

IN3R11-1 – IN3R11-2 – IN3R21

Cours 1 : Introduction à UNIX, concepts et commandes de base

Damien MASSON
d.masson@esiee.fr

<http://esiee.fr/~massond/Teaching/2011-12/>

Dernière modification : 31 août 2011

La programmation en I3FR

- Trois unités :
 - IN3R11-1 (C/Java)
 - IN3R11-2 (C/Java)
 - IN3R21 (Système d'exploitation / shell)
- 1 chargé de cours : Damien Masson (moi)
- 1 site web : <http://esiee.fr/~massond/>
- Refonte des trois unités
 - but : apprendre plus doucement, et sur toute l'année
 - vous essayez les plâtres : vos retours seront les bienvenus
- Évaluation :
 - 1 tp noté fin novembre
 - 1 contrôle fin janvier
 - 1 mini projet, à commencer en TP, à rendre début mai
 - 1 examen en juin

Système d'exploitation

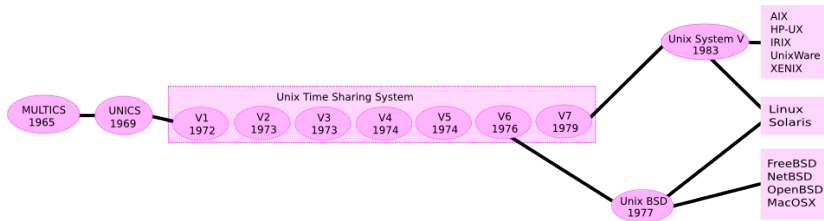
Le système d'exploitation est l'interface logicielle entre *le matériel* (hardware) et *les programmes de l'utilisateur*.

Il gère :

- la mémoire ;
- les périphériques ;
- le(s) processeur(s) ;
- les processus ;
- les entrées/sorties ;
- etc.

Nous allons étudier dans ce cours les systèmes UNIX

UNICS, UNIX, LINUX...

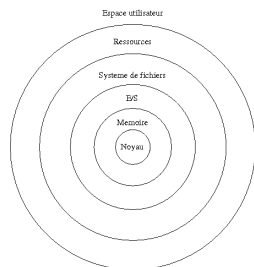


- 1969 : Unics, laboratoires BELL, en assembleur, distribué aux universités avec le code source
- besoins de portabilité conduisent à l'invention du C
- UNIX est réécrit en C et est le premier système d'exploitation portable
- depuis 1988, tentatives de standardisation par les normes POSIX

Caractéristiques d'UNIX

- multi-utilisateurs
- multi-tâches
- protection du système (couches)
- tout est fichier
- contrôle d'accès aux fichiers, donc aux ressources

Architecture en couches



- noyau : interface matériel / logiciel ;
- gestion de la mémoire ;
- gestion des entrées/sorties ;
- gestion des fichiers ;
- gestion de la répartition des ressources entre les processus.

L'utilisateur, pour accéder aux fichiers, ou pour créer des processus, doit donc réaliser des appels systèmes.

Utilisateur

Un utilisateur doit être enregistré auprès du système. Un utilisateur est défini par :

- un identifiant (login) ;
- un mot de passe (password) ;
- un identifiant numérique (uid) ;

En général :

- il dispose d'un répertoire de travail, sous-répertoire de `/home` ;
- il appartient à un ou plusieurs groupes d'utilisateurs.

Sessions

Pour interagir avec le système, l'utilisateur ouvre une *session utilisateur* dans une *console*. Sous linux, les utilisateurs disposent en général de :

- 6 consoles texte [Ctrl+Alt+F_{1...6}];
- 1 console graphique [Ctrl+Alt+F7].

Dans la console graphique s'exécute un gestionnaire de session (kdm sur les machines de l'école). Lorsque l'on se loge, on démarre un gestionnaire de bureau et de fenêtre (kde par défaut). On peut ensuite démarrer un émulateur de terminal, comme `konsole`.

