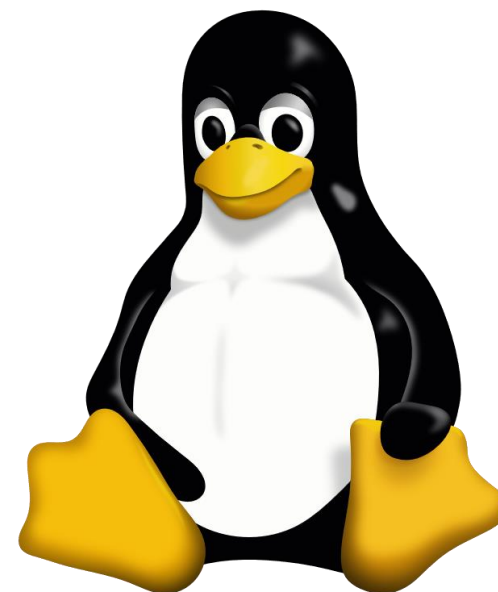


Introduction à Unix - GNU/Linux

Dr Nils Beaussé

nils.beausse@isen-ouest.yncrea.fr

Bureau au 2ème étage (ou voir en salle recherche A1-50)



Le système de fichier : rappel

Propriétés standard d'un fichier :

- Taille
- UID du propriétaire
- GID du propriétaire
- Nombre de liens physiques
- Permissions
- Dates

Le système de fichier (3)

- Permissions, "la première ligne de défense" :
- Droits d'accès :
 - Lecture (Read : r)
 - Écriture (Write : w)
 - Exécution (x)



Le système de fichier (4)

Permissions :

- Trois niveaux de permissions :
 - Permissions du propriétaire du fichier (**U**ser)
 - Permissions du groupe propriétaire (**G**roup)
 - Permissions pour les autres (**O**ther)

Le système de fichier (5)

- Permissions: Exemple

drwxrwxrwt	13	root	root	4096	Aug	22	14:59	.
drwxr-xr-x	18	root	root	4096	Aug	20	08:32	..
-rw-rw-r--		1	edr	users	100	Aug	22	11:51
	File1							
-rw-rw-r--		1	edr	edr	1000	Aug	22	14:59
	File2							
drwxrwxr-x	2	edr	users	4096	May	15	09:02	esd
-rwxrwxr-x	2	root	bin	4096	Aug	20	08:33	ls

- Les noms de fichiers commençant par un point sont masqués.
- Le répertoire « . » représente le répertoire courant.
- Le répertoire « .. » représente le répertoire parent.

Le système de fichier (6)

- Modes spéciaux :
 - Le mode "sticky bit" (t au lieu de x dans le dernier champ (autres)) : appliqué aux répertoires, cela signifie que seul le propriétaire d'un fichier peut le supprimer.
 - Le mode "set uid" (s sur le champ d'exécution du mode utilisateur du propriétaire) : le fichier s'exécutera sous l'identifiant (ID) du propriétaire du fichier.

Le système de fichier (7)

- Modes spéciaux :
 - Le mode "set gid" (s sur le champ d'exécution du mode utilisateur du groupe) :
 - Appliqué à un fichier : le fichier s'exécutera sous l'identifiant (ID) du groupe propriétaire du fichier.
 - Appliqué à un répertoire : les nouveaux fichiers seront la propriété du groupe défini.

Le système de fichier (8)

- Changer les permissions :
 - Commande : `chmod [droits] fichier(s)`
- Première syntaxe : utilisation du mode octal
- 3 chiffres octaux : un pour chaque mode utilisateur (premier bit = exécution, deuxième bit = écriture, troisième bit = lecture)
- 4 chiffres octaux : un pour les modes spéciaux (premier bit = sticky, deuxième bit = setgid, troisième bit = setuid), un pour chaque mode utilisateur.

