et un **mot de passe** **(*password*)**. Habituellement un utilisateur fait une partie d'un **groupe** (***group***) d'utilisateurs.

Avec ces renseignements un utilisateur peut approcher un **terminal**, choisir un **serveur**, et taper sur le clavier les mots d'identification :

```
login : login_name <ENTREE>
password : mon_mot_de_passe <ENTREE>
```

qui lui donnent l'accès aux ressources du serveur choisi.

ENTRER ET SORTIR DU SYSTEME (2)

Le shell envoie un message, indiquant que le système est prêt à recevoir les commandes. Ce message symbolique s'appelle **l'indicatif** du shell (***shell prompt***), et il peut prendre de nombreuses formes, en fonction du système et du shell lui-même. Par exemple :

Shell	Indicatif
Bourne shell	$
C shell	%

Pour se **déconnecter** (***logout***) on peut choisir une de commandes suivantes :

```
% exit  <ENTREE>
% logout <ENTREE>
% <Ctrl-D>
```

LA SYNTAXE GENERALE DES COMMANDES

nom	options	arguments
ls	-lR	/tmp

NOM un mot ou une abréviation, en minuscules (par exemple, ***ls = list***, affiche le contenu d'espace de travail).

OPTIONS précision de l'action de la commande, en minuscules ou en majuscules ; l'ensemble des options est précédé d'un tiret afin de ne pas confondre les arguments et les options.

ARGUMENTS spécification des objets sur lesquels la commande doit agir.

Chaque élément (le nom, les arguments, le groupe des options) est séparé des autres par un **espace**.

LA COMMANDE LA PLUS UTILE

Pour consulter le manuel (***online manual***) sur une commande ou un mot-clé (***keyword***):

`man commande` ou **`man –k mot-clé`**

```
% man man

MAN(1)

NAME
     man - print entries from the on-line reference manuals; find manual
entries by keyword

SYNOPSIS
     man [-cdwWtpr] [-M path] [-T macropackage] [section] title ...
     man [-M path] -k keyword ...
     man [-M path] -f filename ...

DESCRIPTION
     man locates and prints the titled entries from the on-line reference
manuals
--More–
```

LES COMMANDES DE BASE

`passwd` (***password***) : changement de mot de passe

Exemples :

```
% passwd
Changing password for dbio1
Old password:
New password:
Re-enter new password:

% passwd
Changing password for dbio1
Old password:
New password:
Password must contain at least two alphabetic characters and at least one
numeric or special character.
New password:
Passwords must differ by at least 3 positions
Too many failures
Try again later.
```

LES COMMANDES DE BASE (2)

who : la liste des utilisateurs connectés au système

Exemples :

```
% who
dbio1      ttyq1        Dec 29 11:12
dbio2      ttyq0        Dec 29 10:06
dbio2      ttyq2        Dec 18 23:34
% who –q (quick)
dbio1 dbio2 dbio2
# users=3
% who am i
dbio1      ttyq1        Dec 29 11:12
% whoami
dbio1
```

LES COMMANDES DE BASE (3)

hostname : affichage du nom de la machine

uname : affichage de l'information sur le système utilisé

date : affichage de la date et l'heure du système

```
% date
Tue Dec 29 11:51:31 MET 1998

% date '+DATE: %m/%d/%y%nTIME: %H:%M:%S'
DATE: 12/29/98
TIME: 12:02:54
```

LES COMMANDES DE BASE (4)