Invite de commande

```
user@machine:~/directory/$
```

Une invite de commande (prompt) récapitule en général

- l'utilisateur connecté au terminal : `user` ;
- le nom de la machine à laquelle est connectée le terminal : `machine` ;
- le répertoire de travail courant : `directory`.

Le terminal, ou console, permet à l'utilisateur de démarrer des applications.

Exemples d'applications

- compilateurs : gcc, g++, javac...
- éditeurs : gvim, emacs, gedit...
- bureautique : L^AT_EX, GhostView, kpdf, OpenOffice...
- internet : Firefox, Thunderbird, Pidgin...
- images et multimédia : The Gimp, display, GQView, VLC...

Par défaut, le terminal attend la terminaison de l'application. Pour lancer une application en tâche de fond : `&`.

On peut suspendre une application avec des signaux : ex `[Ctrl+z]` pour "zombifier"

Commandes internes du terminal

- cd : change directory, change de repertoire de travail ;
- ls : affiche le contenu d'un repertoire ;
- man : affiche de l'aide sur une application ou une commande interne du terminal ;
- mkdir : créé un repertoire ;
- touch : créé un fichier ou modifie sa date de dernière modification ;
- rm/rmdir : efface un fichier/une repertoire ;
- bg,fg : passe en tâche de fond / ramène au premier plan ;
- logout : termine la session ;
- exit : termine la session et ferme le terminal ;
- ...

Options

Il est possible de spécifier des options aux applications, comme aux commandes internes. ex. `ls -a` ou `ls --all`

Pour connaître les options disponibles :

- `man cmd` (recherche dans le manuel avec /)
- `cmd --help` (parfois `cmd -h`)
- `info cmd`

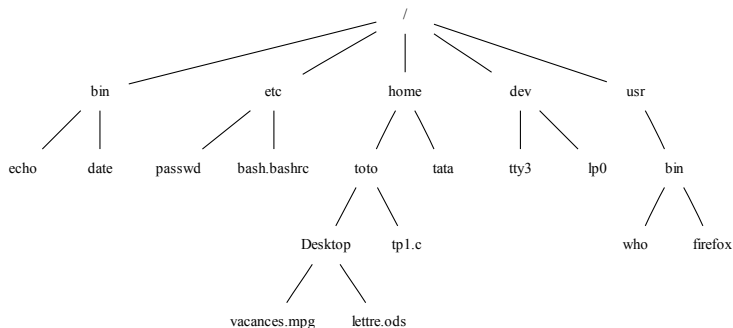
Tout est fichier

Tout est fichier : les fichiers, les répertoires, les imprimantes, les cartes internes et externes, les lecteurs de disques etc.

Arbre des fichiers

Les fichiers sont présentés à l'utilisateur sous la forme d'un arbre (presque).

- racine : /
- nœuds internes : répertoires
- feuilles : fichiers ordinaires / spéciaux (périphériques...).



Références d'un fichier

Il y a trois manières de référencer un fichier :

- *Référence absolue* à partir de la racine : /
/home/toto
/home/toto/Desktop/lettre.odt
/dev/tty3
- *Référence relative* à partir de répertoire de travail courant : ./,
ou son parent : ../
lettre.odt
./lettre.odt
../tp1.c
- *Référence relative* à partir du répertoire de travail de
l'utilisateur courant : ~/ ou d'un autre utilisateur : ~user/
~/Desktop/lettre.odt
~toto/Desktop/lettre.odt

I-nœuds

Le système maintient une *table des i-nœuds* dont les entrées sont :

- un numéro d'i-nœud ;
- les adresses des blocs du disque où est stocké le fichier ;
- la taille ;
- le propriétaire, les droits d'accès ;
- le type (ordinaire, spécial, répertoire, lien symbolique) ;
- nombre de liens directs ;
- ...

Un fichier peut être à plusieurs endroits, il garde un seul i-nœud.

Liens directs / Liens symboliques

- Lien direct : c'est le même fichier, il peut avoir un nom différent, être à un endroit différent dans l'arbre, il n'est stocké qu'une seule fois en mémoire et a un seul i-nœud ;
- Lien symbolique : c'est un fichier différent qui référence un autre fichier.

