Administration de système UNIX et Réseau

Time Université 2014 -2015

Préparation à la certification UNIX

PLAN

Chapitre 1
- UNIX INTRODUCTION

Chapitre 2
- ARCHITECTURE ET COMMANDE

Chapitre 3
- GESTION D'ACCES UTILISATEURS

Chapitre 4
- GESTION DISQUES ET SYSTEMES DE FICHIER

Chapitre 5
- SAUVEGARDE ET RESTAURATION

Chapitre 6
- GESTION DES PACKAGES ET MISE A JOUR

Chapitre 7
- RESEAUX

Chapitre 8
- SECURITE SYSTÈME

Chapitre 1

UNIX : INTRODUCTION
GÉNÉRALITÉS, ADMINISTRATION

Un système d'exploitation est une application comme une autre.

Exemple, version kernel d'Unix Solaris:

- SunOS hostname 5.10 Generic_144488-17 sun4u sparc SUNW,SPARC-Enterprise

Le rôle d'un système d'exploitation est :

- La mise à disposition de ressources matérielles: espace disque, temps d'exécution sur le microprocesseur central, espace mémoire, etc.
- Le partage équitable de ces ressources entre les utilisateurs pour atteindre le but de système multi-utilisateurs.

UNIX est un système multi-tâches.

UNIX permet une exécution **concurrente** et **indépendante** de plusieurs applications initiées par plusieurs utilisateurs.

Concurrence : les applications demandent l'accès à des ressources partagées (processeur, mémoire, disques durs, etc.).

Unix implémente un ordonnanceur pour gérer et arbitrer les demandes :

- C'est un chef d'orchestre

Indépendance : Les applications s'exécutent de manière autonome et indépendante entre eux.

UNIX est un système multi-utilisateurs.

Chaque programme exécuté sous Unix est étroitement lié à un utilisateur:

- Authentification de l'utilisateur
- Protection de l'exécution de chaque programme utilisateur
- Comptabilité des ressources allouées à l'utilisateur : Privilège sur les répertoires

Un utilisateur correspond à :

- Un identifiant
- Un mot de passe

Les versions d'Unix sont interchangeables.

Solaris pour les machines ORACLE SUN

- Solaris 10 , Open Solaris 10
- http://www.oracle.com

MacOS X pour les machines APPLE

- OS X 10.x
- http://www.apple.com

AIX pour les machines IBM

- AIX 7.X
- http://www.ibm.com

Linux

- noyau 2.6.x
- http://www.kernel.org

Constructeur Hardware

Non constructeur

- **Du point de vue de l'utilisateur, les divers UNIX se ressemblent beaucoup.**

- **Du point de vue de l'administration, les divers UNIX ont chacun des spécificités**

  – Les commandes liées au hardware varient, on trouve des extensions propres à chaque constructeur.

→ En pratique, l'administrateur attend toujours plusieurs tentatives d'unification

  – *System V Interface Definition* de AT&T (SVID, SVID2, SVID3 *en 1989)*
  – *IEEE POSIX (POSIX1003.1 en 1990)*
  – *X/OPEN Portability Guide (XPG4 en 1993) du consortium X/OPEN (créé en 1984)*

Mais. . .

Il reste 2 grandes familles d'UNIX

- la famille **System V** avec notamment la dernière version connue sous le nom de **System V release 4** (alias *SVR4*)
- la famille **BSD** issue de l'université de Berkeley (BSD *Berkeley Software Distribution*)

LINUX ?

Ken Thompson et Dennis M. Ritchie, les parents d'UNIX
On notera les teletypes 33 !

LINUX n'est pas un système UNIX au sens propre car son code ne provient pas du code originel des UNIX.

On parle de « UNIX like » pour LINUX et plus généralement de « **système GNU/LINUX** ».

En 1991, LINUX fait son apparition comme système d'exploitation pour la communauté Open Source par Linus Torvald

La première version de Linux peut faire fonctionner que quelques applications tels que:

- bash, gcc, sed, etc.

Aujourd'hui :

LINUX est devenu une plateforme de référence pour les applications d'entreprises

Raison du succès :

- Code source sous licence GPL (GNU Public Licence == cette licence empêche le logiciel libre d'être modifié et de devenir propriétaire)
- Linux appartient à une Communauté mondiale de développeurs

→ Il existe beaucoup de distributions LINUX: Redhat, Suse, …

Avantages des distributions UNIX/LINUX

- Faible consommation en mémoire vive
- Faible consommation en espace disque
- Grande stabilité
- Grande rapidité en multi-tâches, même sur un système monoprocesseur
- Evolution très rapide
- Système totalement paramétrable
- Système très complet, utilisable dès la fin de l'installation
- Parfaitement adapté au mode multi-utilisateur

Rôle de l'Administrateur

- Gérer les comptes utilisateurs (tâche simple et automatisable)
- Assister et éduquer les utilisateurs (réponses à leurs questions, documentation à jour)
- Gérer les logiciels (installation, configuration, mise à jour)
- Gérer le matériel (panne, remplacement, ajout)
- Assurer la sécurité du système et des utilisateurs (sauvegardes fiables et régulières, contrôle d'accès, utilisations abusives de ressources)
- Vérification de l'adéquation du matériel avec son utilisation (identifier les goulets d'étranglement)
- Maintenance de premier niveau (diagnostiquer une panne, appel de la maintenance constructeur)
- Gestion quotidienne (multiples tâches, petites ou grosses)

- L'administrateur est le première intervenant lorsqu'un problème surgit. C'est lui qu'on incrimine naturellement lorsque quelque chose ne marche pas.

