

UNIX - Introduction

D. Bogdaniuk – P. Portejoie

Antoine Gicquel**07/09/16****Info1 - A2**

Présentation

Unix est né dans les laboratoires BELL qui dépendent d'AT&T (American Telephone & Telegraph) sous l'impulsion de 2 personnes :

- Ken Thompson
- Dennis Ritchie

Unix désigne une famille de systèmes d'exploitation non propriétaires, issus plus ou moins directement de différentes versions diffusées par AT&T. Certains membres de cette famille sont de complètes réécritures qui ne doivent rien à AT&T tout en restant compatibles au niveau des interfaces.

L'originalité de la démarche ayant présidé à sa conception repose sur deux aspects :

- le système a été conçu par des utilisateurs pour leurs propres besoins
- la conception et le développement ont été faits par une très petite équipe de très haut niveau scientifique.

Une première version, mono-utilisateur, a été réalisée en assembleur au cours des années 1969-1970, suivie dès 1971 d'une nouvelle version autorisant la multiprogrammation. L'objectif initial était essentiellement de fournir sur des petites machines un système interactif offrant un environnement de programmation comparable à ceux des grands systèmes.

Unix fut réécrit pour 90% en C en 1973. C'était le premier système d'exploitation à ne pas être entièrement écrit en assembleur. C'était aussi le premier système, en principe, portable.

L'année 1975 marque la date de première commercialisation du système.

Le système Unix est maintenant devenu une "norme" en matière de SE et a été installé sur de nombreuses machines qui vont du micro-ordinateur au super-calculateur. Il est "simple" et "facile" à utiliser, propose un bon environnement de programmation et possède des outils dont ne disposent pas toujours des systèmes beaucoup plus performants.

Unix s'appuie sur trois grands principes :

1 - Le système de fichiers

Le système de fichiers permet à l'utilisateur de conserver les informations en les nommant. Il doit supporter les pannes matérielles et doit comporter un mécanisme de sécurité contre les accès non autorisés. Il distingue trois types de fichiers différents :

- le fichier ordinaire qui est une suite de caractères
- le répertoire qui contient des noms de fichiers ou de répertoires
- les fichiers spéciaux qui correspondent aux périphériques d'entrée/sortie.

2 - Les processus

Tous les travaux des utilisateurs sont réalisés par des processus.

Un processus est une séquence d'actions qui est caractérisée par la taille mémoire nécessaire à son exécution et par les fichiers auxquels il accède.

3 - Le shell

Le shell est un langage de programmation qui permet d'accéder au SE. Il exécute des commandes qui peuvent provenir soit d'un terminal, soit d'un fichier.

Unix présente les caractéristiques suivantes :

- multi-utilisateurs
plusieurs utilisateurs connectés simultanément se partagent les ressources par découpage du temps et mise à profit des temps morts d'entrées/sorties ; le système possède un mécanisme d'identification et de protection des utilisateurs.
- multi-tâches
un utilisateur peut lancer plusieurs tâches en même temps
- système de fichiers hiérarchisé arborescent
les entrées/sorties sont généralisées, vues par l'utilisateur à travers des périphériques considérés comme des fichiers

Principales commandes

Connexion/Deconnexion

login	connexion login [nom utilisateur]
exit	quitte l'interpréteur de commandes et permet une déconnexion exit
su	changement d'identité su [nom_utilisateur]

Manipulation de répertoires

pwd	affiche le nom du répertoire courant pwd
cd	change le répertoire courant cd [[chemin d'accès] répertoire]
mkdir	crée un répertoire mkdir [chemin d'accès] répertoire
rmdir	supprime un sous-répertoire (s'il est vide) rmdir [chemin_d'accès] répertoire
ls	affiche la liste des fichiers et sous-répertoires d'un répertoire ls [chemin d'accès]
pushd	empile le chemin du répertoire courant pushd .
popd	dépile le chemin du répertoire enregistré avec pushd et y retourne popd

