

Introduction aux SE et à leur fonctionnement

CM2 : Systèmes d'exploitation et shell

Mickaël Martin Nevot

V1.6.1



Cette œuvre de [Mickaël Martin Nevot](#) est mise à disposition selon les termes de la [licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage à l'Identique 3.0 non transposé](#).

Introduction aux SE et à leur fonctionnement

- I. Présentation
- II. Intro.
- III. OS et shell
- IV. Installation
- V. Apps

Architecture de référence

Couches
hautes

- Couche des applications
- Couche distribution

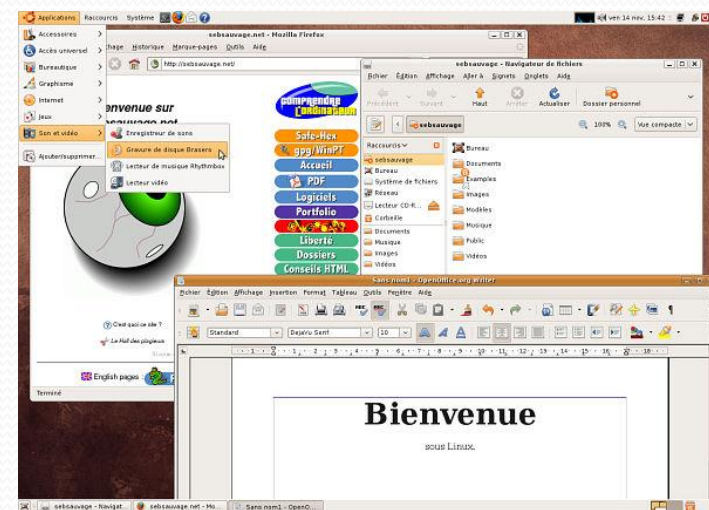
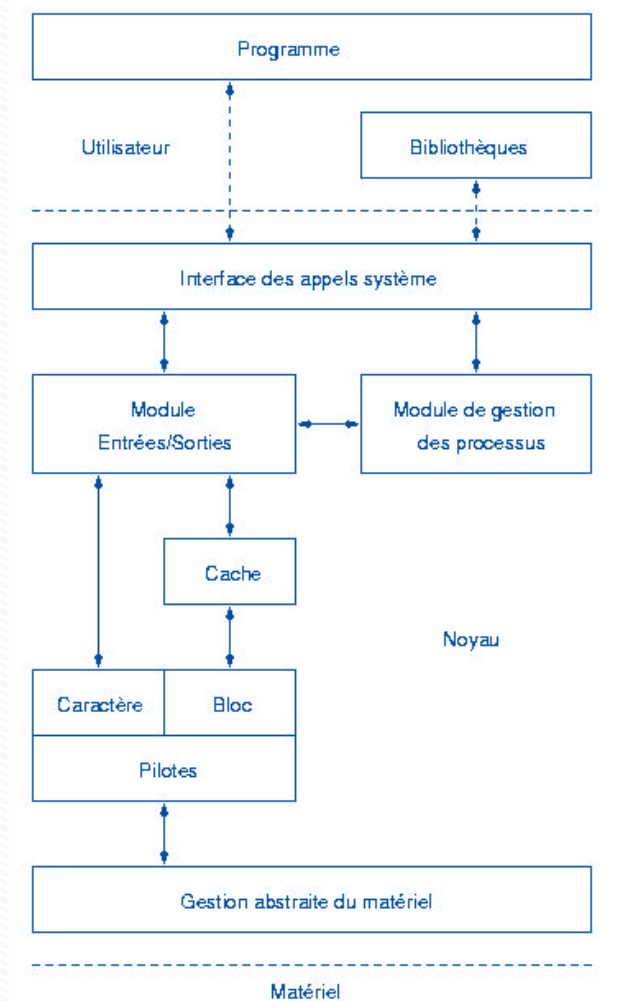
Couche
système/réseau