Charge de travail :

  - dépend de la taille du site : Nombre des machines

Attention:

- Soyez sûr de pouvoir revenir en arrière : sauvegarder tout fichier qui doit être modifié :
  - **% mv config.ini config.ini.orig**
  - **% cp config.ini.orig config.ini**
  - **% vi config.ini**

- Documentez ce que vous faites.
  - Les lignes commentaires doivent commencer par le caractère **#**

Exemple :

```
# echo 'Bonjour ….'
```

- Faite attention lors d'exécution de la commande Shell rm

rm /tmp *	**rm /tmp/***

- Utiliser la documentation en ligne « man » pour connaitre les différents options de la commande
  - **Exemple : man rm**

Administrateur d'UNIX :

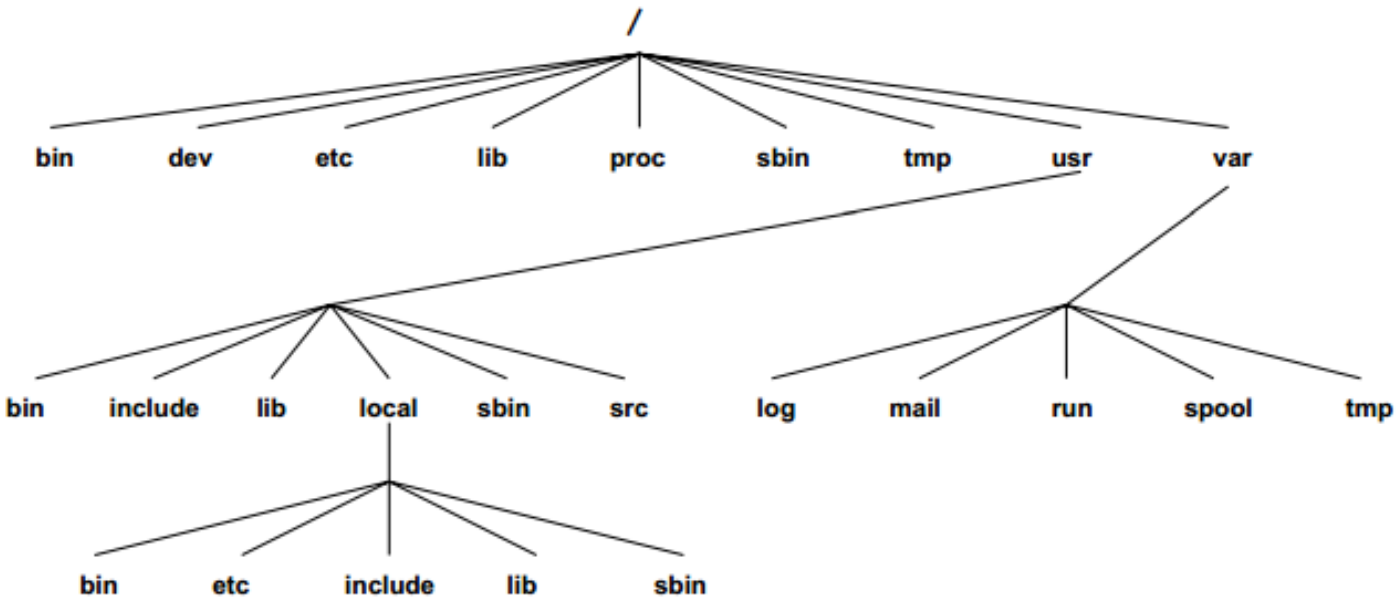
- C'est un utilisateur expert d'UNIX qu 'exploite :
  - L' environnement utilisateur
  - L' aide en ligne « man »
  - Le système de fichiers
  - Le shell
  - L' éditeur de texte
  - Les commandes de base
  - Les fonctionnalités fournie via la programmation shell pour automatiser les tâches d'administration

Chapitre 2

UNIX : ARCHITECTURE ET COMMANDE

Arborescence

Le système d'exploitation Unix est basé sur une structure de type arbre



Répertoire	Description
/	La racine de l'arborescence UNIX.
/bin	Lien symbolique vers le répertoire /usr/bin. Répertoire qui contient les fichiers des commandes standards du système.
/dev	Répertoire qui contient les noms logiques des périphériques. Ce répertoire contient des liens symboliques qui pointent vers les fichiers des noms physiques des périphériques (répertoire /devices).
/etc	Ce répertoire contient les fichiers de configuration de la machine et les bases de données de l'administration système.
/export	Répertoire utilisé, par défaut, pour le partage de systèmes de fichiers tels que les répertoires home des utilisateurs, les logiciels, ou d'autres systèmes de fichiers partagés.

Répertoire /sbin

/sbin = system binaries

Il contient les binaires système primordiaux utilisés pendant le boot de la machine.

(historiquement, ces binaires étaient dans /etc)

Il réside dans la même partition que / ce qui garantit sa disponibilité à ce moment-là où d'autres partitions ne sont pas encore disponibles.

Exemple non exhaustif de commandes :

- fsck, halt, init, shutdown …

Répertoire /bin

/bin = binaries

Il contient les binaires système primordiaux utilisés pendant le boot de la machine.