`cal` (*calendar*) : le calendrier pour le mois courant

```
% cal
   January 1999
 S  M Tu  W Th  F  S
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31


% cal 1 2001
   January 2001
 S  M Tu  W Th  F  S
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

TYPES ET NOMS DE FICHIERS

Types : (i) normal, (ii) répertoire ou (iii) spécial

Noms : tous caractères sauf "/" ; éviter les métacaractères :

? @ # $ ^ & * ( ) [ ] \ | ; ' " < > { }

Les "+", "-" et ".." ne devraient pas commencer un nom !

NOTES : - les majuscules et les minuscules sont différentes !
- il n'y a pas de types de fichier dédiés (sauf applications)

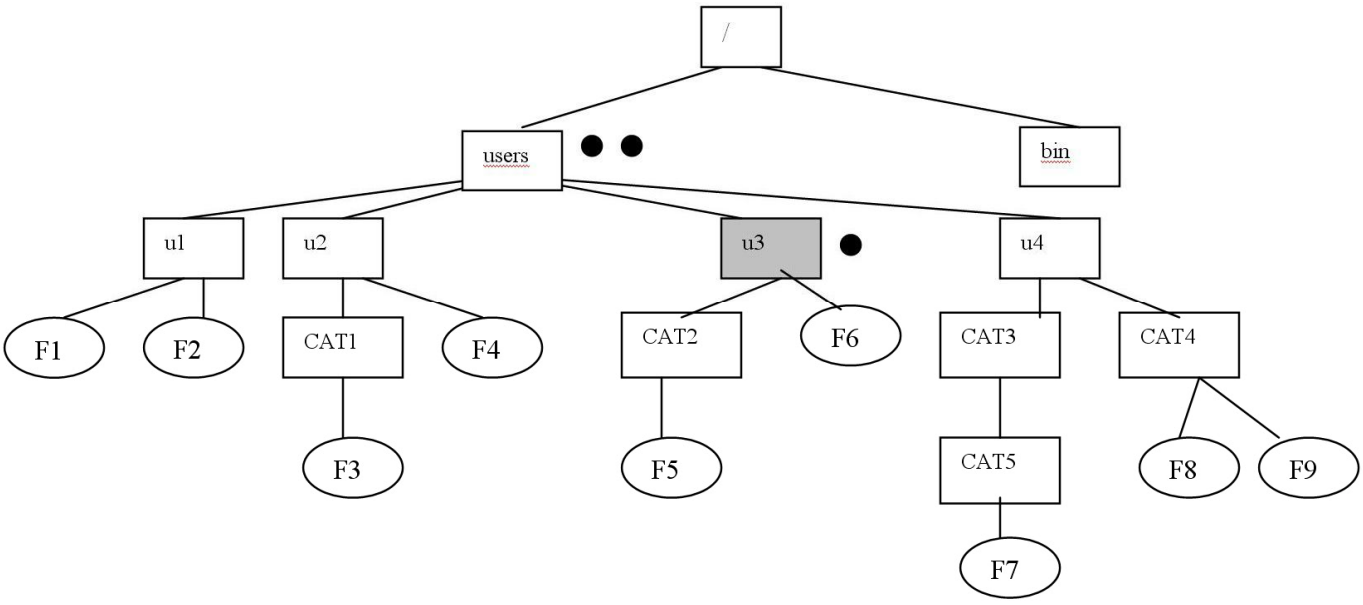
Jokers (*wildcards*) :

`*`	→	chaîne quelconque de caractères (sauf fichiers cachés)
`?`	→	caractère quelconque
`[ccc]`	→	caractère quelconque dans la liste spécifiée
`[l-u]`	→	caractère quelconque dans la plage spécifiée
`~`	→	abréviation pour le répertoire de login

Exemples :

```
ls test*   ls test?.dat   test[0-9].dat     ls ~bionf1 ls
```

L'ORGANISATION DES FICHIERS ET DES REPERTOIRES SUR LE DISQUE



Pour trouver un fichier ou un répertoire, Unix doit connaître son **chemin d'accès** (***path name***).

Exemple : chemin absolu et relatif du fichier **F5 :**

```
/users/u3/CAT2/F5                    ./CAT2/F5
```

LA LOCALISATION ET LES DEPLACEMENTS DANS L'ARBORESCENCE DES FICHIERS ET DES REPERTOIRES

pwd (***print working directory***) : affichage du chemin du répertoire courant

Exemple :

```
% pwd
/users/u3
```

cd nom (***change directory***) : se déplacer dans l'arborescence

Exemples :

```
% cd CAT2
% cd ..
% cd bin
bin - No such file or directory
% cd /bin
/bin – Permission denied
```

AFFICHAGE DU CONTENU DES REPERTOIRES

`ls` (***list***): la liste du contenu du répertoire courant

```
% ls
CAT2  F6   p2g

% ls -l
drwxr-xr-x  2 u3  dea  512 Dec 29 17:18 CAT2
-rw-r--r--  1 u3  dea   26 Dec 29 17:18 F6
-rwxr-xr-x  1 u3  dea  999 Dec 29 17:18 p2g

% ls –lR
total 2
drwxr-xr-x  2 u3  dea  512 Dec 29 17:18 CAT2
-rw-r--r--  1 u3  dea   26 Dec 29 17:18 F6
-rwxr-xr-x  1 u3  dea  999 Dec 29 17:18 p2g

