

INTRODUCTION



HISTORIQUE UNIX

Unix est né il y a près de 36 ans, au sein des laboratoires de la compagnie Bell (AT & T). L'historique suivant résume certaines dates et des éléments clés du développement de ce système d'exploitation.

1968 : Ken Thompson et Dennis Ritchie produisent une première version d'un système d'exploitation pour un ordinateur DEC PDP7.

1970 : Brian Kernighan baptise le nouveau système d'exploitation : UNIX.

Fin 1970 : L'achat d'un PDP 11/20 constitue la base pour un projet de traitement de texte et UNIX est transporté dans cette ordinateur.

1971 : Ken Thompson et Denis Ritchie publient la documentation de cette première version.

1972 : Une deuxième version du système est créée pour supporter les tubes (pipes). Thompson qui avait travaillé sur le langage B de BCPL, réécrit l'assembleur de UNIX en B. Or B est un langage interprété. De ce fait, les performances sont médiocres. Le langage B est ensuite muni de types pour supporter les octets du PDP 11 et devient le langage NB. Une tentative de réécriture de UNIX en NB se solde par un échec. Ritchie produit alors un générateur de code exécutable pour le langage NB qu'il nomme C5.

HISTORIQUE UNIX

1973 : Ajout du concept de structures au langage C et réécriture d'UNIX en C. Cette nouvelle mouture est ensuite distribuée par Western Electric aux Universités et écoles, ce qui permet d'étendre la notoriété à la fois du langage C et du système d'exploitation Unix.

1975 : La sixième version d'UNIX est commercialisée pour la première fois.

1977 : Kernighan et Ritchie écrivent la première : « The C programming language » (Prentice-Hall, 1978). Ce livre restera longtemps l'ouvrage de référence du C.

1979 : L'année 79 voit l'amélioration du langage C (intégration des unions, définitions de types et opérateurs de changement de types). La septième version de UNIX est alors transportable sur d'autres machines.

Nota : il existe de nos jours, de multiples versions d'Unix développées par différents fabricants d'ordinateurs. A titre d'exemple, on recense :

- AIX (IBM)
- Oracle Solaris (anciennement Sun solaris)
- HP/UX (HP)
- MAC OSX

HISTORIQUE UNIX

Face à ces multiples déclinaisons d'Unix, des organismes internationaux (Open group) ont été constitués afin d'établir des règles et normes permettant de garantir la portabilité et la compatibilité. Parmi ces organismes, on note :

- POSIX (Portable Operating System Interface UniX), pour la définition des normes de portabilité.

1-Le système d'exploitation UNIX

- 1) UNIX est un système d'exploitation commercial
- 2) Système multi-utilisateurs : plusieurs utilisateur peuvent exploiter ,en même temps mais à partir de terminaux différents, les mêmes applications, le même fichier et des périphériques (imprimantes, disque dur....), en mode local ou distant.
- 3) Système multi-tâches : Plusieurs programmes ou portions de programmes peuvent être exécutés « simultanément » (en parallèle) sur la même (ou sur différentes machines interconnectées en réseau) et partager les ressources du système de traitement de l'information.

Nota : Unix est système fondé sur le concept de Temps partagé (Time sharing).

- 4) Système multi-processeurs : l'unité centrale peut comporter plusieurs processeurs
- 5) Système « ouvert »(par opposition aux systèmes propriétaires).
- 6) Structure modulaire du noyau, permettant une relative extensibilité
- 7) Système basé sur la gestion des processus,

1-Le système d'exploitation UNIX

8) Système sécurisé associé notamment au concept de compte :

- # Identificateur(login)

- # Mot de passe (password)

- # Répertoire principal(home directory)

- # permissions

9) Système comportant de nombreuses commandes avec de multiple options (voir concept de « shell »).

