



# Système d'exploitation avancé

## Systèmes et généralités

Pierre LEROY – [leroy.pierre1@gmail.com](mailto:leroy.pierre1@gmail.com)

# Sommaire

---

- I. Fondamentaux
- II. Système linux
- III. Notions élémentaires
- IV. Utilisation du shell
- V. Essentiel
- VI. Conclusion

# Définition et usage

- Le rôle du système d'exploitation se définit en **deux** points distincts à savoir :

## INTERFACE (-API)

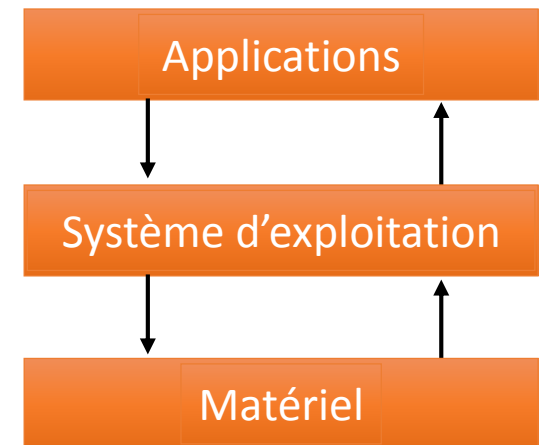
### UTILISATION

- Interface entre le matériel et les applications
  - ✓ Fournit une **API** (Application Programming Interface) à l'utilisateur

## GESTION

### FONCTIONNEL

- Le système doit prendre en charge **5** fonctions essentielles :
  - ✓ La gestion de la **mémoire**
  - ✓ Les **entrées/sorties E/S**
  - ✓ Les **fichiers**
  - ✓ Les **processus**
  - ✓ La gestion des **utilisateurs**



# Définition et usage

---

- Les principaux types d'éléments sont au nombre de **deux** à savoir :

## ELEMENTS

### TYPOLOGIE

- Les mécanismes d'un système d'exploitation reposent sur les types :
  - ✓ Fichier (données [*+ métadonnées*])
  - ✓ Processus (traitement)

# Définition et usage

---

- Un fichier se définit de la sorte :

## ENSEMBLE

### METADONNEES

- Une suite d'octets
- Un ensemble d'attributs dépendant du (*FS*)
  - ✓ Nom
  - ✓ Taille
  - ✓ Type
  - ✓ Propriétaire
  - ✓ Permissions
  - ✓ ...
- Présent et persistable sur un support physique selon une méthodologie définie (*SGF*)
- Le stockage d'un fichier est réalisé par bloc

# Définition et usage

---

- Un processus se définit de la sorte :

## ENSEMBLE

- Programme en cours d'exécution (*fichier exécuté*)
- Propriétaire de métadonnées :
  - ✓ **numero (PID : Process Identifier)**
  - ✓ **numero du pere (PPID)**
  - ✓ chemin de l'exécutable
  - ✓ infos utilisation processeur
  - ✓ infos ordonnancement (priorite, ...)
  - ✓ **fichiers ouverts**
  - ✓ localisation memoire (code, pile, ...)
  - ✓ **proprietaire (uid, gid)**
  - ✓ **code retour**
  - ✓ ...

