

Cours de Systèmes d'Exploitation (Unix)

H.Bourzoufi

Caractéristiques d'UNIX

- ❑ **Portabilité** (écrit en C)

- ❑ **Multi-utilisateurs** Plusieurs utilisateurs peuvent se connecter et travailler en même temps.
Chaque utilisateur a son environnement personnel.

- ❑ **Multi-tâches** Un même utilisateur peut lancer plusieurs travaux simultanément.

- ❑ **Interactif** Il est possible de dialoguer avec l'ordinateur.
Possibilité aussi de lancer des processus (tâches) en arrière plan et en mode différé (batch).

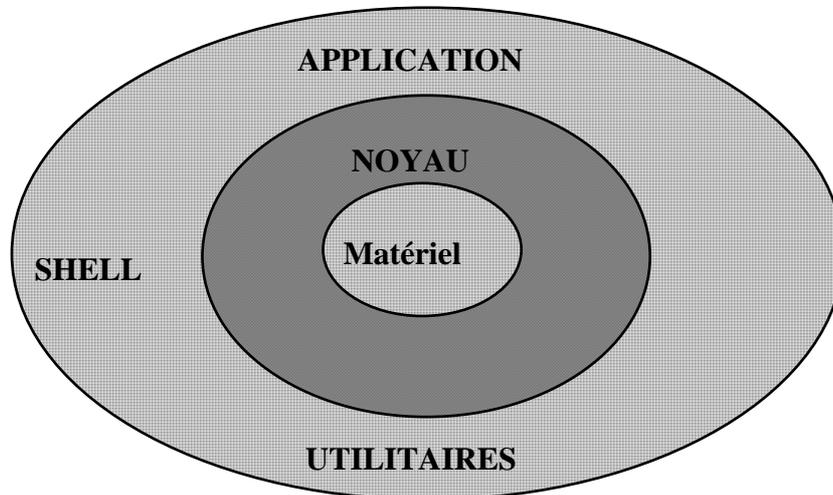
- ❑ **Un système de fichiers hiérarchisé**
(Organisation arborescente)

- ❑ **Un mécanisme de protection**
 - identification des utilisateurs par mot de passe
 - protection des fichiers
 - Un super-utilisateur

- ❑ **Une vision simplifiée par l'utilisateur des entrées-sorties** Les périphériques sont représentés par des noms de fichier, et peuvent être utilisés comme des fichiers ordinaires)

- ❑ **Le choix d'un langage de commandes :**
Les shells.

Structure du système UNIX



- Le noyau gère les tâches de base du système :
 - L'initialisation du système
 - La gestion des ressources
 - La gestion des processus
 - La gestion des fichiers
 - La gestion des Entrées/Sorties

- L'utilisateur communique avec le noyau par l'intermédiaire d'un SHELL. Les shells sont aussi des langages de commandes et de programmation.

- Les shells les plus connus sont :
 - BOURNE SHELL KORN-SHELL
 - C-SHELL TC-SHELL

- Les utilitaires sont des outils d'interfaçage avec le système, de programmation et de communication.

UNE SESSION UNIX

❑ Etablir une connexion

Login : dupont

password : mot de passe sans écho

Exécution du fichier d'initialisation ".login"

/users/licinfo/linf01 % Le prompt

(La connexion est effectuée,
le système attend les commandes)

❑ Quelques commandes simples :

date ; cal Date; calendrier

who Qui est connecté?

logname; pwd Qui suis je?; Où suis je ?

tty nom du terminal

passwd Changer le mot de passe

❑ Dialoguer avec un autre utilisateur

- Partage des fichiers
- messagerie : *write, mail*

write nom_de_utilisateur
ceci est un essai

.....

Ctrl D (termine le message)

mail nom_de_utilisateur

subject :

Bonjour,

Ceci est un essai,

. (termine le message)

Pour consulter sa boîte aux lettres : mail

LES FICHIERS UNIX

❑ Un fichier dans UNIX est une séquence d'octets, le noyau n'impose aucune structure spécifique aux fichiers.

❑ Le rôle d'un fichier est de conserver, de traiter et transmettre de l'information.