- Services additionnels
- Noyau

Couches
basses

- Couche du microcode et du langage machine
- Couche du matériel

Architecture d'un système Unix



Fonctions d'un OS

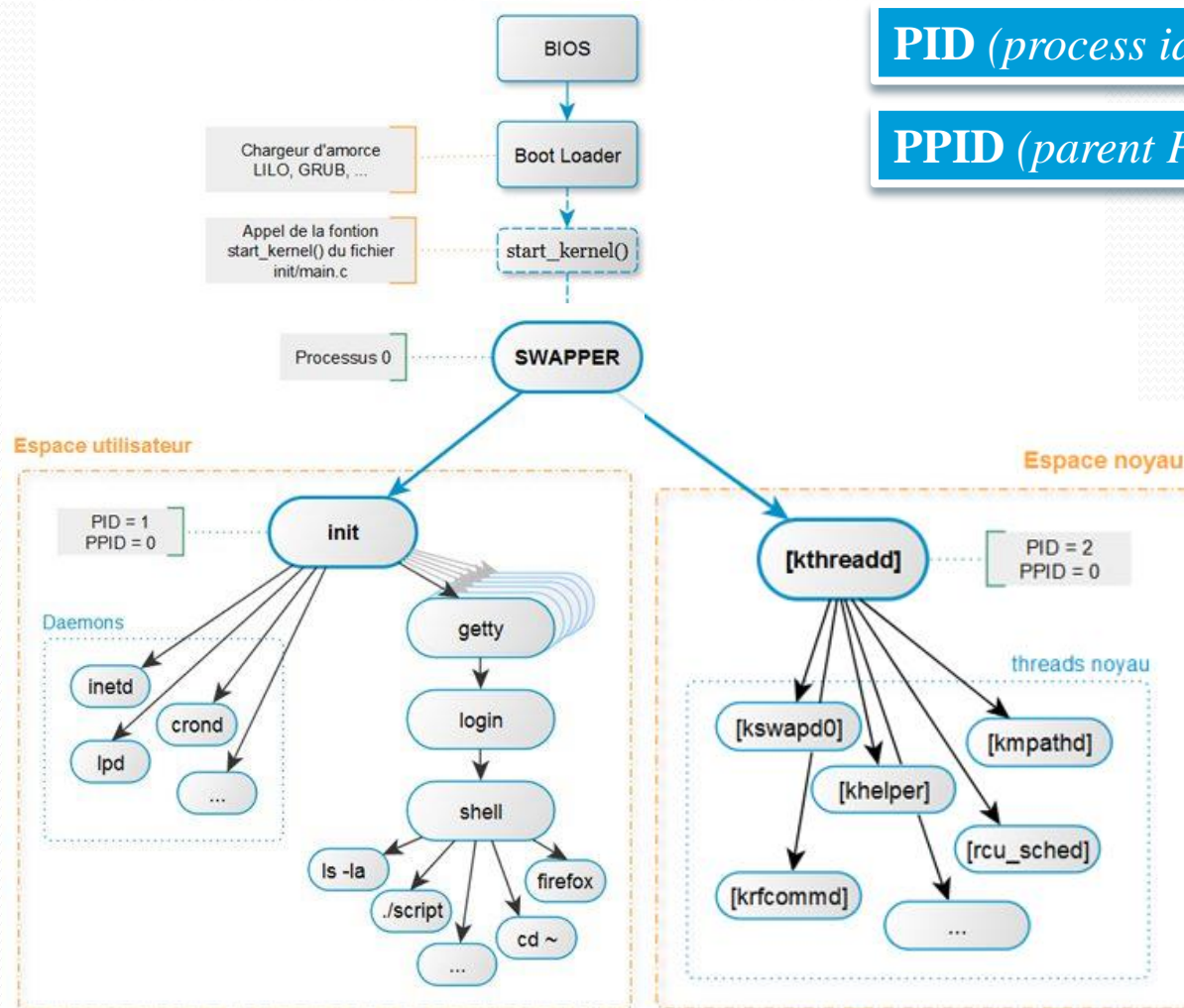
Fonction	Description	Ressources matérielles	Concepts abstraits
Gestion des activités	Création, destruction, suivi des activités	Processeur, programme, mémoire centrale, interruptions	Processus
Gestion des données persistantes	Accès, stockage, partage des données	Disque, sauvegardes	Fichier, répertoires
Gestion des communications	IHM (interface homme/machine), entrées/sortie, réseaux	Clavier, écran, souris Périphériques Infrastructures réseau	Fenêtres Flots de données
Gestion de la sécurité (données, ressources)	Intégrité, confidentialité, disponibilité		Utilisateurs, droits