./CAT2:
total 1
-rw-r--r--  1 u3  dea    1 Dec 29 17:18 F5

% ls *2*
CAT2  p2g
```

GESTION DE REPERTOIRES

`mkdir nom` (***make directory***) : créer le répertoire **nom**

Exemple :

```
% mkdir sortie
% ls -F
CAT2/  F6   p2g*  sortie/
% mkdir –r sortie/1/toto
%
```

`rmdir nom` (***remove directory***) : enlever le répertoire **nom**

Exemple :

```
% rmdir sortie/1/toto
%
```

PERMISSIONS D'ACCES

```
% ls -l
drwxr-xr-x  2 u3  dea  512 Dec 29 17:18 CAT2
-rw-r--r--  1 u3  dea   26 Dec 29 17:18 F6
```

type	u (*user*)	g (*group*)	o (*other*)	signification
d ou -	**r w x**	**r w x**	**r w x**	r (***read***) = lecture w (***write***) = écriture x (***execute***) = exécution

`chmod zzz nom` (***change mode***) : changer les bits des permissions du fichier ou répertoire **nom**

Permissions : '+' = ajout ; '-' = suppression ; '=' = attribution ; 'a' (***all***) = tout le monde

Exemples :

```
% chmod g=rwx CAT2        (résultat : drwxrwxr-x)
% chmod g+w CAT2          (résultat : drwxrwxr-x)
% chmod a=r F6            (résultat : -r--r--r--)
% chmod u=rx,o=rw F6      (résultat : -r-xr--rw-)
```

PERMISSIONS D'ACCES (2)

Représentations numériques de permissions :

r = 4	w = 2	x = 1

Total : entre 0 et 7

Exemples :

```
chmod 777 fichier
      ugo
```

```
chmod 750 fichier
u = rwx
g = r-x
o = ---
```

AFFICHAGE DE FICHIERS

Commande	Résultat
echo	envoi d'une chaîne de caractères
cat	concaténation
head	affichage de 10 premières lignes d'un fichier
tail	affichage de 10 dernières lignes d'un fichier
more	affichage du contenu page par page
less	affichage du contenu page par page

(voir exemples ci-dessous)

GESTION DES FICHIERS

`cp nom1 nom2` (***copy***) : copier le fichier **nom1** vers **nom2**

```
% cp  -i  F6   f6
% cp  CAT2/F5   ./F55
% mkdir CAT22 ; cp –r CAT2 CAT22
% ls CAT22
CAT2
% ls CAT22/CAT2
F5
```

`rm nom` (***remove***) : supprimer le fichier **nom1**

```
% rm f6
% rm –r CAT22
% rm –i F*
```

GESTION DES FICHIERS (2)

mv nom1 nom2 (***move***) : déplacer et renommer les fichiers

```
% mv  -i  F6  f6
% mv  CAT2/F5  .
% mv  f6 /tmp
```

cat nom (***catenate***) : voir le contenu du fichier **nom**

```
% cat f6
Ceci est le contenu du fichier f6.
% cat f6 CAT2/F5
Ceci est le contenu du fichier f6.
Ceci est le contenu du fichier F5.
```

GESTION DES FICHIERS (3)

chown prp nom (***change owner***) : changer le propriétaire du fichier **nom** à **prp**

```
% ls –l f6
-rw-r--r--  1 u3  dea   26 Dec 29 17:18 f6
% chown u2 f6
% ls –l f6
-rw-r--r--  1 u2  dea   26 Dec 29 17:18 f6
```

chgrp grp nom (***change group***) : changer le groupe du fichier **nom** à **grp**

```
% ls -l f6
-rw-r--r--  1 u2   dea  26 Dec 29 17:18 f6
% chgrp biol f6
-rw-r--r--  1 u2  biol  26 Dec 29 17:18 f6
```

GESTION DES FICHIERS (4)

more –opt nom : afficher le contenu du fichier **nom**

```
% more f6
Ceci est la première ligne d'écran
…
Ceci est la dernière ligne d'écran
--More–(x %)              (q, f, /xxxx, =, ?)
% more +57 f6
% more +/'Christine' f6
```

head nom (***header***) : afficher le début du fichier **nom**
tail nom (***tail***) : afficher la fin du fichier **nom**

```
% head f6                 (par défaut : 10 lignes)
% head –20 f6
% tail –40 f6
```

GESTION DES FICHIERS (5)

grep exp nom : extraire du fichier **nom** les lignes répondant à des critères de recherche