10) La connexion au système (ouverture d'une session)

- # Login

- # password

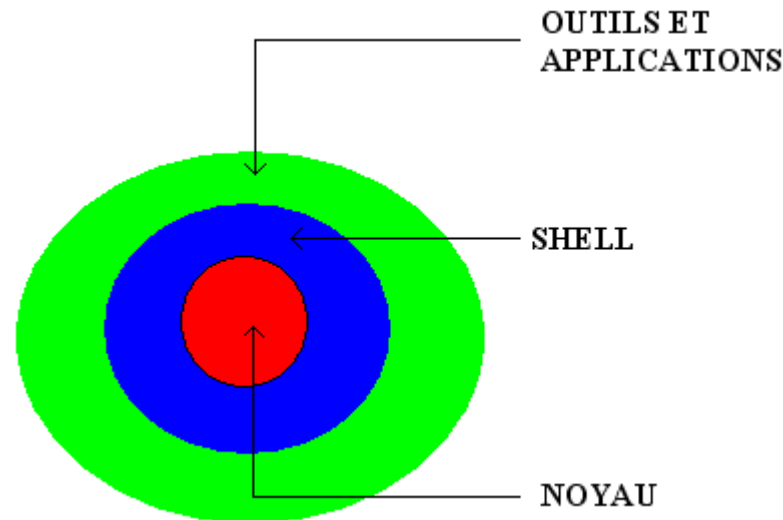
11) Système opérant une différence entre les majuscules et les minuscules,

2-Structure d'UNIX

12) Système possédant une organisation des fichiers sous forme d'arborescence et associée au concept de montage / démontage des systèmes de fichiers

Les fichiers sont organisés logiquement en files-systèmes, qui correspondent à des zones physiques du périphérique de stockage (disque dur, disquette, cdrom).

D'un point de vue utilisateur, les systèmes de fichier apparaissent sous forme d'arbres dont les branches sont les répertoires(directory) et les feuilles sont les fichiers (files).



3-Le système d'exploitation LINUX

Août 1991, un étudiant finlandais de l'université d'Helsinki, Linus Torvalds, envoie un message sur comp.os.minix :

“Hello everybody out there using minix .I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386 (486) AT clones”.

En effet, cet étudiant, intéressé par le système d'exploitation Minix pour P.C.(un clone d'Unix développé par Andrew Tanenbaum⁵ et ses étudiants) dont les sources étaient disponibles librement, souhaitait adapter ce système à ces besoins et à ses conceptions personnelles et se lançait dans l'écriture de LINUX. Linus Torvalds fut rapidement rejoint dans sa tâche par de nombreux programmeur qui, par le biais d'Internet, apportèrent leur contribution au développement de ce système d'exploitation pour le rendre plus stable , plus ouvert, plus compatible avec les différents matériels, plus User friendly (agréable à utiliser).....

Ainsi apparut, en 1994, la première version stable de Linux(version1.0).

3-Le système d'exploitation LINUX

Linux est donc un système d'exploitation gratuit, fruit de l'action commune de centaines de développeurs dans le monde et qui évolue continuellement grâce aux efforts conjugués de ces bénévoles passionnées (Hackers).

D'un point de vue technique, Linux fonctionne de manière quasi analogue à Unix (puisque'il respecte notamment les normes POSIX).

4- Architecture du système de fichier (notion d'arborescence)

La mémoire de stockage d'un ordinateur est structuré en « répertoires » (en anglais : directory /directories) et « sous répertoires » dans les quels il est possible d'enregistrer des « fichiers » (en anglais : file /files). Le répertoire de plus haut niveau est nommé « répertoire racine » et désigné, sous Unix/Linux par / (slash).

L'ensemble des chemins que l'on peut parcourir à partir d'un répertoire (en allant de sous répertoires en sous-sous-répertoires) est appelé « arborescence ».

L'accès à un répertoire peut être effectué de manière :

-absolue, par une description complète du *chemin d'accès au répertoire depuis le* répertoire racine(/).

Ex : /usr/bin

-relative à la position courante.

Ex : Si la position courante est /usr, le sous répertoire bin est directement accessible par une commande appropriée de la forme.