❑ Différents types de fichiers

Ordinaires (ascii ou binaires)

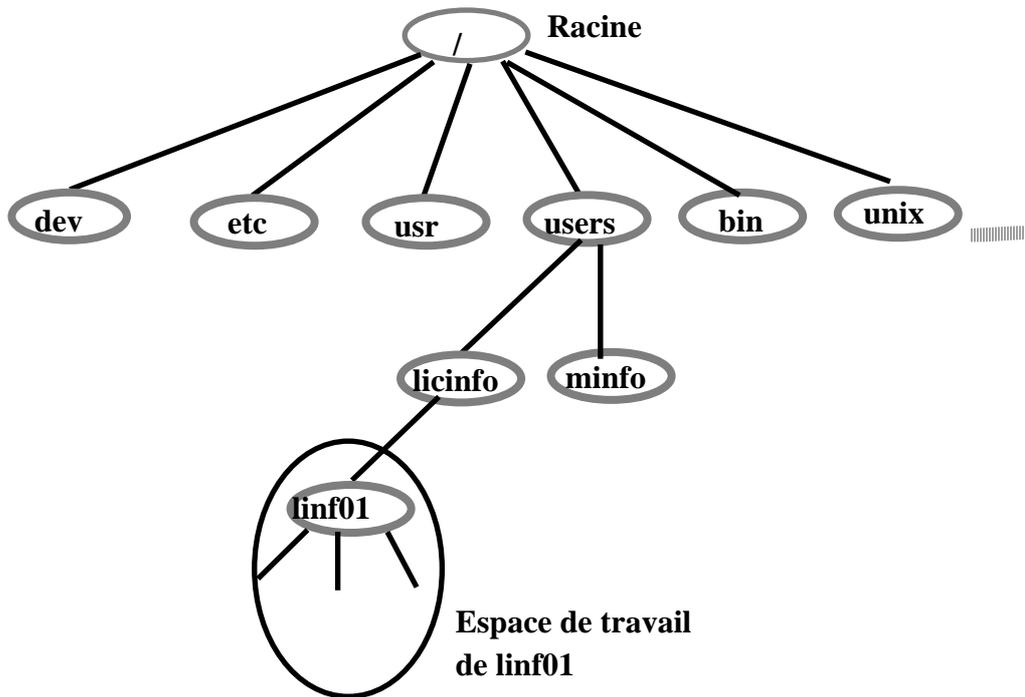
répertoires (contiennent des noms de fichiers
et/ou de catalogues)

Spéciaux Ils sont associés aux périphériques

Ex : disques, imprimantes, terminaux

Les filtres : permettent la communications entre processus

Organisation des fichiers



/dev contient les fichiers associés aux périphériques :

Ex : **lp** : imprimante

tty n terminal n° **n**

hkn disque dur n° **n**

mt n bande magnétiques n° **n**

/etc contient les fichiers d'administration

/bin contient les commandes Unix

/usr commandes et bibliothèques supplémentaires

/users répertoire des utilisateurs

/unix fichiers systèmes

Remarque : Unix possède un seul système de fichiers

DEPLACEMENT DANS LE

SYSTEME DE FICHIERS

- ❑ A la connexion, l'utilisateur est automatiquement positionné dans son répertoire de connexion .

Ex : /users/deustiosi/iosi01

- ❑ **Pour se déplacer dans la hiérarchie :**

cd <nom_répertoire>

- nom_répertoire peut être absolu ou relatif :
 - absolu** : préciser tout le chemin
 - relatif** : par rapport au répertoire courant

Ex : cd /users/deustiosi (chemin absolu)
cd iosi01 (chemin relatif)

Remarque: un nom absolu commence toujours par "/" .

- ❑ **Quelques commandes simples**

pwd affiche le répertoire courant
cd retour au répertoire de connexion
cd .. monter d'un niveau dans la hiérarchie
cd ../.. monter de 2 niveaux dans la hiérarchie

la caractère "." désigne le répertoire courant.

Principales commandes sur les fichiers

❑ Opérations de base sur les répertoires

➤ **Affichage du contenu d'un répertoire :**

commande : `ls [options] <nom_répertoire>`

l'option **-l** permet d'obtenir l'ensemble des informations relatives à chaque fichier du répertoire :

- type de fichier : "-" (fichier ordinaire),
"d" (répertoire),
"b ou c" (fichiers spéciaux)
- droits d'accès
- nom du propriétaire
- nombre de liens
- taille
- nom
- date de création
-

L'option **-R** permet d'afficher récursivement le contenu d'un répertoire.

Exemple: `ls /` (afficher les répertoires à la racine)

`dev bin usr users etc unix ...`

`ls -R /`

`/dev :`

liste des fichiers dans /dev

`/bin :`

liste des fichiers dans /bin

`/users :`

`/deustiosi :`

`/iosi01 :`

`/iosi02`

`/minfo :`

`/minf01`

Principales commandes sur les fichiers

❑ Opérations de base sur les répertoires

➤ **Création d'un répertoire**

mkdir <nom_répertoire>

rmdir <nom_répertoire> Supprime un répertoire vide.

rm -R <nom_répertoire> supprime tout le répertoire.