processus != programme  
*dynamique != statique*

METADONNEES

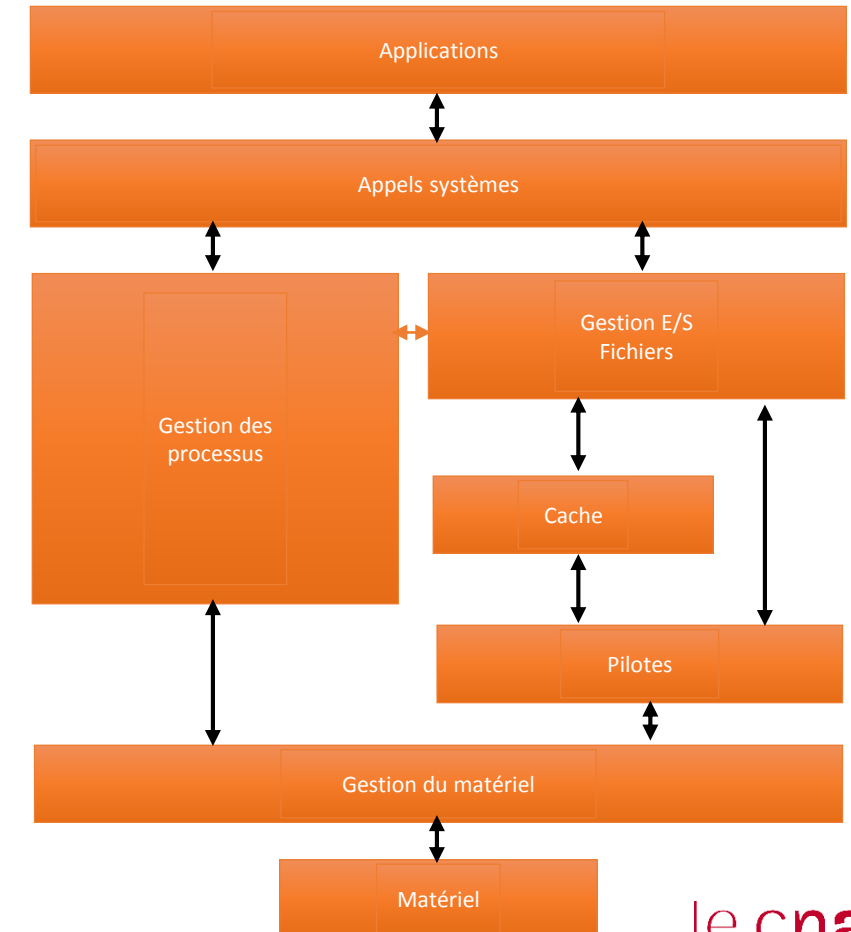
# Définition et usage

## ■ Architecture d'un système UNIX :

### ENSEMBLE

#### 5 FONCTIONS NOMINALES

- Appels systèmes
  - ✓ *Communication avec le S.E.*
- Gestion des processus
  - ✓ Ordonnancement (gestion des priorités, temps CPU)
  - ✓ Commutation des tâches
- Gestion des E/S
  - ✓ Lecture/écriture des données
- Pilotes
  - ✓ Prise en charge du matériel (Gérer en bas niveau le matériel ou les structures logiques associées)



# Définition et usage

- Systèmes d'exploitation usuels :

## ENSEMBLE

### SYSTEMES USUELS

- Windows
  - ✓ *Personnal, Professional, Server etc.*
- Mac OS
- « Famille UNIX »
  - ✓ SUN Solaris, IBM AIX etc.
  - ✓ BSD (OpenBSD, NetBSD, FreeBSD etc.)
  - ✓ Linux





# Sommaire

---

- I. Fondamentaux
- II. Système linux
- III. Notions élémentaires
- IV. Utilisation du shell
- V. Essentiel
- VI. Conclusion

# Engouement && usage

- Systèmes d'exploitation apprécié :

## RAISONS

### SYSTEMES USUELS

- Système d'exploitation libre (open source /GPL)
  - ✓ Multitâches
  - ✓ Multi-utilisateurs
  - ✓ Modifiable (sources disponibles)
- Outil pédagogique
  - ✓ Permet l'observation et la manipulation des concepts de base d'un SE



torvalds / linux

Watch 5,841 Star 49,975 Fork 18,685

Code Pull requests 171 Projects 0 Insights

Linux kernel source tree

706,551 commits 1 branch 527 releases ∞ contributors GPL-2.0

# Historique

- Système d'exploitation apprécié :

## DATES

### MOMENTS CLEFS

- **1964** Projet MULTICS Laboratoire Bell Labs de AT&T
  - ✓ MultiPlexed Information and Computing Service
  - ✓ Temps partagé
- **1969** UNICS Ken Thompson et Dennis Ritchie
  - ✓ Uniplexed Information and Computing System
- **1973** Réécriture de Unix en langage C
  - ✓ Amélioration de la portabilité
- **1974** Diffusion d'UNIX aux universités
  - ✓ Berkeley en Californie
- **1978** Licence plus restrictive
  - ✓ École BSD (Berkeley Software Distribution)
  - ✓ Unix system V
- **1980 à 1994** Guerre des Unix
  - ✓ Procès
  - ✓ Clones d'Unix (Solaris, AIX, HP-UX, Windows NT, ...)
- **1984** Création de X/Open
  - ✓ Tentative de standardisation
- **1988** AT&T et Sun produisent un Unix unifié
  - ✓ System V Release 4
- **1988** IEEE produit IEEE1003
  - ✓ POSIX : Portable Operating System Interface
  - ✓ Tout ce que doit contenir un Unix
- **1993** Novell rachète SVR4
  - ✓ 1993 cède la marque Unix à X/Open
  - ✓ 1995 X/Open et OSF fusionnent