Il réside dans la même partition que / ce qui garantit sa disponibilité à ce moment-là où d'autres partitions ne sont pas encore disponibles.

Exemple non exhaustif de commandes :

- bash, chmod, chown, cp, date, dd, df, echo, hostname, kill, mkdir, more, mv, ps, pwd, rm, rmdir, sed, umount, etc

Répertoire /usr/sbin

/usr/sbin = user level system binaries

Il contient des binaires système (d'où le 's' dans son nom) non primordiaux et plus utilisés pour la gestion au jour le jour de la machine.

Il réside dans une même partition pouvant ne pas être disponible pendant les premières étapes du boot.

Exemple non exhaustif de commandes :

- adduser, addgroup…

Répertoire /usr/bin

/usr/sbin = user level binaries

Il contient la plupart des commandes Unix pour utilisateurs (plusieurs centaines en général).

Exemple: date, grep, ln, perl, sed, vi, who, etc.

Répertoire /usr/local/sbin

/usr/local/sbin = local user level system binaries

Il contient les commandes Unix d'administration propres au site ou propres à la machine.

Répertoire /usr/local/bin

/usr/local/bin = local user level binaries

Il contient la plupart des commandes Unix utilisateurs installées localement.

Répertoire /lib

/lib = libraries

Il contient en général les librairies dynamiques primordiales pour le fonctionnement du système.

ATTENTION : ne pas effacer les librairies dynamiques de ce répertoire

Exemple non exhaustif de fichiers : ld.so, libc.so.5, libm.so.5, etc

Répertoire /usr/lib

/usr/lib = user level binaries

Il contient les librairies dynamiques non primordiales au fonctionnement du système ainsi que les librairies statiques '.a' de programmation en langage C.

Exemple non exhaustif de fichiers : libm.a, libm.so.

Répertoire /usr/local/lib

/usr/local/lib = local user level libraries

Il contient les librairies dynamiques ou statiques installées localement.

Répertoire /usr/include

/usr/include = user level include files

Il contient les fichiers '.h' de programmation en langage C

Répertoire /usr/local/include

/usr/local/include = local user level include files

Il contient les fichiers '.h' installés localement

Répertoire /etc

/etc = system level config files and etc

Il contient les fichiers de configuration des divers utilitaires primordiaux assurant le fonctionnement du système.

Exemple : group, passwd, hosts, vfstab, resolv.conf, syslog.conf, etc.

Répertoire /dev

/dev = system devices

l contient les fichiers spéciaux du système permettant d'accèder aux ressources physiques de la machine comme les disques durs, la mémoire RAM etc.

les fichiers spéciaux sont générés automatiquement (faire en plus « boot –r » sur SOLARIS)

Pas de /usr/dev ni /usr/local/dev

Les droits d'accès aux fichiers spéciaux dans /dev sont très importants.

Une précaution : conserver une trace écrite d'un "ls -l" de /dev

Arborescence

- Organisation structurée (SOLARIS, HP-UX, etc.) en sous répertoires

Exemple Solaris :

Répertoire	Contenu
/dev/rdsk	disques en mode caractère
/dev/dsk	disques en mode bloc
/dev/rmt	bandes magnétiques

Fichier	Contenu
/dev/audio	le device de la carte audio de la machine
/dev/console	le device de la console texte de la machine
/dev/null	trou noir

Répertoire /dev/null

/dev/null = Trou noir pour redirection

On peut vouloir se débarrasser d'une partie de l'affichage.

Solution inefficace :

```
% application > /tmp/toto
...
% rm /tmp/toto
```

La solution est de rediriger vers /dev/null :

```
% application > /dev/null
...
```

C'est un fichier spécial (logique car dans /dev…) :

crw-rw-rw- 1 root sys 13, 2 Oct 19 01:13 /dev/null

/dev/null est indispensable dans la vie de l'administrateur système

Répertoire /proc

/proc = system process

Il contient une représentation sous forme de fichiers virtuels et directories virtuels d'informations de la machine Unix

Représentation virtuelle de :

- informations sur les processus
- informations sur le hardware
- informations sur le noyau

```
% echo $$
23247
% cd /proc/23247
% ls -l
total 0
-r--r--r--   1 besancon sysadmin    0 Sep 22 14:30 cmdline
lrwx------   1 besancon sysadmin   64 Sep 22 15:45 cwd -> [0001]:1523515394
-r--------   1 besancon sysadmin    0 Sep 22 15:45 environ
lrwx------   1 besancon sysadmin   64 Sep 22 15:45 exe -> [0801]:10251
dr-x------   1 besancon sysadmin    0 Sep 22 15:45 fd
pr--r--r--   1 besancon sysadmin    0 Sep 22 15:45 maps
-rw-------   1 besancon sysadmin    0 Sep 22 15:45 mem
lrwx------   1 besancon sysadmin   64 Sep 22 15:45 root -> [0801]:2
-r--r--r--   1 besancon sysadmin    0 Sep 22 14:29 stat
-r--r--r--   1 besancon sysadmin    0 Sep 22 15:45 statm
-r--r--r--   1 besancon sysadmin    0 Sep 22 15:45 status
```

Répertoire /tmp

/tmp = Temporary

Le répertoire /tmp sert à stocker des fichiers temporaires

Ses droits d'accès :

drwxrwxrwt 12 root sys 2648 Sep 28 13:02 /tmp/

c'est-à-dire 1777 exprimé en octal :

- signification de 777 : tout le monde sur la machine peut créer, modifier, effacer des fichiers
- signification de 1000 : un utilisateur ne peut effacer que les fichiers qui lui appartiennent

En principe, /tmp est vidé au boot de la machine Unix.