Le système de fichier (9)

- Changer les permissions :
 - Commande : `chmod [droits] fichier(s)`
 - Deuxième syntaxe : ajout ou suppression de droits ; mode caractère
 - user mode: u(ser), g(roup), o(ther) or a(ll)
 - access rights: r(ead), w(rite), (e)x(ecute), s(et), (s)t(icky)
 - action : +, -, =

Le système de fichier (10)

- Exemples :

- Mode Octal :

- `chmod 744 myfile`
 - `chmod 4755 myexec`

- Mode caractere:

- `chmod g+r myfile`
 - `chmod g=rw,o=r myfile`
 - `chmod u+s myexec`

Le système de fichier (11)

- Définition des permissions par défaut sur le système :
 - À travers une commande définie dans l'un des scripts de connexion :
 - Commande :

umask masque-octal

- Les permissions réelles sont obtenues grâce à une opération XOR avec un mot octal 777.

- Exemples :

umask 002 ⇔ rwxrwxr-x

umask 113 ⇔ rw-rw-r--

Le système de fichier (12)

- Changer la propriété des fichiers (réservé à l'utilisateur root) :

- Commande :

chown user[:group] file(s)

- Changer le propriétaire et le groupe propriétaire du fichier.

- Commande :

chgrp group file(s)

- Changer le groupe propriétaire du fichier.

Le système de fichier (13)

Modification des attributs de date :

- Commande : **touch [option(s)] fichier(s)**
 - Si le fichier n'existe pas, crée un fichier vide.
 - Si le(s) fichier(s) existe(nt) :
 - Option -a : modifie l'heure d'accès du(des) fichier(s).
 - Option -m : modifie l'heure de modification du(des) fichier(s).

Le système de fichier (14)

- Création de "liens physiques" (hard links) :

- Commande :

`ln fichier_source fichier_destination`

- Limitations :
 - Le fichier_source et le fichier_destination doivent être des fichiers "normaux".
 - Le fichier_source et le fichier_destination doivent être sur le même système de fichiers.

Le système de fichier (15)

- Création de "liens symboliques (liens symboliques souples)" :

- Commande :

ln -s fichier_source fichier_destination

- Veuillez noter :

- Le fichier_source peut être de n'importe quel type.
- Le fichier_source et le fichier_destination peuvent être sur des systèmes de fichiers différents.
- Le fichier_source peut même ne pas exister.

Le système de fichier (16)

- Recherche de fichiers dans une hiérarchie de répertoires :
- Commande :
find répertoire expression
- L'expression est composée d'options, de tests et d'actions.

Le système de fichier (17)

- *-name pattern*
- *-links N*
- *-user name*
- *-nouser*
- *-group name*
- *-nogroup*
- *-size N*
- *-perm octal_value*
- *-type T*
- *-atime N*
- *-mtime N*
- *-ctime N*
- *-print*
- *! Condition*
- *cond1 -o cond2*
- *cond1 [-a] cond2*
- *-exec command \;*

Le système de fichier (18)

- Exemples:
- `find /home/isen ! -user isen -print`
- `find /home/isen/ ! -user isen -exec rm {} \;`
- `find /tmp -atime +7 -exec rm {} \;`

Le système de fichier (19)

Périphériques :

- Sont présentés au système sous forme de fichiers spéciaux dans le répertoire /dev.
- Sont divisés en deux groupes différents :
 - Périphériques de bloc : généralement des périphériques stockant des données (disques, CD, DVD, etc...)
 - identifiés par un caractère 'b' dans la liste des répertoires.

Le système de fichier (20)

Périphériques :

- Sont présentés au système sous forme de fichiers spéciaux dans le répertoire /dev.
- Sont divisés en deux groupes différents :
 - Périphériques de caractères : généralement des périphériques transmettant des données (claviers, souris, modems, etc...)
 - Identifiés par un caractère 'c' dans la liste des répertoires.

Le système de fichier(21)

- Périphériques :
- Possèdent deux attributs supplémentaires :
 - Le numéro majeur : identifie l'entrée du pilote de périphérique dans le noyau.
 - Le numéro mineur : identifie le périphérique au sein du pilote de périphérique.