Si on efface un fichier, cela n'a aucun effet sur un lien direct (diminue le compteur de lien dans la table des i-nœuds), mais rend obsolète tous les liens symboliques.

Manipulation de fichiers

- Manipulation de fichiers
 - `ln` (link) : créer un lien direct ou symbolique (-s) ;
 - `cp` (copy) : copie un fichier ;
 - `mv` (move) : déplace un fichier ;
 - `rm` (remove) : supprime un fichier.
- Manipulation de répertoires
 - `pwd` (print working directory) : affiche le rép. courant ;
 - `cd` (change directory) : change le repertoire courant ;
 - `mkdir` (make directory) : créé un nouveau repertoire ;
 - `rmdir` (remove directory) : enlève un repertoire (vide).

Contenu des répertoires

- `ls`
- `ls -a` : fichiers cachés
- `ls -i` : numéros des i-nœuds
- `ls -l` : informations complètes

Substitutions

Caractères spéciaux désignant un ensemble de fichier :

- * : toute chaîne de caractère correspondant à un nom de fichier
- ? : un seul caractère
- [gbzm] : un caractère parmi ceux donnés
- [A-G] : un caractère dans l'intervalle donné
- [^h] : un caractère différent de celui donné

Fichiers spéciaux

→ console, imprimante, lecteur de disquette, clé usb, partition de disque dur, répertoire distant. . .

- dans /dev
- taille nulle
- Le i-nœud contient :
 - propriétaire, groupe, droits d'accès
 - *majeur* : classe de ressource associée
 - *mineur* : exemplaire particulier de cette classe
 - *mode* : bloc ou caractère spécifiant l'accès (lecture/écriture, lecture seule...)
- un répertoire est une liste de couple i-nœud / nom de fichier

Affichage de texte

- cat : brut
- more : paginé
- less : déplacement dans le texte

Recherche de fichiers

- `find` : recherche dans une sous-arborescence
- `locate` : recherche partout un fichier contenant une chaîne donnée, via une table précalculée
- `updatedb` : met à jour la table

Archivage

- gunzip
- tar

Partitions

- `cfdisk` : séparer un disque en partitions
- `mkfs` : formater une partition
- `/proc/partitions` : partitions des disques branchés
- `mount` : monter une partition
- `df` : espace disponible (ne pas confondre avec *du*)

Monter/démonter

Chaque disque : table d'i-nœuds distincts

Disque monté : son répertoire racine est branché sur un répertoire de l'arbre

- `mount` : monter un disque
- `umount` : démonter
- `/etc/mtab` : liste des partitions montées
- `/etc/fstab` : montages à faire au démarrage ou autorisés à chaud pour des groupes d'utilisateur.

Droits d'accès

4 catégories d'utilisateurs :

- propriétaire du fichier (u)
- groupe du propriétaire (g)
- autres (o)
- super utilisateur (root) : tous les droits.

3 types de droits :

- *lecture* (r)
- *écriture* (w)
- *exécution* (x).

chmod : change les droits de fichiers (dont on est propriétaire)

chmod

- représentation textuelle $\{u|g|o|a\}\{+|- \}\{r|w|x\}$
 - u user
 - g group
 - o other
 - a all
- représentation octale : trois chiffres codant les trois permissions.

Droit	Valeur alphanumérique	Valeur octale
aucun	---	0
exécution seule	--x	1
écriture seule	-w-	2
écriture et exécution	-wx	3
lecture seule	r--	4
lecture et exécution	rx-	5
lecture et écriture	rw-	6
tous	rxw	7

Processus

Processus : *exécution* d'un programme, d'une suite d'instructions.

Système multitâche : exécution concurrente de plusieurs processus.

- *ordonnanceur* (processus 0) : alloue régulièrement le processeur à un processus
- *initiateur* (processus 1) : associe à chaque terminal un processus
- *démons* : tournent en permanence
- processus *fil* (`fork`) : c'est la seule façon de créer un nouveau processus.