Commandes Unix générales

Code	Description	Exemple
echo	Afficher une ligne de texte	echo Salut ça va ?
ls	Lister les fichiers d'un répertoire	ls ou ls -l ou ls -a ou ls -al
cd	Changer de répertoire courant	cd dir ou cd ou cd ~ ou cd ..
xterm	Lancer un terminal	xterm
clear	Effacer l'écran du terminal	clear
wc	Compter caractères / mots / lignes	wc -l file
sort	Trier des fichiers	sort -n nb (tri numérique)
<, >, >>, 2>, 2>>, &>, &>>	Redirection d'entrées, de sorties, de sorties d'erreur et des deux sorties	ls > file ou sort < file1 >> file2
	Canal de communication unidirectionnel entre processus	ls -l grep key less
;	Lancement séq. de commandes	echo "Hello " ; echo "world"
alias	Lister les alias ou en créer	alias la='ls-A'

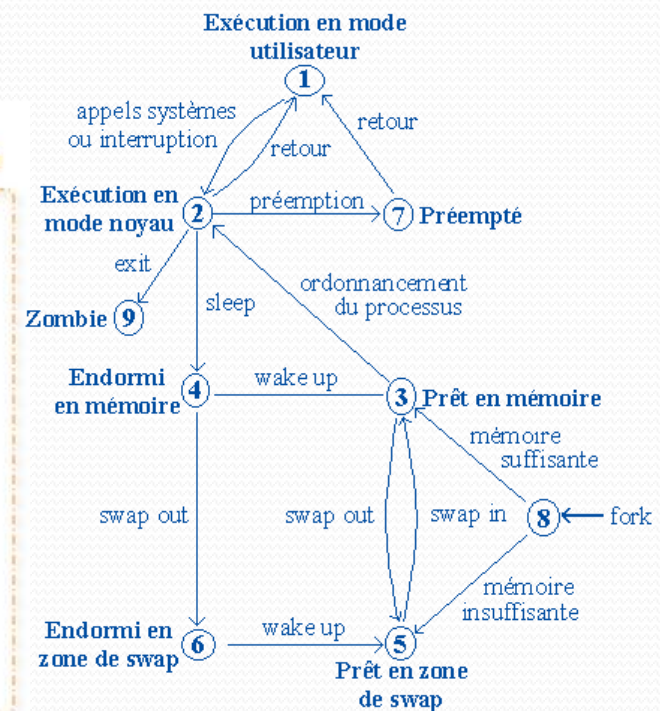
CTRL + L

Gestion des activités



PID (*process identifiant*) : identifiant unique

PPID (*parent PID*) : identifiant unique



Mythologie des processus



Zombie

Achevé (défunt)