➤ **Montage d'un système de fichiers**

Il est possible d'ajouter un système de fichiers extérieur en l'insérant dans la hiérarchie.

les commandes *mount* et *unmount* permettent de démonter et de démonter un système de fichiers.

=> L'ajout d'une mémoire de masse est transparent pour l'utilisateur.

Principales commandes sur les fichiers

❑ Opérations de base sur les fichiers

➤ Visualisation du contenu d'un fichier

cat <nom_fichier>

more <nom_fichier>

➤ Renommage et déplacement d'un fichier

mv <source> <destination>

mv essai.c tp1.c (renommage)

mv tp TPSE (déplacement)

mv tp TPSE/tp2 (déplacement et renommage)

➤ Copie d'un fichier

cp <source> <destination>

➤ Création de liens sur un fichier

ln <ancien> <nouveau>

➤ Suppression d'un fichier

rm <nom_fichier>

Ex : rm *.o supprimer tous les fichiers
 d'extension ".o"

rm a* supprimer tous les fichiers dont
 le nom commence par "a"

rm * supprime tous les fichiers

rm *.* supprimer tous les fichiers ayant
 une extension d'une lettre.

* : une chaîne quelconque de caractères

? : un caractère quelconque

AUTRES COMMANDES UTILES

touch créer un fichier

wc donne le nombre de caractères (-c),
de mots (-w) ou de lignes (-l)

sort permet de trier par ordre alphabétique
les lignes d'un fichier.

grep recherche d'un motif dans un fichier

Exemple: `grep printfessai.c`

`grep -l printf *.c` (affiche la liste
des fichiers contenant "printf").

head afficher les premières lignes

tail afficher les dernières lignes

diff permet de comparer deux fichiers

find permet de rechercher un fichier

lpr imprimer un fichier

lpq afficher les fichiers en attente
d'impression.

lprm détruire des fichiers en attente
d'impression.

man(Très utile) donne le manuel
d'utilisation d'une commande.

PROTECTION DES FICHIERS

- ❑ L'accès aux fichiers est déterminé par trois bits de permission: **r w x** (Read, Write, eXecute) applicables à trois classes d'utilisateurs : **u g o** le propriétaire, le groupe et les autres (Users, Group, Others).

➤ mode symbolique

chmod <qui><permission><opération><fichier>

<qui> valant :

u: utilisateur **g**: groupe **o**:autres et **a** :tous

<permission> :

+ : pour autoriser **-** : pour interdire

<opération> :

r : lecture **w** : écriture **x** : exécution

Exemples

`chmod g+w montp.c` (les membres du groupe peuvent écrire dans le fichier "montp.c")

`chmod og-rwx montp.c` (protection en lecture, écriture et exécution)

➤ mode octal

chmod <permission><fichier>

permission : **U**G**O** (**U**ser, **G**roup , **O**thers : chiffre octal codant les bits **r w x**)

Exemple

`chmod 740 montp`

(rend le fichier accessible en lecture au groupe et inaccessible aux autres)

LES PROCESSUS UNIX

- ❑ Le processus est l'entité d'exécution dans le système UNIX.

Toute activité dans UNIX correspond
à un processus

- ❑ Dans UNIX, il existe deux types de processus :

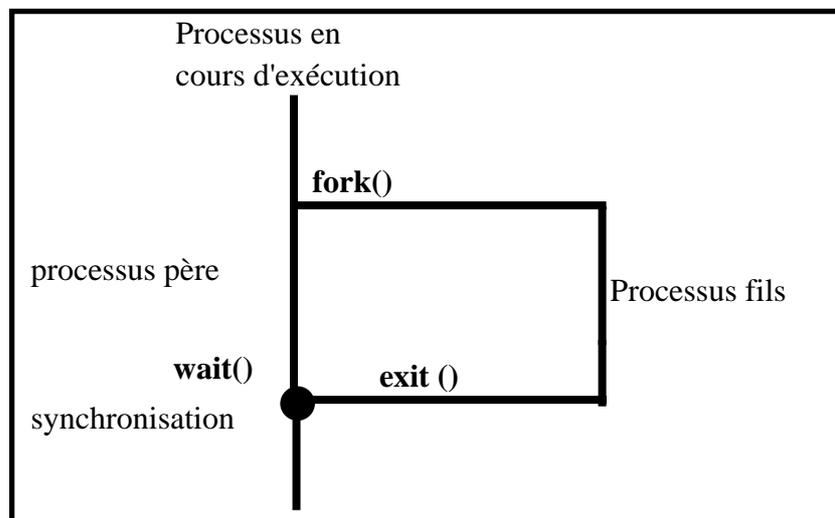
- Processus système
(Ex : swapper, crons, getty)
- Processus utilisateur qui correspondent à l'exécution d'une commande ou d'une application.