Liste des processus

- `pstree` : arbre généalogique des processus
- `ps` : processus fils
- `ps -a` : de tous les utilisateurs
- `ps -u` : d'un utilisateur donné
- `ps -l` : informations complètes
- `top` : ressources utilisées

Tâches

jobs : gestion des processus fils sans solliciter le noyau (ps est une commande gourmande)

États des processus

- *actif* : exécuté par le processeur
- *prêt* : attend que le système lui attribue le processeur
- *bloqué* : attend une ressource

Deux modes d'exécution : mode utilisateur et mode noyau
(correspond à l'exécution d'un appel système par le processus)

Signaux

- envoyés à un processus
- depuis un autre processus ou le clavier
- enregistré dans une table contenue dans le bloc de contrôle du processus
- traitées par le processus

Signaux

- `kill` : envoyer un signal
- `trap` : exécute une commande à la réception d'un signal

Quelques signaux :

- 2 (`INT`) : interruption
- 9 (`KILL`) : fin
- 11 (`SEGV`) : référence mémoire invalide
- 17 (`CHLD`) : terminaison d'un fils
- 29 (`WINCH`) : redimensionnement de fenêtre
- ...

Caractères de contrôle

Le terminal envoie les sigaux correspondant au processus qui est au premier plan.

- `erase` : efface le dernier caractère, [backspace]
- `kill` : efface la ligne, [Ctrl-u]
- `eof` : fin de fichier, [Ctrl-d]
- `newline` : valide une commande, [Ctrl-j]
- `intr` : interruption, [Ctrl-c]
- `quit` : interruption avec image mémoire, [Ctrl-\]
- `stop` : suspension, [Ctrl-z]

Mémoire

Les processus ont des besoins en temps processeur, mais aussi en mémoire. Chaque processus dispose de sa propre zone mémoire, inaccessible aux autres processus.

- Tant que la mémoire centrale peut contenir toutes les demandes, tout va bien ;
- En cas de saturation, priorités d'accès à la mémoire dépendantes des priorités d'exécution ;
- Les processus endormis sont "swappés" .

En réalité, chaque processus opère sur de la mémoire virtuelle (il considère qu'il dispose de toute la mémoire) que le système fait coïncider avec la mémoire physique.

Pour limiter les accès aux disques physiques, mécanisme de pagination.

Entrées/Sorties

Tous les processus disposent de trois *fichiers standards*, seuls moyen de communication entre les processus (avec les signaux) :

- l'*entrée standard* `stdin`
- la *sortie standard* `stdout`
- la *sortie d'erreur standard* `stderr`

Par défaut : clavier et écran

Redirections

Redirection : consiste à spécifier des fichiers différents pour les entrées/sorties. Exemples :

```
date > tutu, date >> tutu, wc -w < tutu, ls tata 2>  
tutu, ls tutu tata &> tonton
```

Descripteurs de fichiers

sur l'exemple précédent l'erreur standard a été désigné par un *descripteur*

- une table pour chaque processus
 - identifie chaque fichier manipulé par un numéro
 - pointe vers la table des fichiers ouverts, qui pointe vers la table des i-nœuds
 - héritée par les processus fils
-
- L'entrée standard a comme descripteur 0.
 - La sortie standard a comme descripteur 1.
 - La sortie erreur standard a comme descripteur 2.

Enchaînement de processus

- ; : plusieurs commandes séquentiellement
- | : plusieurs commandes simultanément
qui communiquent via un tube (sortie de l'un = entrée de l'autre)

Exemple : `date; who; ps tp2 / cat tutu | wc`

Filtres

- lit l'entrée standard
- écrit sur la sortie standard
- `cat` : retranscription exacte → [Ctrl-d]
- `head` / `tail` : début / fin du texte exemple : `head -4 < wtc`
| `tail +3` / `tr [a-z] [A-Z]` / `sed s/ek/ic/`
- `tr` : remplacement d'un caractère
- `cut` : sélection de caractères / champs
- `grep` : recherche dans un fichier