Orphelin

Qui a perdu son père



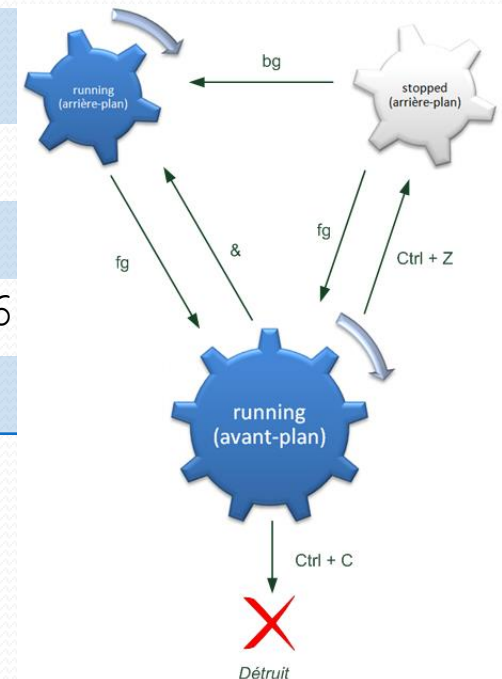
Démon

N'est pas contrôlé
par l'utilisateur

Gestion des activités

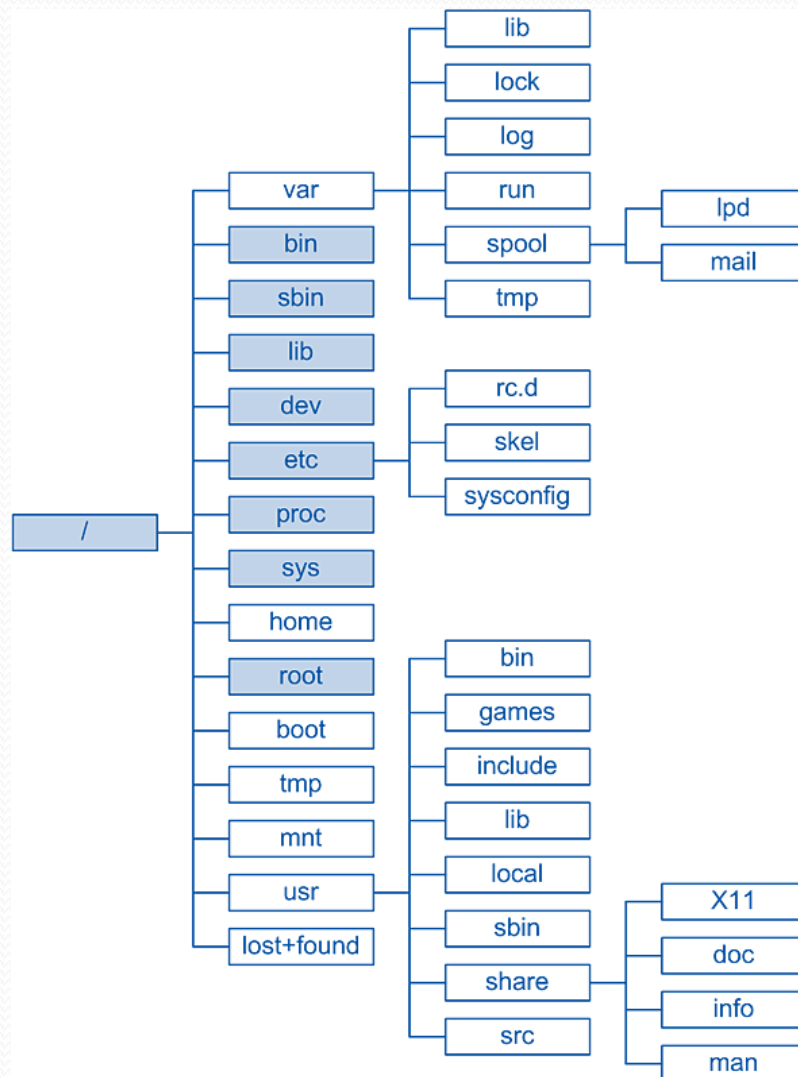
Code	Description	Exemple
------	-------------	---------

ps	Afficher l'état des processus	ps ou ps -e ou ps -Af ou ps auxfw
bg	Placer un processus en arrière-plan	bg %1
fg	Passer un processus d'arrière-plan en premier plan	fg %1
jobs	Afficher le processus en arrière-plan	jobs
&	Lancer un programme en arrière-plan	xterm &
kill	Envoyer un signal à un processus	kill -9 666
top	Afficher les processus en continu	top



Signal	Valeur	Action	Commentaire
SIGHUP	1	Term	Déconnexion détectée sur le terminal de contrôle ou mort du processus de contrôle.
SIGINT	2	Term	Interruption depuis le clavier.
SIGQUIT	3	Core	Demande « Quitter » depuis le clavier.

Gestion des données persistantes



var : fichiers variables

- log : journaux systèmes
- lpd : files d'impression
- mail : *e-mails*

bin (*binaries*) : commandes « externes »

sbin (*system bin*) : commande d'admin.

dev (*devices*) : périphériques

etc (*editable text configuration*) : fic. de conf.

home : répertoires personnels des utilisateurs

root : répertoire personnel de l'admin.