Répertoire /var/log

/var/log= ***log files***

Il stocke différents fichiers de bord du système

Exemple:

```
total 1900
-rw-------   1 root     sys            0 Oct 29  2002 authlog
-rw-r--r--   1 root     other     294141 Nov  2 18:50 ppp.log
-rw-r--r--   1 root     root         844 Aug 28 13:06 skipd.log
-rw-r--r--   1 root     root         844 Aug 28 12:55 skipd.log.0
-rw-r--r--   1 root     root         844 Aug 28 12:52 skipd.log.1
-rw-r--r--   1 root     root         844 Aug 28 12:45 skipd.log.2
-rw-r--r--   1 root     root         844 Aug 28 12:41 skipd.log.3
-rw-r--r--   1 root     other        254 Oct 29  2002 sysidconfig.log
-rw-r--r--   1 root     sys        57297 Nov  2 18:50 syslog
-rw-r--r--   1 root     sys        74519 Oct 31 03:10 syslog.0
-rw-r--r--   1 root     sys        85749 Oct 24 03:10 syslog.1
-rw-r--r--   1 root     sys        79963 Oct 17 03:10 syslog.2
-rw-r--r--   1 root     sys        69391 Oct 10 03:10 syslog.3
-rw-r--r--   1 root     sys        85748 Oct  3 03:10 syslog.4
-rw-r--r--   1 root     sys        69120 Sep 26 03:10 syslog.5
-rw-r--r--   1 root     sys        74101 Sep 19 03:10 syslog.6
-rw-r--r--   1 root     sys        63515 Sep 12 03:10 syslog.7
```

Répertoire /var/log

/var/log= ***variable log files***

Il stocke différents fichiers de bord du système

Exemple:

```
total 1900
-rw-------   1 root     sys            0 Oct 29  2002 authlog
-rw-r--r--   1 root     other     294141 Nov  2 18:50 ppp.log
-rw-r--r--   1 root     root         844 Aug 28 13:06 skipd.log
-rw-r--r--   1 root     root         844 Aug 28 12:55 skipd.log.0
-rw-r--r--   1 root     root         844 Aug 28 12:52 skipd.log.1
-rw-r--r--   1 root     root         844 Aug 28 12:45 skipd.log.2
-rw-r--r--   1 root     root         844 Aug 28 12:41 skipd.log.3
-rw-r--r--   1 root     other        254 Oct 29  2002 sysidconfig.log
-rw-r--r--   1 root     sys        57297 Nov  2 18:50 syslog
-rw-r--r--   1 root     sys        74519 Oct 31 03:10 syslog.0
-rw-r--r--   1 root     sys        85749 Oct 24 03:10 syslog.1
-rw-r--r--   1 root     sys        79963 Oct 17 03:10 syslog.2
-rw-r--r--   1 root     sys        69391 Oct 10 03:10 syslog.3
-rw-r--r--   1 root     sys        85748 Oct  3 03:10 syslog.4
-rw-r--r--   1 root     sys        69120 Sep 26 03:10 syslog.5
-rw-r--r--   1 root     sys        74101 Sep 19 03:10 syslog.6
-rw-r--r--   1 root     sys        63515 Sep 12 03:10 syslog.7
```

Répertoire /var/mail

/var/mail= variable mailbox files

Il stocke les boites aux lettres de courrier électronique.

total 2

-rw-rw---- 1 root mail 0 Dec 4 2002 root

Parfois, on rencontre encore /var/spool/mail

Répertoire /var/run

/var/run= variable run time data

Il contient des informations sur l'état du système après le dernier boot.

Exemple : pid

→Nettoyer au moment du boot les anciens fichier

Répertoire /var/spool

/var/spool= variable spool data

ll stocke des sous répertoires dédiés à des applications consommatrices de nombreux fichiers temporaires comme par exemple le système d'impression ou la messagerie électronique.

- Pour le système d'impression, c'est « /var/spool/lpd ».
- Pour la messagerie électronique, c'est « /var/spool/mqueue ».

Répertoire /var/tmp

/var/tmp= variable temporary

Le répertoire /var/tmp sert à stocker des fichiers temporaires

Différence avec /tmp : /var/tmp n'est pas vidé au boot de la machine Unix

Répertoire /mnt

/mnt= mount point

Point de montage pour rendre disponible provisoirement une partition (locale ou réseau).

/mnt est un répertoire.

Répertoire /opt

/usr/local **local add-ons**

contient les ajouts locaux de commandes.

/opt **local add-on applications software**

contient les ajouts locaux de « paquets » complets de commandes (càd nombreuses commandes exécutables et fichiers annexes)

Syntaxe:

- Une commande UNIX = un ensemble de mots séparés par des caractères blancs (caractère espace, tabulation)

Le premier mot : le nom de la commande

Le reste des mots : les paramètres de la commande

Particularités de certains mots : des options qui changent le comportement de la commande

En pratique on trouvera donc écrit :

commande [options] paramètres

- Les 2 crochets « [» et «] » indiquent que les options ne sont pas obligatoires.
- Il ne faut pas taper ces crochets sur la ligne de commande.
- ***Toujours consulter la documentation pour avoir plus de détail sur la manière d'utiliser les différents options***
  - ***Example : man ls***

Composant d'un fichier:

- Tous les fichiers de l'environnement d'exploitation Solaris se servent d'un nom de fichier et d'un enregistrement nommé un **inode** pour accèder aux blocs de données.
- Un nom de fichier est associé à un ***inode*** (noeud d'index)
- L'inode fournit l'accès aux blocs de données.

