

UNIX - Introduction

X. Roirand – N. Delomez – N. Le Sommer – F. Lesueur

Prénom Nom :

Date :

Groupe :

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 (système d'exploitation) 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.

En résumé, 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 simplifiées (pour plus de détails voir l'Aide en ligne : commande *man* ou *help*) :

Connexion/Deconnexion

login	connexion login [nom utilisateur]
exit	quitte l'interpréteur de commandes et permet une déconnexion exit
su	changement d'identité ("- " force un nouvel environnement) su [-] id_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 et se déplace dans le répertoire indiqué pushd . (empile le chemin du répertoire courant et y reste)
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}
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 groups

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
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]
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
nano, vi, emacs	lance l'éditeur
sed	éditeur en mode ligne de commande
du	espace occupé sur le disque
find	recherche d'un fichier
less, more	filtre permettant l'affichage page par page

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" (ils sont dits "cachés").

TP

Tous les exercices sont à réaliser en ligne de commande dans un terminal (en utilisant l'application *Terminal*). Pour toutes les commandes à taper, il faut taper le nom de la commande, puis <Entrée> ou <Return> afin que l'interpréteur de commande, ici le shell, puisse prendre en compte votre commande.

1/ Analysez la description de la commande *grep* et déduisez-en le sens de [...]

Pour avoir la signification de la commande, utilisez la commande *man*, en tapant :

man ...

ou ... est le nom de la commande pour laquelle on veut le manuel

[...] signifie :

2/ 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 (en général abréviation)	Nom complet	Rôle
man	manual	Aide en ligne
clear		
cp		
rm		
mv		
pwd		
cd		
mkdir		
rmdir		
ls		
cat		

touch		
-------	--	--

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

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

Pour réaliser cela, vous utiliserez la commande *touch* suivi du nom du fichier à tester, puis vous vérifierez que ce fichier a été ou non créé grâce à la commande *ls*.

On considère pour la question 4, que la position de départ est votre répertoire d'accueil. Si vous n'êtes pas sûr de votre répertoire de départ, vous pouvez taper la commande :

cd

4/ Créez à partir de votre répertoire courant, un répertoire appelé *SYS* et vérifiez son existence à l'aide de la commande *ls*.

Indiquez la commande utilisée pour créer le répertoire :

5/ Déplacez-vous dans le répertoire *SYS* et créez-y 2 répertoires appelés *Documents* et *Images*, puis vérifiez leur existence.

Indiquez la ou les commandes utilisées pour créer les 2 répertoires et la commande utilisée pour vérifier l'existence de ces répertoires :

6/ Déplacez-vous dans *Documents* et créez-y un fichier appelé *mondoc.txt*

Pour créer ce fichier qui sera vide, vous pouvez simplement utiliser la commande *touch* qui permet de créer un fichier vide avec le nom que vous voulez. Chercher avec man ou sur internet comment utiliser la commande *touch*.

Indiquez la commande utilisée pour créer le fichier *mondoc.txt* :

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

Pour afficher le contenu d'un fichier, il existe plusieurs commandes. La 1^{ère} affiche le contenu complet du fichier. Donc si il est long l'affichage peut prendre un peu de temps. Les 2 suivantes affichent le fichier avec une pagination. C'est à dire que si le contenu du fichier dépasse la capacité d'affichage de l'écran utilisé (nombre de ligne) alors une pause est faite et il faut appuyer sur entrée pour afficher la ligne suivante ou sur espace pour afficher la page suivante.

Donnez les 3 commandes possibles.

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

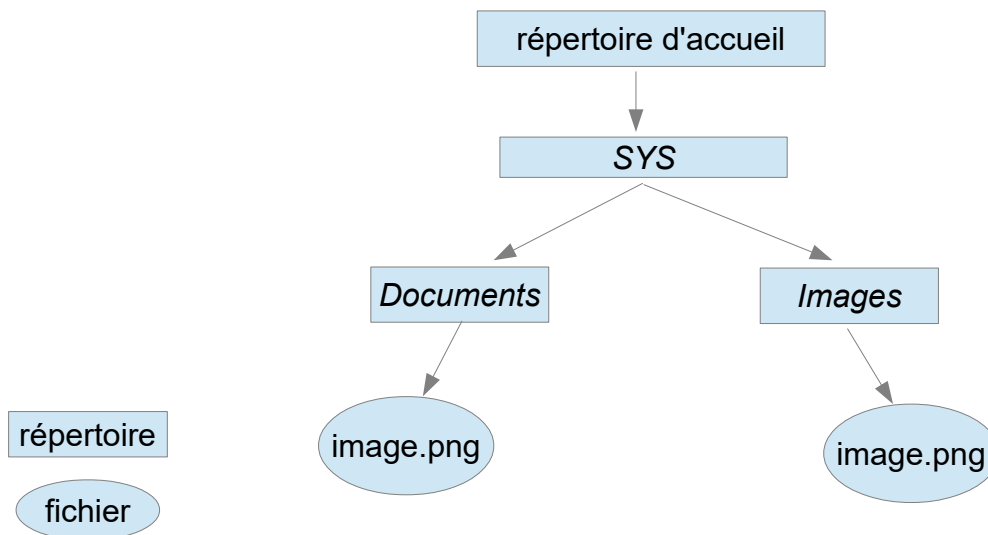
Bien-sûr le fichier *image.png* ne sera pas un vrai fichier image, il aura juste le nom.