mnt (*mount*) : systèmes de fichiers montés

usr (*unix system ressources*) : logiciels installés

Gestion des données persistantes

Code	Description	Exemple
<code>pwd</code>	Afficher le répertoire courant	<code>pwd</code>
<code>touch</code>	Créer un fichier vide ou modifier les dates d'un fichier	<code>touch file</code>
<code>cp</code>	Copier des fichiers	<code>cp file dir</code>
<code>mv</code>	Déplacer ou renommer des fichiers	<code>mv file1 file2</code> ou <code>mv file dir</code>
<code>rm</code>	Supprimer un fichier	<code>rm file</code> ou <code>rm -i file</code>
<code>mkdir</code>	Créer un répertoire	<code>mkdir dir</code>
<code>rmdir</code>	Supprimer un répertoire	<code>rmdir rep</code>
<code>cat</code>	Afficher et concaténer des fichiers	<code>cat file</code> ou <code>cat file1 file2</code>
<code>head</code>	Afficher le début d'un fichier	<code>head -n 20 file</code> (20 lignes)
<code>tail</code>	Afficher la fin d'un fichier	<code>tail -n 20 file</code> (20 lignes)
<code>diff</code>	Comparer des fichiers / répertoires	<code>diff file1 file2</code> ou <code>diff dir1 dir2</code>

Gestion des données persistantes

Code	Description	Exemple
grep	Rechercher dans des fichiers	grep Salut file
find	Rechercher selon des critères multiples	find /usr/src -name ".c"
sed	Editer du texte non interactivement	sed 's/Mr./M./g' file.txt (remplacement global)
cut	Afficher des colonnes ou champs	cut -c 1-3 file (3 caractères)
.	Répertoire courant	find . -ls
..	Répertoire parent	cd ..
~	Répertoire personnel	cd ~

Chemin :

- Absolu : invariant, commence par / (ou ~)
- Relatif : variable

Extension de fichier sans importance

Globbing patterns (filtres)

- Caractère de filtrage (métacaractères, *globs*) :
 - ? : exactement un caractère quelconque
 - * : n'importe quel nombre de caractères (y compris zéro)
 - [*ab...*] : un unique caractère au choix dans la liste
 - [!*ab...*] : un unique caractère au choix **absent** de la liste

Liste : *ab...* ou a-z

Code	Description	Exemple
' , " , \	Echappement des caractères spéciaux	ls 'a*' ou ls "x y" ou echo -e "x\ty"
`	Evaluation préalable de la commande	chown `id -u` dir

Expressions rationnelles

- **Chaîne de caractères appelé motif (ou *pattern*) qui décrit un ensemble de chaînes de caractères possibles**
- Pas de standardisation
- Symboles spéciaux communs :
 - . : n'importe quel caractère
 - \ : échapper (banaliser) un caractère
 - [] : un des caractères au choix parmi ceux entre crochets
 - ^, \$: début/fin (^a, a\$)
 - Quantifieurs :
 - ? : zéro ou une fois
 - * : zéro, une ou plusieurs fois
 - + : une ou plusieurs fois

Expression rationnelle \neq caractères de filtrage

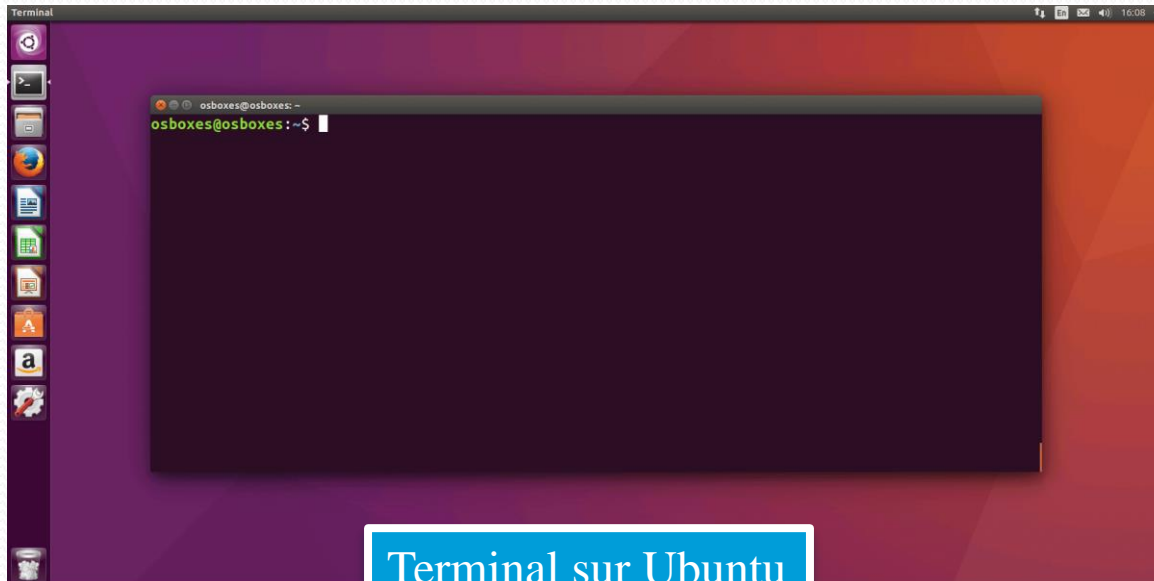
Expressions rationnelles

- Exemple :