# Historique

---

- Système d'exploitation apprécié :

## DATES

### MOMENTS CLEFS

- **1983** Xenix d'IBM premier Unix sur ordinateur personnel
  - ✓ 1987 MINIX par Andrew Stuart Tanenbaum
  - ✓ 1989 386BSD
- **1983** Richard Stallman du MIT décide d'écrire un système libre
  - ✓ GNU (Gnu's Not Unix) de noyau HURD (Hird of Unix Replacing Daemons)
- **1985 FSF** (Free Software Foundation)
  - ✓ Production de la licence GPL (General Public Licence)
  - ✓ Création de la notion de logiciel libre
- **1991 Linux** par Linus Torvalds
  - ✓ Sous licence GPL
- **1993 à 1998**
  - ✓ Apparition des premières distributions Linux :
  - ✓ Red Hat, Debian, Suse, Slackware

# Open source && libre

- La notion de logiciel libre se définit de la façon suivante :

## NOTIONS

### LOGICIEL LIBRE

- Liberté d'utiliser un logiciel quel que soit l'usage qu'on en fait
  - Liberté d'étudier le fonctionnement d'un programme et de l'adapter à votre besoin
  - Liberté de redistribuer des copies afin d'aider votre voisin
  - Liberté d'améliorer le programme et de diffuser les améliorations au public afin d'en faire bénéficier l'ensemble de la communauté
- On parle de notion de **liberté** (cf. Richard Stallmann)

### OPEN SOURCE

- S'applique aux logiciels respectant les critères l'**Open source Initiative** (*licence compatible*)
- Libre accès au code source
- Libre redistribution
- Utilisé dans les slogans pour lier la notion de logiciel libre à diffusion des sources
- **1998** Netscape Communicator devient un logiciel libre



**Libre ne veut pas dire gratuit !**

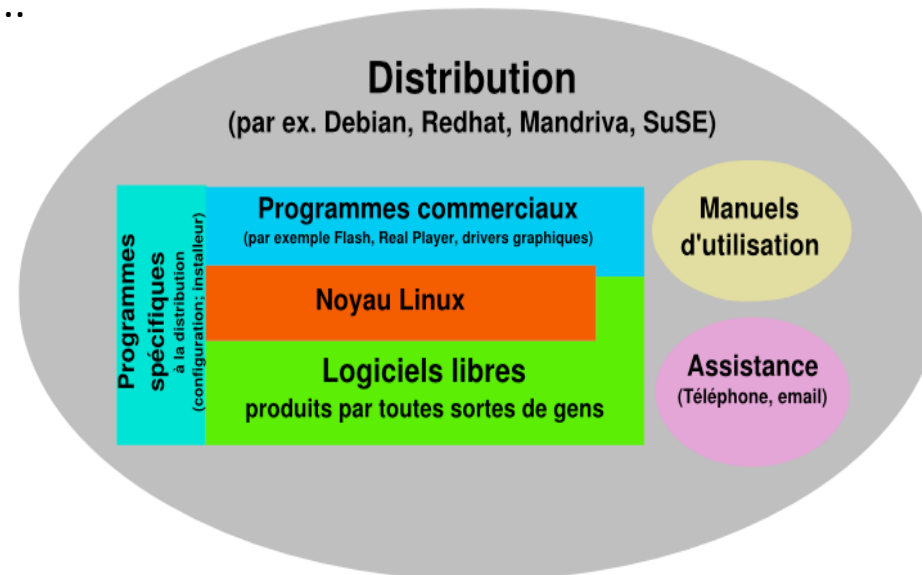
# Distribution ?