Le système de fichier(22)

- Exemples :

- ls -al /dev | less

- brw-rw---- 1 root disk 3, 0 Mar 24 05:37 /dev/hda
- brw-rw---- 1 root disk 3, 1 Mar 24 05:37 /dev/hda1
- crw—w---- 1 bla tty 4, 1 Aug 22 08:41 /dev/tty1
- crw----- 1 root root 4, 2 Aug 22 08:41 /dev/tty2
- crw----- 1 root root 4, 3 Aug 22 08:41 /dev/tty3

Le système de fichier(23)

- Pour accéder à un périphérique contenant des données, vous devez l'associer à un répertoire du système de fichiers. Vous devez spécifier :
 - Le type de système de fichiers (ext3, vfat, ...)
 - Le type d'accès (ro pour lecture seule, rw pour lecture/écriture)
- Commande : mount
 - Exemple :

```
mount -t ext3 -o rw /dev/hda4 /mnt/data
```
- Cette commande monte le périphérique /dev/hda4 avec le système de fichiers ext3 en mode lecture/écriture dans le répertoire /mnt/data.

Le système de fichier (24)

- Une fois monté, les données peuvent être accédées de manière transparente comme si elles faisaient partie de la structure d'arborescence d'origine.
- Pour "libérer" un périphérique, vous devez le démonter (unmount) à l'aide de la commande `umount`. Vous pouvez utiliser l'une des deux syntaxes suivantes :
- Commande :
`umount nom_du_périphérique`
- ou
`umount point_de_montage`

Le shell

- Le shell : un programme d'interface en ligne de commande (CLI) utilisé pour saisir des commandes sur un système UNIX.
- Différents shells :
 - Le Bourne Shell
 - Le C shell
 - Le Korn Shell
 - Etc...

Le shell (2)

- Chaque utilisateur a un shell par défaut appelé le shell de connexion, défini dans le fichier /etc/passwd.
- Un utilisateur peut exécuter d'autres shells.
- Un utilisateur peut également changer son shell par défaut à l'aide de la commande suivante :
- Commande :
`chsh -s programme_shell`
- Avertissement : cette commande peut être extrêmement dangereuse, cela peut empêcher le script de login de s'exécuter.

The Bourne shell

- Trois flux I/O:
 - Descripteur 0: stdin (Par défaut : Le Clavier)
 - Descripteur 1: stdout (Par défaut : L'affichage)
 - Descripteur 2: stderr (Par défaut : L'affichage)

The Bourne shell (2)

- Les flux peuvent être redirigés de la manière suivante :
 - Redirection du flux de sortie : > or >>
 - Redirection du flux d'entrée : < or <<
 - Redirection du flux d'erreur : 2> or 2>>

The Bourne shell (3)

- Le système de pipes :
 - Les flux de sortie peuvent être chaînés aux flux d'entrée en utilisant des « tubes » plus connus sous le terme anglais de pipes :
 - `commande 1 | commande 2 | .... | commande n`