Manipulation de fichiers

cat	affiche le contenu d'un fichier texte cat [chemin d'accès [fichier]]
touch	change les dates d'accès et de modification d'un fichier ou en crée un nouveau

chgrp	changement du groupe d'un fichier chgrp groupe [chemin d'accès] fichier
chmod	modifie la protection (droits) d'un fichier chmod mode [chemin d'accès] fichier
chown	changement du propriétaire d'un fichier chown propriétaire [chemin_d'accès] fichier
cp	copie un fichier source dans un fichier destination cp [chemin_d'accès] f_source {chemin_d'accès [f destination] f_destination}
cut	découpe une ligne en champ numéroté de 1 à n suivant un délimiteur cut -d[délimiteur] -f[numéro du champ] [chemin d'accès] fichier
head	affiche le début d'un fichier head [-nombre de lignes] [chemin d'accès] fichier
more	affiche le contenu d'un fichier texte page par page more [chemin d'accès] fichier
mv	change le nom d'un fichier ou le déplace mv [chemin d'accès] f_source {chemin_d'accès [f_destination] f_destination}
od	visualisation d'un fichier sous forme octale, hexadécimale, caractère od [chemin_d'accès] fichier
pg	similaire à more
rm	supprime un ou plusieurs fichiers, et des sous-répertoires rm [chemin d'accès] fichier
tail	affiche le début ou la fin d'un fichier tail [{+ -}nombre] [lbc] [chemin d'accès] fichier
umask	positionnement des droits par défaut umask [mode]
wc	compte les mots, lignes ou caractères d'un fichier wc [chemin d'accès] fichier

Gestion des processus

ps	affichage les informations sur les processus en mémoire ps
kill	envoie d'un signal kill [-signal] n°_processus

Gestion des utilisateurs

who	affiche tous les utilisateurs connectés sur une machine who
whoami	renvoie les information sur l'utilisateur connecté whoami
groups	affiche les groupes d'appartenance de l'utilisateur connecté

Commande diverses

man	affiche de l'information sur les commandes Unix man commande
clear	efface l'écran clear

env	affiche les variables d'environnement env
setenv	permet de définir des variables d'environnement setenv var chaîne
alias	définition de synonymes alias [nom_alias=commande]
ln	lie un fichier à un autre ln [chemin_d'accès] fichier lié [[chemin_d'accès] lien]
at	exécution différée d'une commande at heure [jour] {fichier commande}
grep	recherche d'une chaîne dans un fichier fichier grep expression [[chemin_d'accès] fichier]
diff	compare les contenus de deux fichiers diff [chemin d'accès] fichier 1 [chemin d'accès] fichier2
date	affiche ou modifie la date courante date
time	affiche l'heure du système ou le temps d'exécution d'une commande time [commande]
ed, vi, emacs	lance l'éditeur
sed	éditeur en mode ligne de commande
du	espace occupé sur le disque; l'option -h permet d'exprimer le résultat en Ko
find	recherche d'un fichier
more	filtre permettant l'affichage page par page
sort	filtre permettant l'affichage selon un critère de tri
lpr	spooler (libère l'ordinateur pendant l'impression) l'option -P permet de spécifier une imprimante
lpq	file d'attente de l'imprimante
lprm	suppression d'une tâche d'impression
tar	archivage
fdisk	partitionne un disque dur
format	formate une disquette ou le disque dur pour utilisation avec MSDOS

Nommage des fichiers et des répertoires

Les noms sont limités à 14 caractères au plus et une différenciation minuscule/majuscule est faite.

Les caractères à proscrire sont les suivants :

\ > < | \$? & [] * ! " ' ' () @ ~

Le point "." joue un rôle particulier dans la mesure où les fichiers dont le nom commence par lui sont "invisibles".

TD - TP

- 1/ a/ Expliquez le sens de la forme [...] utilisée pour certaines commandes ; illustrez votre propos par 2 cas d'utilisation de l'instruction *time [commande]* en prenant *ls* comme commande.

[...] est utilisé pour illustrer une option facultative.

Par exemple : *time [option] [COMMANDE]*

C'est à dire que ces deux commandes sont valides :

> *time*

> *time ls*

- b/ Dans le même esprit analysez la description de la commande *at* et déduisez-en le sens de { } et de |

at diffère l'exécution d'une commande : *at HEURE {fichier | commande}*

HEURE n'est pas entre crochet donc cet argument n'est pas facultatif.

{...} désigne plusieurs options dont au moins une obligatoire et | signifie le ou logique donc la commande *at* peut s'utiliser *at HEURE fichier* ou *AT Heure commande*.

- 2/ Indiquez la validité ou l'invalidité des différents identificateurs de fichier proposés ci-après (en cas d'invalidité indiquez-en la cause) :

essai.txt *Essai* *123essai.txt* *<essai>* *.123essai*

essai.txt → Valide.

Essai → Valide mais l'absence d'extension à la fin du fichier le rend moins pratique.