-relative à l'utilisateur

Ex : ~Raymond/fichiers

4- Architecture du système de fichier (notion d'arborescence)

Nota : A la connexion , l'utilisateur se trouve dans son répertoire « personnel »(en anglais « home directory »).

Le home directory est le répertoire dans le quel un utilisateur a le droit de créer des

Attention : Ne pas confondre le »home directory » qui est le répertoire dans le quel un utilisateur peut écrire ses propres fichiers, et le répertoire « home », situé immédiatement sous la racine, et contenant l'ensemble des « home directory ».

5- Hiérarchie des répertoires sous Linux

La racine /

La racine contient évidemment tout les répertoires que nous allons étudiés.

/bin

Le répertoire /bin contient des exécutables spécifiques au mode utilisateur.

/boot

/boot contient les fichiers nécessaires au démarrage ainsi que des fichiers de configuration des boot loader comme grub par exemple.

/dev

/dev contient des fichiers spéciaux appelés file device qui se réfèrent aux périphériques du système.

Quelques exemples :

le premier lecteur de disquette est appelé /dev/fd0

le second lecteur de disquette est appelé /dev/fd1

le disque maître sur le contrôleur IDE primaire est appelé /dev/hda

le disque esclave sur le contrôleur IDE primaire est appelé /dev/hdb

la première partition du disque maître sur le contrôleur IDE primaire est appelée /dev/hda1

5- Hiérarchie des répertoires sous Linux

/etc

/etc contient des répertoires et des fichiers de configuration des programmes. On y retrouve par exemples les répertoires cups, X11 etc...

L'extension des fichiers de configuration est en général **.conf**

/home

/home c'est le répertoire utilisateurs. Il contient des répertoires propre à chaque utilisateur du système. C'est l'administrateur(**root**) **qui crée ces répertoires.**

/lib

/lib contient les librairies nécessaire à l'amorçage du système et à l'exécution de commandes.

/mnt

/mnt pour mount, ce répertoire contient des répertoires appelés points de montage. Ce sont des répertoires qui pointent vers les fichiers de périphériques, tels que les cdrom, les disques dur externes, ou des systèmes de fichiers comme des partitions ntfs, dos, etc

5- Hiérarchie des répertoires sous Linux

/proc

Le répertoire donne des informations sur le système et le noyau. On y trouve de nombreux fichiers intéressants comme par exemple : configuration du CPU

/root

C'est le répertoire personnel de l'administrateur.

/sbin

Le répertoire /sbin contient des exécutable spécifiques au système.

/sys

Le répertoire /sys permet à l'administrateur de l'ordinateur d'activer et de désactiver des fonctions du noyau. Ce répertoire fournit également des informations sur le système.

/tmp

/tmp contient les fichiers temporaires issus des processus en cours d'exécution.

/usr

/usr est un répertoire important du système Linux. Il contient des données et des programmes communs aux utilisateurs.

5- Hiérarchie des répertoires sous Linux

Le répertoire `/proc` donne des informations sur le système et le noyau. On y trouve de nombreux fichiers intéressants comme par exemple :

`/var`

`/var` contient les journaux d'activités du système appelés logs. Ces fichiers contiennent les dates d'accès de certains programmes, les tâches journalières exécutées par `crontab` y sont également reportées.

Pour plus d'informations vous pouvez utiliser la commande **`man`**

6- LE SHELL

Le « Shell » est ce que l'on appelle un « Interpréteur de commandes ».

Il date de l'époque d'UNIX, où le seul moyen de communiquer avec sa machine était d'écrire des lignes textes au clavier, dans un langage compréhensible à la fois par l'humain et la machine.

Le rôle de la machine étant d'exécuter les commandes de l'utilisateur et d'afficher le résultat à l'écran.

Le shell c'est un programme qui se trouve dans le répertoire **/bin**.

Par définition, il doit être léger et rapide, et reste un service d'accès privilégié aux Noyau Linux (Kernel) pour l'exécution de primitives système.