Noms des fichiers

- Les noms des fichiers sont les objets les plus régulièrement utilisés pour accéder et manipuler les fichiers.
- Un fichier doit posséder un nom associé à un inode.

Inodes

- L'environnement d'exploitation Solaris se sert des inodes pour enregistrer des informations sur les fichiers.

Les inodes contiennent généralement deux parties :

- La première partie contient des informations sur le fichier, telles que le propriétaire, les permissions d'accès et la taille.
- La seconde partie de l'inode contient des pointeurs vers les blocs de données associés au fichier.

Blocs de données

- Les blocs de données sont des unités d'espace disque qui permettent le stockage de données.

- Les fichiers ordinaires, les répertoires et les liens symboliques utilisent des blocs de données.

- Les fichiers spéciaux ne contiennent pas de données.

Il existe différents types de « fichiers », les plus courants sont :

- **Normal :** représenté par **« - »**
- **Répertoire :** représenté par **« d »**
- **Lien symbolique :** représenté par **« l »**
- **Périphérique** : représenté par **« c ou b » → fichiers spéciaux**

- Commande pour voir le détail des fichiers : **ls -l**

```
# cd /etc
# ls -l
total 428
drwxr-xr-x 2 adm adm 512 Apr 3 10:42 acct
lrwxrwxrwx 1 root root 14 Apr 3 11:05 aliases ->
./mail/aliases
drwxr-xr-x 2 root bin 512 Apr 3 10:45 apache
-rw-r--r-- 1 root bin 50 Apr 3 10:45 auto_home
-rw-r--r-- 1 root bin 113 Apr 3 10:45 auto_master
(résultat tronqué)
```

```
# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l
total 0
brw------- 1 root sys 136, 0 Apr 3 11:11 dad@0,0:a
crw------- 1 root sys 136, 0 Apr 3 11:11 dad@0,0:a,raw
brw------- 1 root sys 136, 1 Apr 4 11:06 dad@0,0:b
crw------- 1 root sys 136, 1 Apr 3 11:11 dad@0,0:b,raw
(résultat tronqué)
```

- Les *Unix* définissent les droits d'accès sur un objet du *filesystem* relativement à trois droits fondamentaux que *GNU/Linux* a repris (et que les *xBSD* suivent aussi). Leur notation symbolique est :
  - **r** : lecture [*read*];
  - **w** : écriture [*write*];
  - **x** : exécution [*execution*].

Commandes de création des fichiers standards: touch, vi

Répertoires

- Les répertoires stockent les informations qui associent les noms des fichiers aux numéros d'inodes.

- Contrairement aux fichiers, qui peuvent contenir toutes sortes de types de données, les répertoires ne peuvent contenir que des associations entre un nom de fichier et un numéro d'inode.

- Un répertoire contiennent la liste des noms de fichiers et leurs numéros d'inodes associés

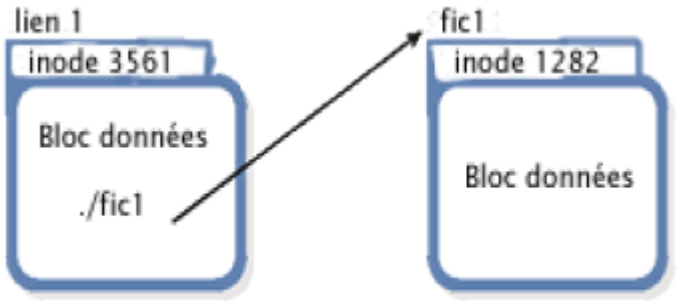
Liens symboliques

Un lien symbolique contient le chemin d'accès au fichier vers lequel il pointe. Les liens symboliques peuvent pointer vers des fichiers placés dans d'autres systèmes de fichiers parce qu'ils précisent les chemins d'accès.

- La taille d'un lien symbolique correspond toujours au nombre de caractères contenus dans le chemin d'accès.

```
# cd /
# ls -l bin
total 135
lrwxrwxrwx 1 root root 9 Mar 22 10:39 bin -> ./usr/bin
```

```
# ln -s fic1 lien1
```



Méthodes de création

`ln -s chemin cible`

Fichiers spéciaux

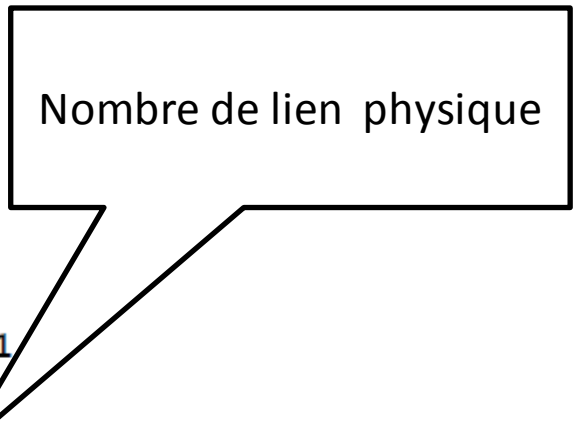
Un fichier spécial donne accès à un périphérique. Contrairement aux fichiers ordinaires, aux répertoires et aux liens symboliques, les fichiers spéciaux n'utilisent pas de blocs de données. Les informations des inodes des fichiers spéciaux contiennent des numéros qui référencent les périphériques

- Numéro majeur :
  - identifie le pilote de périphérique nécessaire pour accéder au
- Numéro mineur :
- identifie l'unité particulière que le pilote de périphérique contrôle.