123essai.txt → Valide.

<essai> → Invalide car le nom du fichier commence par un caractère spécial (<).

.123essai → Valide mais sera considéré comme fichier caché sous système UNIX, et ne possède pas d'extension.

- 3/ Familiarisation avec l'usage des commandes UNIX : pour chacune des commandes données ci-après, précisez la signification de son abréviation (tout d'abord en anglais, puis en français) puis décrivez son usage.

man, *clear*, *cp*, *rm*, *mv*, *pwd*, *cd*, *mkdir*, *rmdir*, *ls*, *cat*, *touch*

Exemple : *man* est mis pour *manual* (manuel) et lance l'aide en ligne

Vous donnerez le résultat sous la forme suivante :

Nom de commande	Nom complet	Traduction	Rôle
<i>man</i>	<i>manual</i>	manuel	Aide en ligne
<i>clear</i>	<i>clear</i>	nettoyer	Effacer l'écran du terminal
<i>cp</i>	<i>copy</i>	copier	Copier des fichiers et des répertoires

rm	remove	supprimer	Effacer des fichiers
mv	move	déplacer	Déplacer ou renommer des fichiers
pwd	print working directory	affiche le répertoire courant	Afficher le nom du répertoire de travail en cours
cd	current directory	répertoire courant	Changer le répertoire de travail en cours
mkdir	make directory	créer un répertoire	Créer des répertoires
rmdir	remove directory	déplacer un répertoire	Supprimer des répertoires
ls	list segments	liste les segments	Afficher le contenu d'un répertoire
cat	catenate	concaténer	Concaténer des fichiers et les afficher sur la sortie standard
touch	touch	toucher	Modifier l'horodatage d'un fichier

En TP les exercices suivants seront bien évidemment à réaliser en ligne de commande dans un terminal (il vous est recommandé d'utiliser l'application *Terminal* ainsi que *Ksnapshot* pour les copies d'écran). En TD, au fur et à mesure, vous illustrerez graphiquement votre travail en construisant l'arborescence du système de fichiers correspondant. On considère que la position actuelle est votre répertoire d'accueil.

4/ Créez à partir de votre répertoire courant, un répertoire appelé *SYS*

5/ Déplacez-vous dans le répertoire *SYS* et créez-y 2 répertoires appelés *Documents* et *Images*

6/ Créez un fichier appelé *mondoc.txt* dans le répertoire *Documents*

7/ Affichez le contenu du fichier *mondoc.txt* (souvenez-vous qu'il est vide...)

8/ Créez un fichier appelé *image.png* dans le répertoire *Documents*

9/ Copiez le fichier *image.png* du répertoire *Documents* dans le répertoire *Images*

10/ Supprimez le fichier *image.png* du répertoire *Documents*

11/ Affichez le contenu du répertoire *Documents*

12/ Sans changer de répertoire, affichez le contenu du répertoire *Images*

Capture d'écran du terminal :

```
[e1600718@ens-iutva-0408 e1600718]$ ls
Bureau  datamodeler  Documents  firefox  gradle-app.setting  Images  'Mes documents'
Modèles  Musique  Public  Sqldeveloper  Téléchargements  TPs  Vidéos
[e1600718@ens-iutva-0408 e1600718]$ mkdir ~/SYS
[e1600718@ens-iutva-0408 e1600718]$ cd ~/SYS/
[e1600718@ens-iutva-0408 SYS]$ mkdir Documents Images
[e1600718@ens-iutva-0408 SYS]$ touch Documents/mondoc.txt
[e1600718@ens-iutva-0408 SYS]$ more Documents/mondoc.txt
[e1600718@ens-iutva-0408 SYS]$ touch Documents/image.png
[e1600718@ens-iutva-0408 SYS]$ cp Documents/image.png Images/
[e1600718@ens-iutva-0408 SYS]$ rm Documents/image.png
[e1600718@ens-iutva-0408 SYS]$ ls Documents/
mondoc.txt
[e1600718@ens-iutva-0408 SYS]$ ls Images/
image.png
```

En TP uniquement :

Le point de départ de l'arborescence du système est un répertoire particulier qui s'appelle la racine (/). Votre répertoire d'accueil n'est qu'une sous-partie de l'arborescence globale.

13/ Depuis votre répertoire courant lancez la commande qui permet de visualiser le contenu de la racine du système

14/ Déplacez-vous à la racine du système et lancez cette même commande, mais sans argument. Que constatez-vous ? Concluez

La commande `ls` lancée sans argument affiche le contenu du répertoire courant.