Depuis, cette « interface » avec la machine a perduré car elle est rapide et fiable pour l'utilisateur qui la maîtrise.

6- LE SHELL

Le shell a évoluée aux travers des années, plusieurs types de Shell existent :

- le /bin/sh **shell Bourne**
- le /bin/bash **shell Bourne Again SHell**
- le /bin/csh **C shell**
- le /bin/ksh **Korn shell**
- le /bin/tcsh **C shell amélioré**

mais le principe de base et toujours resté le même : Les shells sont des interpréteurs, ils lisent chaque commande saisie par l'utilisateur (ou lue à partir d'un fichier), vérifient et traitent la syntaxe pour l'exécuter.

Ici, nous utiliserons le **Shell Bash**, l'un des plus couramment utilisé sur les systèmes **GNU/Linux**. **Bash** est un logiciel libre publié sous **GNU GPL**.

6- LE SHELL

Avantage du shell:

- facilité de mise en oeuvre et installé d'office sous tous système GNU/Linux (pas besoin d'installer un autre langage sur votre système).
- le shell manipule essentiellement des chaînes de caractères : pas de structures complexes, pointeurs, etc...
- le langage est adapté au prototypage rapide d'applications : exécutions d'instructions systèmes fiables, rapides et robustes. Le bash se révèle un outil puissant lorsqu'on le maîtrise.
- c'est un langage « glu » : exécuter et agglomérer des composants divers écrits dans d'autres langages.

Inconvénients du shell:

- messages d'erreurs parfois difficiles à exploiter, ce qui rend la mise au point des scripts fastidieuse.
- Temps d'apprentissage : la syntaxe est cohérente mais ardue. De nombreuses années d'utilisation sont nécessaires pour être « à l'aise » avec la ligne de commande.

6- LE SHELL

ATTENTION :

Le shell permet avant tout d'exécuter des commandes, d'explorer l'arborescence du système, de créer, d'éditer et de supprimer des fichiers, etc.

Au travers du Shell, vous touchez donc à des parties très importantes et sensibles de votre système d'exploitation, et exécuter des commandes que l'on ne comprend pas peut vraiment avoir des conséquences « catastrophiques » sur ce dernier ! raison de plus pour faire quelques rappels et d'être à l'aise avec ces commandes.

La recommandation principale dans l'utilisation du shell est la « **prudence** » ! **executer des commandes en « root »** ne doit s'effectuer que pour des tâches bien spécifiques d'administration que l'utilisateur maîtrise parfaitement.

Les commandes effectuées dans ce cours peuvent s'effectuer avec un simple compte utilisateur, pas besoin d'être en « root » !

6- LE SHELL

ATTENTION :

Le shell permet avant tout d'exécuter des commandes, d'explorer l'arborescence du système, de créer, d'éditer et de supprimer des fichiers, etc.

Au travers du Shell, vous touchez donc à des parties très importantes et sensibles de votre système d'exploitation, et exécuter des commandes que l'on ne comprend pas peut vraiment avoir des conséquences « catastrophiques » sur ce dernier ! raison de plus pour faire quelques rappels et d'être à l'aise avec ces commandes.

La recommandation principale dans l'utilisation du shell est la « **prudence** » ! **executer des commandes en « root »** ne doit s'effectuer que pour des tâches bien spécifiques d'administration que l'utilisateur maîtrise parfaitement.

Les commandes effectuées dans ce cours peuvent s'effectuer avec un simple compte utilisateur, pas besoin d'être en « root » !

6- LE SHELL

Lancer un Shell (mode console ou terminal):

Pour accéder à la ligne de commande, il est possible d'utiliser un terminal (xterm, kterm, gterm) ou encore une console virtuelle.

Pour ce faire, plusieurs possibilités sont offertes :

- utiliser le menu du bureau (Gnome, Kde, etc.). Il s'agit de la méthode conseillée.
- utiliser le menu *lancer une application*. Dans la fenêtre ainsi ouverte, taper le nom de terminal et valider. La fenêtre **lancer une application peut être ouverte avec Alt+F2**
- Utiliser les consoles virtuelles (il y en a 6).