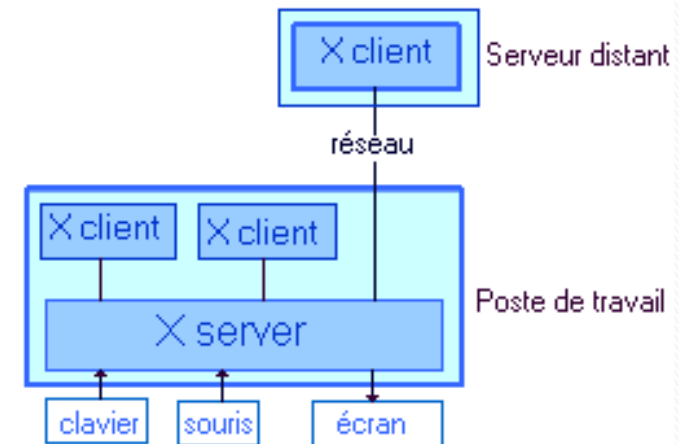
- `a.b`
- `b.1..re`
- `[0-9][0-9]aaa.*`
- `[a-z]+-[a-z]+`
- `^[0-9]*`
- `crée(|e|es|s)`
- `^r.*[0-9]$`
- `.*[aeiouy](1)[aeiouy].*`
- `^[a-z0-9._-]+@[a-z0-9._-]{2,}\.[a-z]{2,4}$`

Gestion des communications

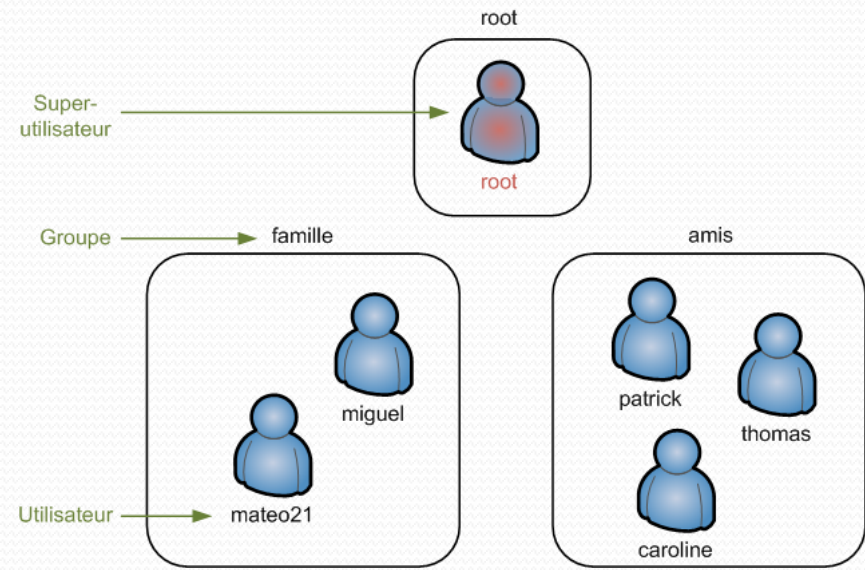
- IPC : communication inter-processus
 - Locaux : signaux, tubes
 - Distants : *sockets* ← Hors du cadre du cours
- Communication homme-système