- ❑ Chaque processus est identifié par un numéro : PID
(*Process IDentifier*)

- ❑ La commande **ps** donne la liste des processus

LES PROCESSUS UNIX(*suite*)

- ❑ La création d'un processus se fait par dédoublement : Un processus est toujours créé par un autre processus par le mécanisme de fourche "fork".
- ❑ On appelle le processus créateur processus père. Le processus créé est appelé processus fils .

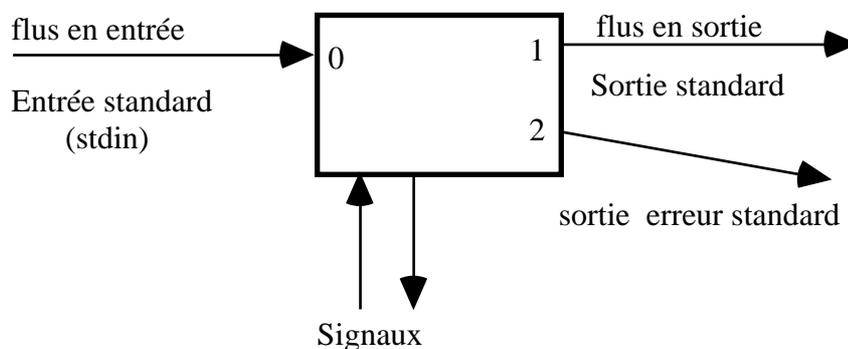


- ❑ L'appel système FORK crée une copie exacte du processus original (L'appelant).
- ❑ La valeur de retour de la fonction FORK est 0 pour le processus fils et est égale au PID du fils dans le processus père.
- ❑ Le processus fils peut exécuter un nouveau code à l'aide des primitives EXEC.

LES PROCESSUS UNIX(*suite*)

❑ Les fichiers standard et leur redirection

On peut assimiler un processus à un objet abstrait qui dispose de 3 flus d'informations :



Redirection de l'entrée standard

- commande < nom_de_fichier

L'entrée standard est redirigée sur le fichier de référence *nom_de_fichier*.

Redirection de la sortie standard

- commande > nom_de_fichier

- commande >> nom_de_fichier
(redirection sans écrasement)

Redirection de la sortie en erreur standard

- commande 2> nom_de_fichier (en Bourne Shell)

- commande >& nom_de_fichier (en c-shell)

❑ Espace d'adressage d'un processus

Tout processus UNIX a un espace d'adressage constitué de trois segments : code, données et pile :

code	Les instructions qui forment le programme
pile	pile d'exécution
données	données statiques et dynamiques du programme

❑ Contexte d'un processus

Le contexte d'un processus comprend :

- le contenu de son espace d'adressage ,
- les contenus des registres matériels,
- les structures de données du noyau qui ont un rapport avec le processus.

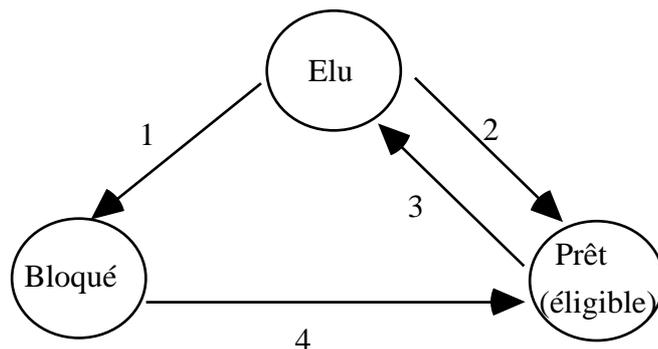
❑ Le système effectue un changement de contexte dans les circonstances :

- le processus se met lui-même en sommeil
- le processus fait un exit

- le processus revient d'un appel système au mode utilisateur alors qu'il n'est pas le processus le plus éligible en vue d'une exécution.

□ Etats et transitions d'un processus

La durée de vie d'un processus peut être conceptuellement divisée en 3 principaux états qui décrivent le processus:



L'état **Elu** : le processus dispose de toutes les ressources dont il a besoin.

L'état **bloqué** : le processus dispose de toutes les ressources dont il a besoin à l'exception de l'UC.

L'état **Bloqué** : le processus se met en attente d'un évènement extérieur.

Un processus Unix passe par d'autres états intermédiaires :

- processus s'exécutant en mode noyau,
- processus s'exécutant en mode utilisateur,
- processus résidant en mémoire principale,
- processus en cours de création
- processus en cours de terminaison (Zombie).
- . . .

Création des processus dans UNIX