Lien physique

Un lien physique est une association entre un nom de fichier et un inode.

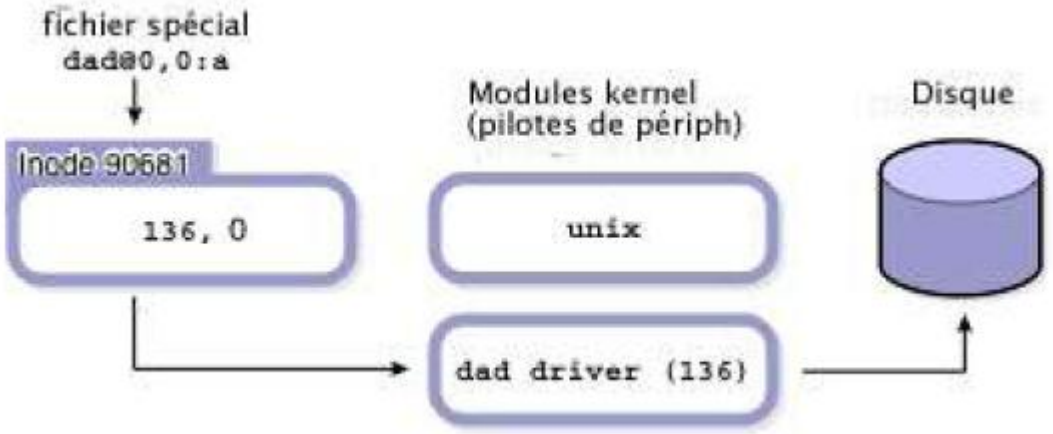
- Chaque type de fichier utilise au moins un lien physique.
- Chaque entrée d'un répertoire constitue un lien physique.




```
# cd rep1
# touch fic1
# ls -l
total 0
-rw-r--r-- 1 root other 0 Apr 7 15:26 fic1
```

```
# ln fic1 fic2
# ls -l
total 0
-rw-r--r-- 2 root other 0 Apr 7 15:26 fic1
-rw-r--r-- 2 root other 0 Apr 7 15:26 fic2
# ls -li
total 0
1282 -rw-r--r-- 2 root other 0 Apr 7 15:26 fic1
1282 -rw-r--r-- 2 root other 0 Apr 7 15:26 fic2
# find . -inum 1282
./fic1
./fic2
```

```
# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l dad@0*
total 0
brw-r----- 1 root sys 136, 0 Sep 6 15:40 dad@0,0:a
crw-r----- 1 root sys 136, 0 Sep 6 15:40 dad@0,0:a,raw
brw-r----- 1 root sys 136, 1 Sep 7 07:37 dad@0,0:b
```



Les informations d'inode de dad@0,0:a contiennent le numéro majeur 136 et le numéro mineur 0. Le numéro majeur 136 identifie le pilote de périphérique dad. Le pilote de périphériques dad contrôle les disques IDE (Integrated Device Electronics). Le numéro mineur 0, dans ce cas, identifie la partition 0.

```
# ls -l

# ls -a | more
```

Lien symbolique

Méthodes de création

`ln -s chemin cible`

Lien physique

```
# ln fic1 fic2

# find . -inum 1282

# rm fic1
# ls -li
total 0
1282 -rw-r--r-- 1 root other 0 Apr 7 15:26 fic2
```

Exercice : Identifier les types des fichiers (niveau 1)

Dans cet exercice, vous effectuez les tâches suivantes :

- naviguer dans l'arborescence de répertoires,
- identifier les différents types de fichiers.

Tâches à effectuer

Effectuez les tâches suivantes :

- Identifier le premier lien symbolique listé dans le répertoire / (racine). Notez la taille du lien symbolique et le nom du fichier qu'il référence. Identifiez les types de fichiers contenus dans le répertoire `/dev/dsk` et les types des fichiers référencés par les liens symboliques, s'il y en a. Identifiez les types de fichiers contenus dans le répertoire `/dev/pts` et les types des fichiers référencés par les liens symboliques, s'il y en a.

- Identifiez les types des fichiers contenus dans le répertoire `/etc/init.d`. Notez le numéro d'inode et le nombre de liens du fichier `volmgt`. Servez-vous de la commande `find` pour localiser tous les autres fichiers sous le répertoire `/etc` qui utilisent le même numéro d'inode.

- Créez un répertoire nommé `/testdir`. Dans ce répertoire, créez un fichier et un lien symbolique qui pointe vers ce fichier. Déterminez si les deux fichiers utilisent ou non le même numéro d'inode.

  Créez un répertoire nommé `newdir` dans le répertoire `/testdir`. Identifiez son numéro d'inode, le nombre de liens et le nom des autres fichiers qui utilisent le même numéro d'inode que le répertoire `newdir`.