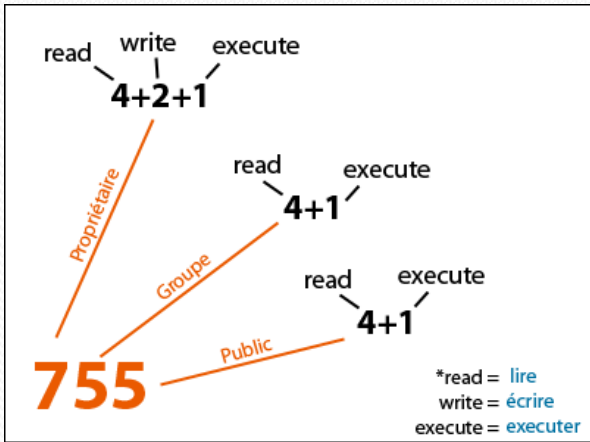
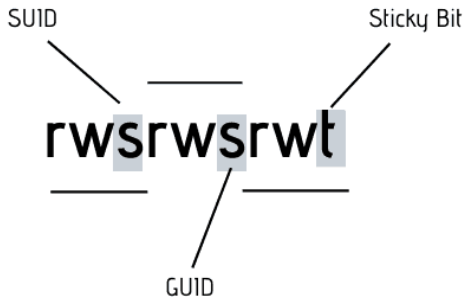
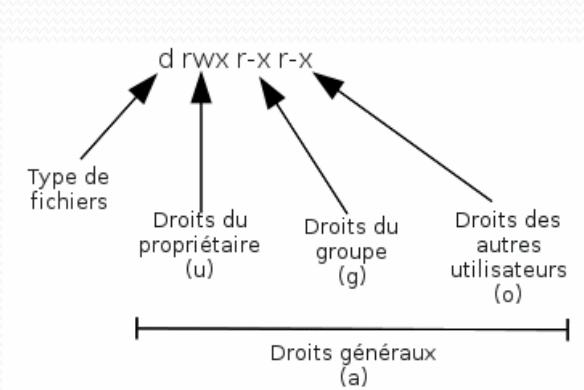
Terminal sur Ubuntu



Gestion de la sécurité



Binaire r w x	Décimal	Droits correspondants
0 0 0	0	aucun
0 0 1	1	exécution
0 1 0	2	écriture
0 1 1	3	écriture & exécution
1 0 0	4	lecture
1 0 1	5	lecture & exécution
1 1 0	6	lecture & écriture
1 1 1	7	lecture, écriture, exécution



Sous Unix « Tout est fichier »

Fichiers « cachés » commence par .

```
ls -l -R -s -r -- 1 root root 0 Mar 15 03:35 sqid-file
                  2 root root 4096 Mar 15 03:32 sqid-folder
                  1 root root 0 Mar 15 03:35 sticky-bit-file
                  2 root root 4096 Mar 15 03:32 sticky-bit-folder
                  1 root root 0 Mar 15 03:35 suid-file
                  2 root root 4096 Mar 15 03:32 suid-folder
```

Gestion de la sécurité

Code	Description	Exemple
whoami	Afficher nom / identifiant de l'utilisateur	whoami
chmod	Changer les droits d'accès	chmod 644 file
umask	Définir les droits par défaut	umask 0022
id	Afficher les attributs UID, GID...	id
finger	Afficher les données d'un utilisateur	finger user
groups	Afficher les groupes d'un utilisateur	groups ou groups user
sudo	Exécuter comme root	sudo apt-get update

Gestion de la sécurité

Action	Utilisateur	Groupe
Créer	useradd user	groupadd group
Modifier	usermod user	groupmod group
Supprimer	userdel user	groupdel group
Changer mot de passe	passwd user	gpasswd group

Beaucoup de distributions offre des commandes interactives : adduser et addgroup

```
himanshu@ansh:~$ sudo adduser htf
[sudo] password for himanshu:
Adding user `htf' ...
Adding new group `htf' (1001) ...
Adding new user `htf' (1001) with group `htf' ...
Creating home directory `/home/htf' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for htf
Enter the new value, or press ENTER for the default
Full Name []: 
```

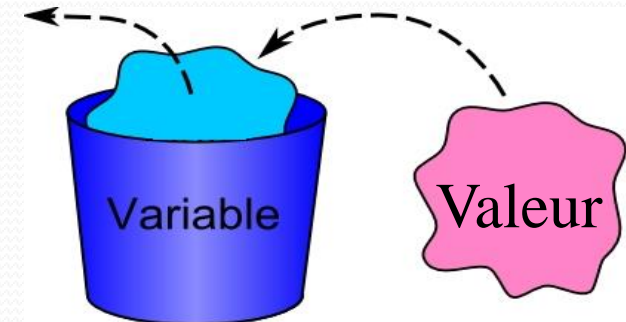
Variable d'environnement shell

- `HOME` : chemin du répertoire personnel
- `PWD` : répertoire courant
- `PS1` : *prompt*
- `PATH` : liste des répertoires de recherche des commandes