Indiquez la commande utilisée pour créer le fichier *image.png* :

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

Indiquez la commande utilisée pour copier le fichier *image.png* :

Après toutes ces commandes que vous avez tapées, l'arborescence de vos répertoires et fichiers peut être dessinée comme ceci :



10/ Afficher l'arborescence complète à partir de votre répertoire d'accueil.

L'utilitaire *tree* permet d'afficher les arborescences complète.

Indiquez la commande utilisée pour afficher l'arborescence :

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

Indiquez la commande utilisée pour supprimer le fichier *image.png* :

12/ Affichez le contenu du répertoire *Documents*

Indiquez la commande utilisée pour afficher le contenu du répertoire :

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

Indiquez la commande utilisée pour afficher le contenu du répertoire *Images* :

14/ Cherchez avec la commande *man* comment afficher le contenu du répertoire *Images* sur une seule colonne

Indiquez la commande utilisée pour afficher le contenu du répertoire *Images* sur une seule colonne :

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.

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

Indiquez la commande utilisée pour afficher le contenu de la racine système :

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

17/ Y a t-il un intérêt à se déplacer dans le répertoire dont on veut lister le contenu ou pas ?

Partie bonus

18/ Créez un fichier *.test* à la racine de votre home grâce à la commande *touch*.

Indiquez la commande utilisée pour créer le fichier :

19/ Utilisez la commande « *ls* » pour afficher le contenu de votre home, expliquez pourquoi vous ne voyez pas le fichier « *.test* » :

20/ Trouvez l'option dans la commande « ls » qui vous permet d'afficher le contenu de votre home mais qui fait bien apparaître le fichier « .test », et indiquez la commande utilisée :

21/ Retournez dans votre répertoire d'accueil aussi appelé « home directory » ou « home » en lançant la commande cd sans argument, puis, depuis ce répertoire, listez le contenu du répertoire SYS

Indiquez la commande utilisée pour afficher le contenu du répertoire SYS:

22/ Déplacez-vous dans SYS et testez la touche *Tab* pour la complémentation automatique (`ls` D *Tab* par exemple donnera `ls Documents`). Faites la même chose pour lister le répertoire Images. Manipulez les flèches montante et descendante pour visualiser l'historique des commandes.

23/ Il existe une commande pour lister l'historique de toutes les commandes que vous avez déjà tapé dans un terminal (voir dans une session ou plusieurs sessions bash précédente). Trouvez cette commande et testez la.

Indiquez le nom et la syntaxe de cette commande.

24/ Essayer de rappeler une des commandes de l'historique, en utilisant « une astuce », qui permet d'exécuter une commande de l'historique sans avoir à retaper cette commande, mais juste son numéro.

Indiquez le nom et la syntaxe de cette commande.

25/ Il existe une autre astuce (proche de la précédente), qui permet de rappeler une commande qui commence par la ou les même lettre qu'une précédente. Expliquez cette astuce.

26/ Expliquez en quoi, dans certains cas, cette astuce peut-être dangereuse.

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

Indiquez les commandes utilisées:

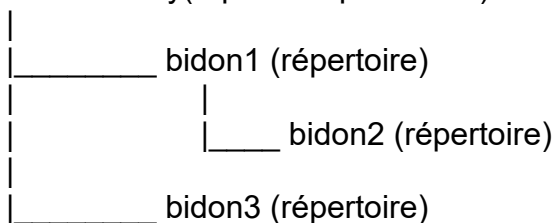
28/ Dans votre répertoire d'accueil créez le fichier *essai.txt* puis cherchez un moyen de le détruire **en forçant l'apparition de la demande de confirmation** (pensez à l'aide en ligne)

Indiquez la commande utilisée :

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

Pensez à regarder le manuel de *rm* (command *man*) pour voir comment supprimer des répertoires.

<home directory(répertoire personnel)>



Indiquez la commande utilisée :