  Créez un autre répertoire sous le répertoire `newdir`. Déterminez la modification du nombre de liens du répertoire `newdir` et trouvez tous les nouveaux fichiers qui utilisent le même numéro d'inode que le répertoire `newdir`.

Exercice : Identifier les types des fichiers (niveau 2)

Dans cet exercice, vous effectuez les tâches suivantes :

- naviguer dans l'arborescence de répertoires,
- identifier les différents types de fichiers.

Tâches à effectuer

Effectuez les tâches suivantes :

- Identifier le premier lien symbolique listé dans le répertoire / (racine). Notez la taille du lien symbolique et le nom du fichier qu'il référence. Identifiez les types de fichiers contenus dans le répertoire `/dev/dsk` et les types des fichiers référencés par les liens symboliques, s'il y en a.

  Identifiez les types de fichiers contenus dans le répertoire `/dev/pts` et les types des fichiers référencés par les liens symboliques, s'il y en a.
- Identifiez les types des fichiers contenus dans le répertoire `/etc/init.d`. Notez le numéro d'inode et le nombre de liens du fichier `volmgt`. Servez-vous de la commande find pour localiser tous les autres fichiers sous le répertoire `/etc` qui utilisent le même numéro d'inode que `volmgt`.
- Créez un répertoire nommé `/testdir`. Dans ce répertoire, créez un fichier et un lien symbolique qui pointe vers ce fichier. Déterminez si les deux fichiers utilisent ou non le même numéro d'inode.
- Créez un répertoire nommé newdirdans le répertoire `/testdir`. Identifiez son numéro d'inode, le nombre de liens et le nom des autres fichiers qui utilisent le même numéro d'inode que le répertoire `newdir`.

- Créez un autre répertoire sous le répertoire `newdir`. Déterminez la modification du nombre de liens du répertoire `newdir` et trouvez tous les nouveaux fichiers qui utilisent le même numéro d'inode que le répertoire `newdir`.

- Tout fichier du système appartient à la fois à un utilisateur (son propriétaire") et à un groupe.
- Pour chaque fichier l'ensemble des utilisateurs potentiels est partitionné selon trois classes nommées :
  - **u** [*user*]: l'utilisateur normal, son propriétaire, bien souvent son créateur, qui n'a pas pour autant tous les droits sur lui !
  - **g** [*group*]: son groupe, ensemble d'utilisateurs ayant parfois des "permissions" particulières ;
  - **o** *others* : tous les autres.

Attention, l'utilisateur propriétaire et le groupe propriétaire du fichier peuvent être indépendants :

Le groupe propriétaire n'est pas forcément le groupe primaire de l'utilisateur propriétaire ;

Et même, le propriétaire n'est pas forcément membre du groupe !

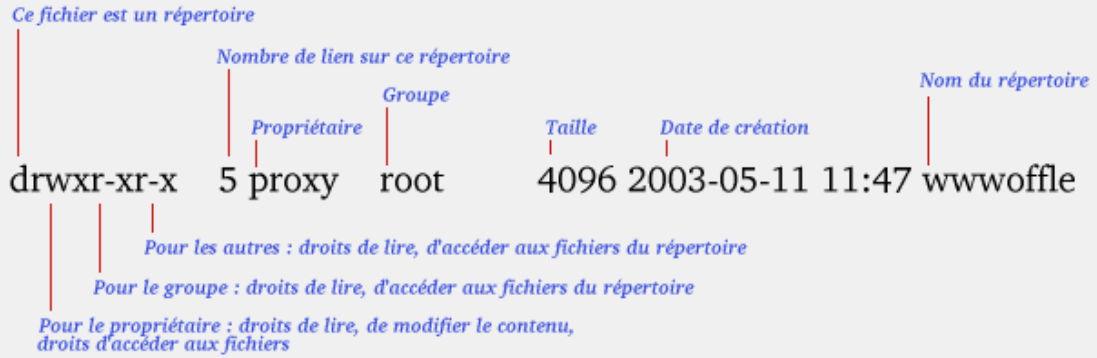
Mais heureusement une règle générale simple s'applique à la création de tout nouveau fichier (ou répertoire) :

Son propriétaire est l'utilisateur (humain ou système) qui l'a créé ;

Son groupe est le groupe primaire de ce même utilisateur.

Figure 6. Exemple cas d'un fichier

Figure 7. Exemple cas d'un répertoire



Changer les droits d'un fichier

chmod [-R] mode_octal fichier

Changer le propriétaire d'un fichier

chown [-R] proprietaire fichier

Changer le groupe d'un fichier

chgrp [-R] groupe fichier

Le Shell est un interpréteur de commandes qui :

- initialise l'environnement
- génère le prompt

Quand une commande est validée, le Shell :

1. effectue les substitutions de variables
2. interprète les Meta caractères
3. gère les redirections et les pipes
4. effectue les substitutions de commandes,
5. exécute la commande.