6- LE SHELL

La console virtuelle :

La console virtuelle (**tty1 à tty6 « Putty »**) est un écran noir où une invite de commande apparaît, de la forme **login** :

Depuis l'interface graphique, il est possible de se connecter à une console virtuelle en utilisant la combinaison de touches **Ctrl+Alt+FN**, où **N** est un chiffre de **1 à 6**.

Pour revenir au mode graphique depuis une console virtuelle, utiliser la combinaison de touches **ALT+F7**.

6- LE SHELL

Les utilisateurs

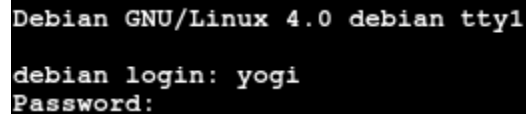
Linux a été pensé dès l'origine comme étant sécurisé, multi-tâche et donc multi-utilisateurs (gestion de droits multiples).

lorsqu'un utilisateur veut utiliser le système sous GNU/Linux, la première étape est tout d'abord de s'authentifier au près du système, se dernier lui attribuant les droits, des espaces propres qui lui ont été prédéfinis et attribués lors de la configuration du système : administrateur, utilisateurs, invité

6- LE SHELL

Connexion dans une console virtuelle

A l'invite **login:**, saisir l'identifiant (login) de l'utilisateur, puis valider en appuyant sur Entrée, l'invite **Password:** s'affichera. Saisir le mot de passe de l'utilisateur et valider.

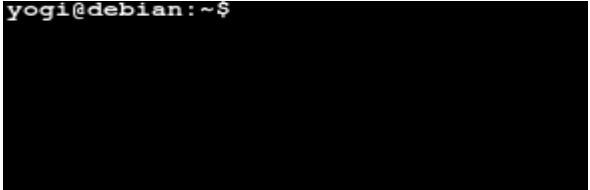


```
Debian GNU/Linux 4.0 debian tty1
debian login: yogi
Password:
```

- 1) le mot de passe n'est pas affiché à l'écran pour des raisons de sécurité. Le fait de ne pas voir à l'écran ce que vous tapez, ne doit pas vous inquiéter.
- 2) Si le mot de passe est valide, une invite de commande comme suite doit s'afficher, indiquant que la connexion a réussie.

6- LE SHELL

3) L'invite de commande du shell après la connexion



```
yogi@debian:~$
```

Suite à la connexion, l'invite de commande (ou prompt) du shell apparaît et a en général la forme suivante :

utilisateur@machine ~ \$

- **utilisateur** - représente l'identifiant ou le nom de l'utilisateur connecté
- **machine-** représente le nom de la machine sur laquelle l'utilisateur est connecté
- **~ est** un raccourci qui signifie le répertoire personnel /home/utilisateur
- **\$** signifie que vous êtes connecté en tant qu'utilisateur

utilisateur@machine ~ #

Si au lieu de \$ le signe # apparaît, alors vous êtes connecté en tant que « superutilisateur » (**root**).

Gardez à l'esprit que les systèmes GNU/Linux utilisent par convention # pour **root** et \$ pour un **utilisateur autre que root**.

6- LE SHELL

ATTENTION : « root » à tous les droits, celui aussi de casser votre système lors de l'exécution d'une commande dangereuse... pensez donc à rester le maximum avec vos droits utilisateur, la simple commande exit devrait vous permettre de vous reloguer sous votre utilisateur.

Changer d'identité

Depuis votre shell vous avez la possibilité de prendre l'identité d'un autre utilisateur existant sur votre système, y compris l'utilisateur "**root**" (**utilisateur qui dispose de tous les privilèges**).

Pour faire cela vous avez à votre disposition la commande **su** ou **su -**

6- LE SHELL

L'utilisation des commandes

Le principe d'exécution des commandes repose sur un principe de fonctionnement simple.