\$ Permet d'avoir la valeur d'une variable *shell*

```
1. root@linuxhint: ~ (ssh)
root@linuxhint:~# echo $BASH
/bin/bash
root@linuxhint:~# echo $OSTYPE
linux-gnu
root@linuxhint:~# echo $EUID
0
root@linuxhint:~#
```



.bash_profile, .bashrc, .bash_logout

- Exécuter au démarrage de session
- .bash_profile : au démarrage d'une nouvelle session
- .bashrc : au démarrage d'un nouveau *shell*
- .bash_logout : à l'arrêt d'un *shell*

```
# .bashrc
#
# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

export PS1="$ "
```

Script bash

Script : série de commandes

Langage de script : langage de programmation non compilé

- Exécution :
 - Par défaut : `bash file`
 - Fichier exécutable : `./file` ou `~/file`
 - Script dans `PATH` : `file`
- Code de retour (également avec `exit`) :
 - 0 : script a réussi
 - 1 à 255 : erreur
- Commence par : `#!/bin/bash`
- Commentaire : `#`

Il ne doit absolument rien y avoir avant



BASH
THE BOURNE-AGAIN SHELL

Paramètres, arguments, variables

- Paramètres :

- Paramètres positionnels

```
script arg1 arg2 arg3  
# $1 vaut arg1, $2 vaut arg2, $3 vaut arg3.
```

- Paramètres spéciaux

- \$0 : nom du script
 - \$# : nombre d'arguments du script
 - @\$: liste des arguments du script
 - \$? : code de retour de la dernière commande

- Arguments

- Variables :

```
a=1  
read -p "Tapez votre réponse : " x  
echo $a "Réponse" ${a}m
```

Pas d'espace

Un **paramètre** (paramètre formel, argument muet) est la **variable** utilisée dans un script, une procédure ou d'une fonction

Un **argument** (paramètre effectif) est la **valeur** (variable) fournie dans une ligne de commande ou lors de l'appel d'un script, d'une procédure ou d'une fonction

Script bash

- Tableaux :

```
name[0]='foo'
name[1]='bar'
echo ${name[0]}
# toto
```

- Chaînes de caractères :

```
name='toto'
echo "M. $name"
echo 'M. $name'
echo "M. \"$name\""
```

- Calculs arithmétiques :

```
(( a=1 ))
echo $(( (a+5)/2 ))
```

Attention espaces !

- Fonctions :

```
show() {
    echo 'Hello $1'
    return 5
}
show toto # Hello toto
```

```
bash-3.2$ cat append_date.sh
#!/bin/sh

file_name=test_files.txt

current_time=$(date "+%Y.%m.%d-%H.%M.%S")
echo "Current Time : $current_time"

new_fileName=$file_name.$current_time
echo "New FileName: " "$new_fileName"

cp $file_name $new_fileName
echo "You should see new file generated"

bash-3.2$ ./append_date.sh
Current Time : 2014.12.15-10.31.42
New FileName: test_files.txt.2014.12.15
You should see new file generated with t
bash-3.2$
bash-3.2$ ls test_files.txt*
test_files.txt
```

Structure de contrôle bash

• Conditionnelle : + case

```
if condition ; then
    instructions
else
    instructions
fi
```

• Boucles : + until

• Tant que :

```
while condition ; do
    instructions
done
```

• Pour :

```
for i in $@ ; do
    instructions
done
for (( i=0 ; i<10 ; ++i )) ; do
    instructions
done
```

• Condition : Attention espaces !

- [condition] :
- Test sur les fichiers :
 - -d : répertoire existe
 - -f : fichier ordinaire
 - Etc.
- Chaîne de caractères :
 - ==, !=, <, etc.
- ((condition)) :
 - Numérique :
 - ==, !=, >, <, >=, <=, &&, ||, !

```
if [ -d ~/dir ]
if [ "$name" = "toto" ]
if (( nb > 3 ) && (nb < 8) )
```

Crédits

Auteur

Mickaël Martin Nevot

mmartin.nevot@gmail.com



Carte de visite électronique

Relecteurs

Cours en ligne sur : www.mickael-martin-nevot.com