Il existe plusieurs Shell sous Unix :

- Bourne Shell « sh » ancêtre de tous les Shell, utilisés seulement pour l'écriture de procédures. Il n'offre aucune facilité pour l'emploi en mode interactif (pas d'historique de commandes, pas de rappels avec les flèches, etc.).
- C Shell « csh » plutôt conçu pour une interface avec les utilisateurs. Il permet le rappel des commandes avec les flèches, de gérer une historique des commandes, etc.
- Korn Shell « ksh » est une extension du Bourne Shell avec une partie des possibilités du C Shell.
- Bourne Again Shell « bash » est une variante du Bourne Shell, disponible dans le domaine public

Vi est un éditeur de textes plein écran présent sur la grande majorité des systèmes UNIX™.

il possède **plusieurs modes de fonctionnement:**

- le mode commande , dans lequel vi démarre
- le mode insertion

Syntaxe : vi nom_fichier

Il possède plusieurs type de commande :

Insertion, concatenation,déplacement,recherche chaine de caractère,repmlacement…

Commandes d'insertion de texte

i	insert	insertion avant le curseur
I	insert	insertion au début de la ligne
a	append	insertion après le curseur
A	append	insertion à la fin de la ligne
o	open	ouvre une ligne blanche en dessous de la ligne courante
O	open	ouvre une ligne blanche au dessus de la ligne courante

Pour abandonner le mode insertion et retourner au mode commande, appuyer sur la touche **<Echap>**

<ctrl/v>	insert	insertion d'un caractère de contrôle

Commande de concaténation

J	concatenate	concatène la ligne suivante à la fin de la ligne courante

Commandes de déplacements

/\	curseur	déplacement d'une ligne vers le haut
\/	curseur	déplacement d'une ligne vers le bas
<=	curseur	déplacement d'un caractère vers la gauche
=>	curseur	déplacement d'un caractère vers la droite
e	end	avance à la fin du mot suivant
w	word	avance au début du mot suivant
<ctrl>F	forward	page suivante
<ctrl>B	backward	page précédente
^ ou 0		début de la ligne courante
$		fin de la ligne courante

Editeur des fichiers VI

Recherche d'un chaîne de caractères

/toutou		recherche la chaîne "toutou" à partir de la position actuelle du curseur vers le bas du fichier
?toutou		recherche la chaîne "toutou" à partir de la position actuelle du curseur vers le haut du fichier
n	new	recherche la prochaine occurence de la chaîne "toutou" *recherche vers le bas du fichier*
N	new	recherche l'occurence précédente de la chaîne "toutou" *recherche vers le haut du fichier*

Suppression et utilisation du buffer

x		suppression du caractère sous le curseur
X		suppression du caractère précédent le curseur
dd	delete	suppression de la ligne courante
yy	yank	copie la ligne courante dans le buffer
p	paste	colle la ligne contenu dans le buffer après la ligne courante
P	paste	colle la ligne contenu dans le buffer avant la ligne courante
u	undo	annule la dernière commande et seulement la dernière

s commandes **x, X, dd, et yy** peuvent être précédées d'un facteur multiplicateur
leur action. Par exemple **7dd** supprimera la ligne courante et les 6 lignes suivantes.

Remplacement

r	replace	remplace le caractère sous le curseur par un nouveau caractère
R	replace	remplace tous les caractères par de nouveaux caractères pour terminer appuyer sur la touche **<Echap>**
C	change	remplace la fin de la ligne par de nouveaux caractères pour terminer appuyer sur la touche **<Echap>**

Editeur des fichiers VI

Partage de l'écran en deux

Le partage de l'écran rend pratique le copier/coller entre deux fichiers.

:split	partage l'écran en deux
<ctrl/w> <ctrl/w>	passer d'un demi-écran à l'autre
:e second_fichier	lit le second fichier dans le demi-écran choisi

---

Expressions régulières et mode commande globale

Les expressions régulières servent à manipuler le fichier texte dans son ensemble.

.	représente un caractère quelconque
*	multiplicateur du caractère précédent
^	début de ligne
$	fin de ligne ou fin de fichier
\	permet de représenter les caractères . * ^ $ en le mettant devant. Exemple : * ou \$
\1	permet la réécriture de l'expression régulière lors d'une substitution
s	commande de substitution
d	commande de destruction de lignes
:	passage en mode commande globale
g	parcours global du fichier ou de la ligne

Exemples :

:1,$s/neant/bof/g	de la ligne 1 à la dernière ligne, substitution de la chaîne **neant** par la chaîne **bof**
:1,$s/neant/bof/	idem mais seulement pour la première occurence de la chaîne **neant** par ligne
:1,3s/^.*=/BRAVO/	de la ligne 1 à la ligne 3, substitution du début de la ligne (^) jusqu'au (.*) caractère = par la chaîne **BRAVO**
1,$s/B.*O/\1 Veinard/	rajoute aux mots commençant par B et terminant par O, le mot "Veinard"
:1,$s/.$//	suppression du dernier ($) caractère quel qu'il soit (.)
:10,20d	suppression des lignes 10 à 20
:g/^#/d	suppression des lignes commençant par un #
:g/<ctrl/v>^M/s//<ctrl/v>^M/g	suppression des ^M en milieu de ligne et substitution par un vrai retour à la ligne. *Taper sur "Enter" pour obtenir le ^M.*

Editeur des fichiers VI

Divers mais utiles

:set showmode	indique le mode (insertion ou commande) en bas de la fenêtre
:set number	affiche le numéro de chaque ligne
:155	positionne sur la ligne 155
:set filetype=unix	transforme le type DOS d'un fichier en type UNIX

Sauvegarder et quitter

:w	write	sauvegarde le fichier en cours d'édition
:q	quit	abandonne l'édition
:q!	quit	abandonne l'édition sans sauvegarder le fichier
:x	exit	sauvegarde le fichier en cours d'édition et abandonne l'édition (idem **:wq**)