```
% cat telephones
Dupuis Nicole       01 45 25 32 22
Herman Pierre       03 58 69 33 24
Poitevin Michelle   05 54 88 71 01
Boutier Martine     05 61 22 58 81
Herman Pierre       03 58 69 33 24
% grep 'Dupuis' telephones
Dupuis Nicole       01 45 25 32 22
% grep '01' telephones
Dupuis Nicole       01 45 25 32 22
Poitevin Michelle   05 54 88 71 01
% grep '^H' telephones
Herman Pierre       03 58 69 33 24
Herman Pierre       03 58 69 33 24
```

GESTION DES FICHIERS (6)

`sort nom` : trier le fichier **nom**

```
% sort telephones
Boutier Martine          05 61 22 58 81
Dupuis Nicole            01 45 25 32 22
Herman Pierre            03 58 69 33 24
Herman Pierre            03 58 69 33 24
Poitevin Michelle        05 54 88 71 01
```

`uniq nom` (***unique***) : extraction des lignes non répétées

```
% uniq telephones
Boutier Martine          05 61 22 58 81
Dupuis Nicole            01 45 25 32 22
Herman Pierre            03 58 69 33 24
Poitevin Michelle        05 54 88 71 01
%
```

GESTION DES FICHIERS (7)

`find chemin –options` : rechercher des fichiers

```
% find / –name toto -print
% find . –name ‘F*’ –print
% find .. –name ‘*.bak’ –exec rm {} \;

% find / -type d –print
% find / -user u3 –print
% find / -user u3 –exec ls –l {} \;
% find / -group dea –print

% find . –mtime  7 –print
% find . –mtime –7 –print

% find / -type f \( -name fid -o -name ser \)  -exec ls -o {} \;
```

GESTION DES FICHIERS (8)

ln –options cible source (link) : créer un lien

```
% ll
drwxr-xr-x   2 cgeorge gmilon  4096 2010-11-30 16:40 bin
-rw-r--r--   1 cgeorge gmilon  1132 2010-11-26 10:57 machid.out

% ln -s machid.out toto

% ll
drwxr-xr-x   2 cgeorge gmilon  4096 2010-11-30 16:40 bin
-rw-r--r--   1 cgeorge gmilon  1132 2010-11-26 10:57 machid.out
lrwxrwxrwx   1 cgeorge gmilon    10 2010-12-20 23:17 toto -> machid.out
```

LES ENTREES ET SORTIES DE DONNEES

entrée → commande → sortie

< : redirection des entrées de données

```
% cat < telephones
```

> : redirection des sorties de données (création)

```
% cat > backup
…                                   (terminer par Ctrl-D)
% date > /users/dbio1/reference
```

>> : redirection des sorties de données (ajout)

```
% cat < telephones >> backup
% ls –l >> /users/dbio1/reference
```

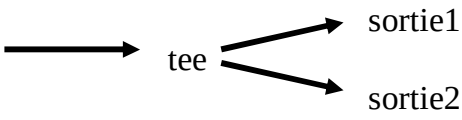
LES ENTREES ET SORTIES DE DONNEES (2)

| (***pipe***): redirection du sortie d'une commande vers l'entrée d'une autre commande

```
% ls -l > temp
% more temp
```

```
% ls -l | more
```

tee s1 | s2(*T*) : dédoubler un tube vers deux sorties




```
% ls –l | tee temp | more
```

RESSOURCES DU SYSTEME

df (***disk free***) : vérification de l'espace libre sur un disque

```
% df -k
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdb2              39G   31G  5.8G  85% /
/dev/sdb1             124M   58M   60M  50% /boot
/dev/sdb5             401G  167G  215G  44% /home
/dev/sdc1             466G  209G  258G  45% /work
```

du (***disk usage***) : vérification de l'espace occupé sur un disque

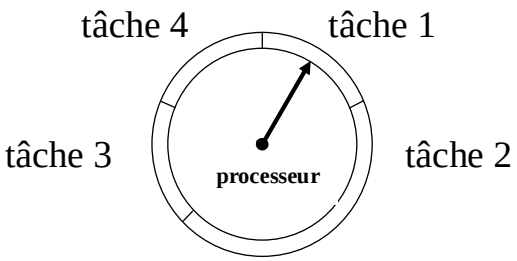
```
% du -H
40K     ./src/src/toolbox/matrix
6.9M    ./src/src/toolbox
7.2M    ./src/src
171M    ./src
4.9G    .
```

TRAVAIL EN MODE MULTI-TACHE

& (***ampersand***) : exécuter une tâche en arrière-plan