L'exécution est séquentielle, les commandes sont exécutées les unes à la suite des autres.

Les commandes peuvent se suivre sur la même ligne mais il faut les séparer par des ;

L'exécution de la commande s'effectue pas l'appuie de la touche « entrée ».

Exemples:

\$ commande1 ; commande2

La suite de commande : % commande1 ; commande2 ; commande3

correspond à la séquence :

\$ commande1

\$ commande2

\$ commande3

6- LE SHELL

Exécution en tâche de fond :

\$ commande &

Exécution asynchrone :

\$ commande1 & commande2

Nous étudierons plus en détails l'utilisation de caractère & dans la partie *Pipeline*
et

parallélisme du cours

Quelques commandes de base

Le catalogue de quelques commandes de bases est déjà assez fourni comme vous allez pouvoir le constater.

L'intérêt étant d'en connaître le maximum sinon la commande man sera toujours là pour vous aider!

Toutes ces commandes sont « sensibles à la case » (majuscule/minuscule), le caractère d'espacement servant de séparateur.

6- LE SHELL

Quelques commandes de base

cat : Lit (concatène) un ou plusieurs fichier(s), affichage sur la sortie standard

cd : ChangeDirectory, change de répertoire. 'cd' seul permet de revenir dans le home directory

chmod : CHangeMODE - change le mode d'accès (permissions d'accès) d'un ou plusieurs fichier(s)

chown : CHangeOWNeR - change le propriétaire d'un ou de plusieurs fichier(s)

cp : copier des fichiers

crontab : planification de tâches

cut : Retire des parties précises de texte dans chaque ligne d'un fichier

date : Affiche la date selon le format demandé

dd : DevicetoDevice - Recopie octet par octet tout ou partie du contenu d'un périphérique (habituellement de stockage) vers un autre périphérique.

df : affichage de la quantité d'espace libre disponible sur tous les systèmes de fichiers

du : DiksUsage - l'utilisation de disque

echo : Affiche du texte sur la sortie standard (à l'écran)

exit : arrête l'exécution du shell

find : recherche de fichiers

fsck : FileSystemCheck - vérification d'intégralité de système de fichiers

grep : recherche dans un ou plusieurs fichiers les lignes qui correspondent à un motif

6- LE SHELL

Quelques commandes de base

groupadd : Ajouter un groupe d'utilisateurs

gunzip : décompression de fichiers

gzip : compression de fichiers

head : affiche les premières lignes (par défaut 10) d'un fichier

help : affiche une aide sur les commandes internes de bash

history : affiche l'historique des commandes déjà utilisées

kill : envoyer un signal à un processus

less : programme d'affichage à l'écran

ln : création de liens

ls : liste le contenu des répertoires

man : Une des plus importantes! affiche les pages de manuel de la commande qui suit.

mkdir : MaKeDIRectory - crée un répertoire

mkfs : MaKeFileSystem - création de systèmes de fichiers

more : programme d'affichage à l'écran

mount : monter un système de fichiers

6- LE SHELL

Quelques commandes de base

ps : affiche les processus en cours d'exécution

pwd : Print name of current/working directory - affiche le chemin complet du repertoire courant

rm : suppression de fichiers

rmdir : Remove empty directories - suppression d'un dossier vide

tail : affiche les 10 dernières lignes d'un fichier

tar : création d'archives

su : Substitute User identity ou Switch User - prendre l'identité d'un utilisateur

uname : Affiche des informations sur le système.

useradd : ajouter un utilisateur

whereis : *localiser une commande (équivalent à which).*

Les «

6- LE SHELL

Quelques commandes de base

Les « man » pages

Comme on l'a vu, le nombre de commandes est assez importante, et encore, nous n'en avons vu qu'une toute petite partie ! Accéder à la documentation en ligne de commande se révèle donc « Essentielle » à tout utilisateur du shell, même à l'expert. La syntaxe pour appeler la documentation d'une commande est simple :

man commande

man n commande

n - le numéro de la page man (vous le verrez en haut à gauche)