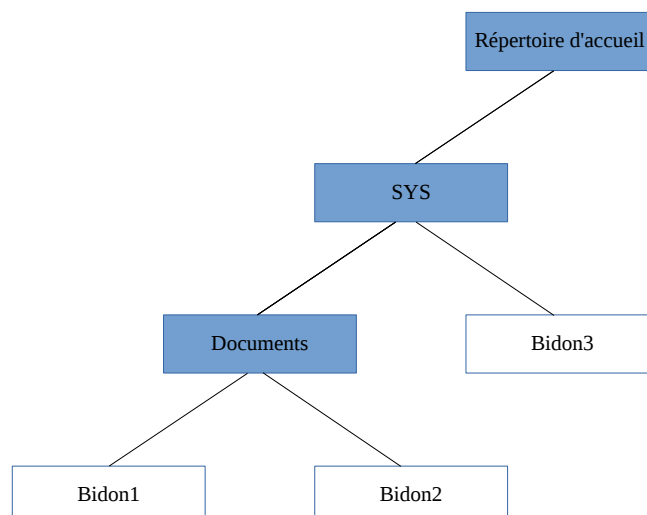
15/ Retournez dans votre répertoire d'accueil en lançant la commande `cd` sans argument, puis, depuis ce répertoire, listez le contenu du répertoire `SYS`

16/ Testez la touche *Tab* pour la complémentation automatique (`ls S Tab` par exemple donnera `ls SYS`), ainsi que les flèches montante et descendante pour le rappel de l'historique des commandes

17/ Lancez une à une les différentes commandes nécessaires pour supprimer de votre espace toute l'arborescence créée au cours de ces différents exercices

18/ Dans votre répertoire d'accueil créez le fichier *essai.txt* puis cherchez un moyen de la détruire **en forçant l'apparition de la demande de confirmation** (pensez à l'aide en ligne)

19/ Créez l'arborescence donnée ci-dessous (bidon1, bidon2 et bidon3 sont des fichiers vides), puis détruisez-la en une seule commande **en forçant l'apparition de la demande de confirmation** (utilisez l'aide en ligne de la commande `rm`)



Capture d'écran du terminal :

```
[e1600718@ens-iutva-0408 SYS]$ ls /
bin boot dev dsi etc home lab lib lib64 lost+found media mnt opt proc
root run sbin srv sys tmp ubs usr var
[e1600718@ens-iutva-0408 SYS]$ cd /
[e1600718@ens-iutva-0408 /]$ ls
bin boot dev dsi etc home lab lib lib64 lost+found media mnt opt proc
root run sbin srv sys tmp ubs usr var
[e1600718@ens-iutva-0408 /]$ cd
[e1600718@ens-iutva-0408 ~]$ ls SYS/
Documents Images
[e1600718@ens-iutva-0408 ~]$ rm -r ~/SYS/
[e1600718@ens-iutva-0408 ~]$ touch essai.txt
[e1600718@ens-iutva-0408 ~]$ rm -i essai.txt
rm : supprimer fichier vide 'essai.txt' ? o
[e1600718@ens-iutva-0408 ~]$ mkdir ~/SYS
[e1600718@ens-iutva-0408 ~]$ mkdir SYS/Documents
[e1600718@ens-iutva-0408 ~]$ cd SYS/
[e1600718@ens-iutva-0408 SYS]$ touch Bidon3 Documents/Bidon1 Documents/Bidon2
[e1600718@ens-iutva-0408 SYS]$ cd
[e1600718@ens-iutva-0408 ~]$ rm -ri ~/SYS/
rm : descendre dans le répertoire '/ubs/home/etud/2016/e1600718/SYS/' ? o
rm : descendre dans le répertoire '/ubs/home/etud/2016/e1600718/SYS/Documents' ? o
rm : supprimer fichier vide '/ubs/home/etud/2016/e1600718/SYS/Documents/Bidon1' ? o
rm : supprimer fichier vide '/ubs/home/etud/2016/e1600718/SYS/Documents/Bidon2' ? o
rm : supprimer répertoire '/ubs/home/etud/2016/e1600718/SYS/Documents' ? o
rm : supprimer fichier vide '/ubs/home/etud/2016/e1600718/SYS/Bidon3' ? o
rm : supprimer répertoire '/ubs/home/etud/2016/e1600718/SYS/' ? o
```