```
% find / -name '*.tmp' –print > out &
```

Priorité



nice com (***nice and friendly***): exécuter une tâche en arrière-plan

```
% nice –15 find / -name '*.tmp' –print > out &
```

TRAVAIL EN MODE MULTI-TACHE (2)

ps –opt (***process status***) : affichage des processus en cours d'exécution

```
% ps
   PID TTY      TIME COMD
 29825 ttyq0    0:00 ps
 29613 ttyq0    0:01 csh
% ps –f
   UID   PID  PPID  C    STIME TTY      TIME COMD
 dbio1 29884 29613  2 18:03:38 ttyq0    0:00 ps -f
 dbio1 29613 29612  0 14:47:44 ttyq0    0:01 –csh
% ps -cf
   UID   PID  PPID  CLS PRI    STIME TTY      TIME COMD
 dbio1 29613 29612   TS  39 14:47:44 ttyq0    0:02 -csh
 dbio1 29992 29613   TS  61 18:13:10 ttyq0    0:00 ps -cf
```

TRAVAIL EN MODE MULTI-TACHES (3)

`kill PID` : terminer un processus

```
% kill –9 29884
% killall nom_du_programme
```

`nohup com` (***no hangups***) : permettre à un processus en arrière-plan de se terminer après déconnexion (***logout***), avec la sortie redirigée vers non-tty

```
% nohup find / -name '*.tmp' –print > out &
```

AUTRES COMMANDES UTILES

`tar -options archive fichiers` : archivage de fichiers

Exemple :

```
% tar czvf arch.tar.gz *.log
```

Options:

- **c → créer un archive**
- **z → comprimer les données d'après l'algorithme gzip**
- **v → verbose (info sur le progrès affiché sur l'écran)**
- **f → rediriger la sortie vers un fichier (nom requis)**

INTERACTION AVEC LES AUTRES ORDINATEURS

`ssh -X user@machine` (***secure shell***) : travailler à distance

```
% ssh -X ondine.cict.fr
…
login:
Password:
```

`sftp` (***secure file transfer protocol***) : transférer les fichiers

```
% sftp m2bio1@ondine.cict.fr
Connecting to ondine.cict.fr...
m2bio1@ondine.cict.fr's password:
...
sftp> dir
...                          (commandes possibles : get, put, cd, lcd, …)
sftp> bye
```

INTERACTION AVEC LES AUTRES UTILISATEURS

`write nom` : écrire à qqn

```
% write bbm30
…
Message from bbm11 (tty??)  [ date ]...
…
user is logged on more than one place.
     You are connected to "terminal".
     Other locations are:
     terminal

UX:write:ERROR:User is not logged on.
(if the person you are trying to write to is not logged on)

UX:write:ERROR:Permission denied.
(if the person you are trying to write to denies that permission (with mesg))

UX:write:Warning: cannot respond, set mesg -y.
(if your terminal is set to mesg n and the recipient cannot respond to you)

UX:write:ERROR:Can no longer write to user.
(if the recipient has denied permission (mesg n) after you had started
writing)
```

INTERACTION AVEC LES AUTRES UTILISATEURS

`talk nom` : bavarder avec qqn

```
% talk bbm30
...
Message from TalkDaemon@ondine...
    talk: connection requested by bbm11@ondine.
    talk: respond with: talk bbm11@ondine
% talk bbm11
…
```

Hello
How are you ?

CREATION DE NOUVELLES COMMANDES SHELL

On peut se servir de variables prédéfinies :

$USER	→	**nom d'utilisateur actuel**
$HOME	→	**nom du répertoire de login**
$PATH	→	**chemin de recherche des commandes**
$PWD	→	**chemin du répertoire courant**

...

Variables positionnelles et spéciales :

$0	→	**nom du script**
$1 $2 ...	→	**paramètres positionnels 1,2,...**
$#	→	**nombre de paramètres positionnels**
$*	→	**ensemble de paramètres positionnels**
$$	→	**PID du shell courant**
$?	→	**code retour de la dernière commande**

Scripts SHELL

```
% ls
anne denise jean robert

% ls anne
fichier1 fichier2

% ls denise
fichier3 fichier4 fichier5
...
```

SCRIPT "contenu" :

```
#!/bin/sh
ls |
while read name; do
   echo $name ":";
   ls $name;
done
```

Exécution du script :

```
% contenu
anne :
fichier1 fichier2
denise :
fichier3 fichier4 fichier5
...
```