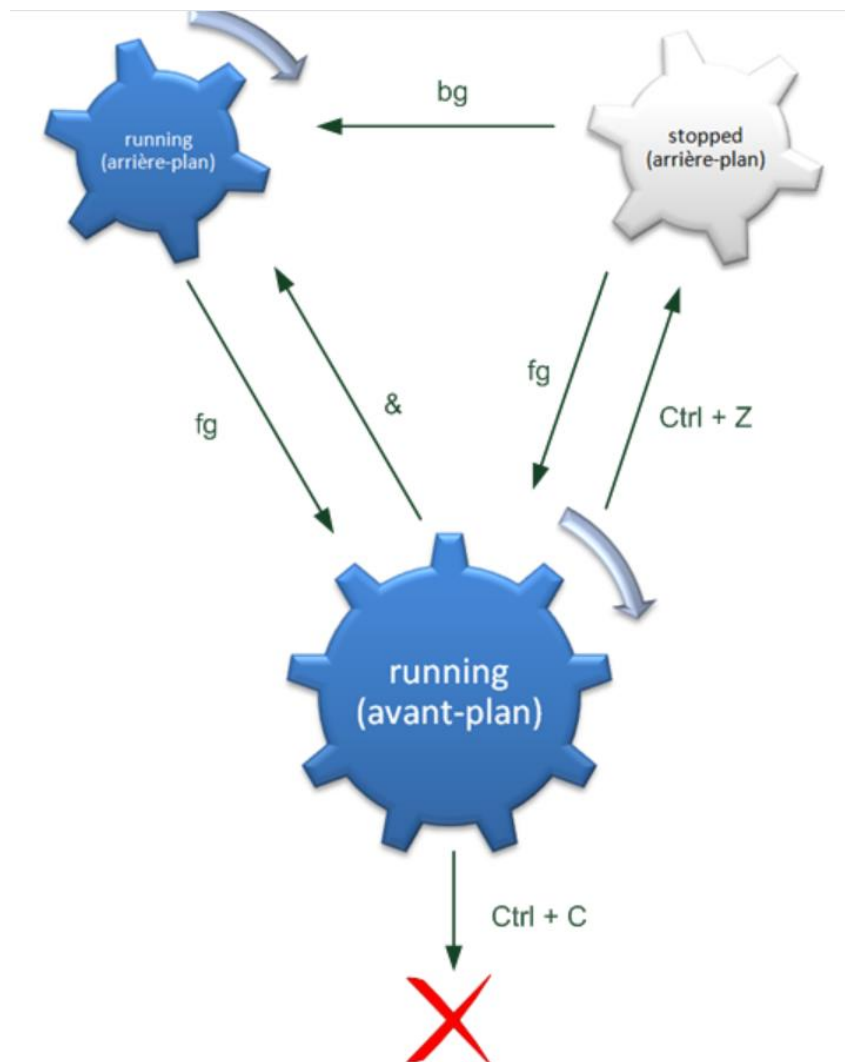


The Bourne shell (4)

Contrôle des processus :

- Exécution d'un processus en arrière-plan :
commande &
- Passage de processus en arrière-plan ou en premier plan :
Commandes : bg [jobspec] et fg [jobspec]
- Informations sur les tâches (jobs) :
Commande : jobs [options]
- Arrêt des tâches : ctrl-z

The Bourne shell (5)



The Bourne shell (6)

Variables :

- Définition :
 - Syntaxe :

`nom_de_variable = valeur`

- Affichage du contenu :
 - Commande :

`echo $nom_de_variable`

- Exemple :

```
> johndoe = bonjour
> echo $johndoe
> bonjour
```

The Bourne shell (7)

Variables :

- Portée (Scope) :
- Une variable n'est définie que dans l'instance actuelle du shell.
- Pour la rendre définie dans les shell enfant, vous devez l'exporter :

Commande :

```
export nom_de_variable
```

- Gestion des nombres :
 - Syntaxe :

```
declare -i nom_de_variable [=valeur]
```

The Bourne shell (8)

- Quelques variables prédéfinies :
 - \$HOME: Le repertoire home de l'utilisateur
 - \$PATH: Les chemins dans lequel sont recherché les noms des programmes
 - (exemple : si on tape vi, vi est un executable present quelque part dans le système : l'interface ne peut pas deviner ou !)
 - \$PS1: Texte utilisé pour l'affichage du shell

The Bourne shell (9)

- Caracteres spéciaux :
 - Command > file
 - Command >> file
 - Command < file
 - Command1 | Command2
 - << *string*
 - * # Correspond à n'importe quelle chaine de caractères
 - ? # Correspond à n'importe quelle caractère seul

The Bourne shell (10)

- [...] # Correspond à n'importe lequel des caractères contenus dans les crochets.
- Command1 ; Command2
- Command &
- (list) # La liste est exécutée dans un subshell
- { list; } # La liste est exécutée dans l'environnement courant
- \$#, \$0, \$1...\$9
- \$variable ou \${variable}

The Bourne shell (11)

- \c
- '....'
- "...."
- #commentaires
- variable=value
- Command1 && Command2 # execute
command2 only if command1 returned 0
- Command1 || Command2 # execute
command2 only if command1 didn't return 0

- ***En avant !***

TP2

- Shell
- *man / echo / who / hostname / date / history / alias*
- *ls / pwd / cd / cp / mv / rm / mkdir / rmdir*
- Fichiers cachés
- Métacaractères, globbing
- *ln / ln -s / wc -l / diff / find*
- *chmod / umask / chown / chgrp*

TP3

- *ps / top / pstree / <ctrl-c>*
- *> 2> >>*
- *|*
- Filters
- *tac / head / tail*
- *sort*
- *grep*
- Regular expressions
- *tr, cut*