- Systèmes d'exploitation apprécié et modifié :

## DISTRIBUTION

### SYSTEMES USUELS

- Linux possède de multiples versions dérivées appelé « *distribution* »
  - ✓ Basé sur le noyau officiel : [www.kernel.org](http://www.kernel.org)
- Les variations s'opèrent :
  - ✓ Gestionnaire de fenêtrage: KDE, gnome,...
  - ✓ Liste des paquets (logiciels)
  - ✓ Outils de gestion des paquets



# Distribution ?

---

- Systèmes d'exploitation apprécié et modifié :

## DISTRIBUTION

### DEBIAN

- DEBIAN

- ✓ Créé en 1993
- ✓ Nombreux packages
- ✓ 100% open source
- ✓ Très stable en production

- Mais

- ✓ Packages souvent anciens
- ✓ Mise à jour de la distribution irrégulière et espacée
- ✓ Installation et configuration moyennement compliquées

➤ *Idéal pour un serveur*



# Distribution ?

---

- Système d'exploitation apprécié et modifié :

## DISTRIBUTION

### UBUNTU

- UBUNTU

- ✓ Créée en 2005
- ✓ Issue de Debian
- ✓ Compatibilité avec les packages Debian
- ✓ Installation très simple
- ✓ Mise à jour de la distribution tous les 6 à 8 mois
- ✓ Environnement graphique agréable

➤ *Idéal pour un serveur ou un poste client*





# Distribution ?

- Système d'exploitation apprécié et modifié :

## DISTRIBUTION

REDHAT/FEDORA

- REDHAT

- ✓ Créé en 1994
- ✓ Référence pendant près de 10 ans
- ✓ Première à proposer un système de gestion de package (rpm)
- ✓ Version commerciale pour entreprise RHEL (Red Hat Enterprise Linux)
- ✓ CentOS (Community Enterprise Operating System) version libre de RHEL
- ✓ *RHEL est idéal pour un serveur avec support commercial*

- FEDORA

- ✓ Créé en 2003
- ✓ projet communautaire Fedora
- ✓ Mise à jour distribution Fedora tous les 6 mois

➤ *Idéal pour un poste client destiné à un utilisateur averti*



# Distribution ?

---

- Système d'exploitation apprécié et modifié :

## DISTRIBUTION

### SUSE/OPENSUSE

- SUSE/OpenSUSE
  - ✓ Créée en 1993
  - ✓ Origine allemande
  - ✓ Portée par la société Novel depuis 2004
  - ✓ Version entreprise
  - ✓ Version grand public : OpenSuse
  - ✓ Nouvelle version stable tous les 6 à 8 mois
  - ✓ Utilisé pour la production d'appliance (ie: VMware)



# Distribution ?

- Système d'exploitation apprécié et modifié :

## DISTRIBUTION

VARIANTES  
EXOTIQUES/BIDOUILLEURS/HACKER

- Slackware (1993)
  - ✓ Une des plus anciennes distributions existantes
  - ✓ La plus pure
  - ✓ Une nouvelle distribution tous les ans
  - ✓ Manque de soutien
- Gentoo
  - ✓ Distribution source, tout est à compiler
- Linux from scratch
  - ✓ Tout est à configurer
  - ✓ Construction de sa propre distribution
- Pour les hackers et autres...
  - ✓ Archlinux
  - ✓ KaliLinux (ex. Backtrack)

slackware®  
linux



# Sommaire

---

- I. Fondamentaux
- II. Système linux
- III. Notions élémentaires
- IV. Utilisation du shell
- V. Essentiel
- VI. Conclusion

# Organisation des fichiers

---

- Système d'exploitation & fichiers :

## DISTRIBUTION

### ORGANISATION DES FICHIERS

- Le découpage en blocs sur un périphérique de stockage (disque dur, RAM)
- Ensemble de fichiers structurés en arbre
  - ✓ Arborescence de fichier
- Le S.E. gère une forêt d'arbres
- Structuration de l'espace de stockage
  - ✓ Système de fichier FS
  - ✓ Gérer plusieurs FS répartis sur des disques ou partitions de disques

- **Système d'exploitation & fichiers :**

disque dur  
320 Go

systèmes  
de fichiers

0 ntfs fat ext3 swap 319 999 999 999

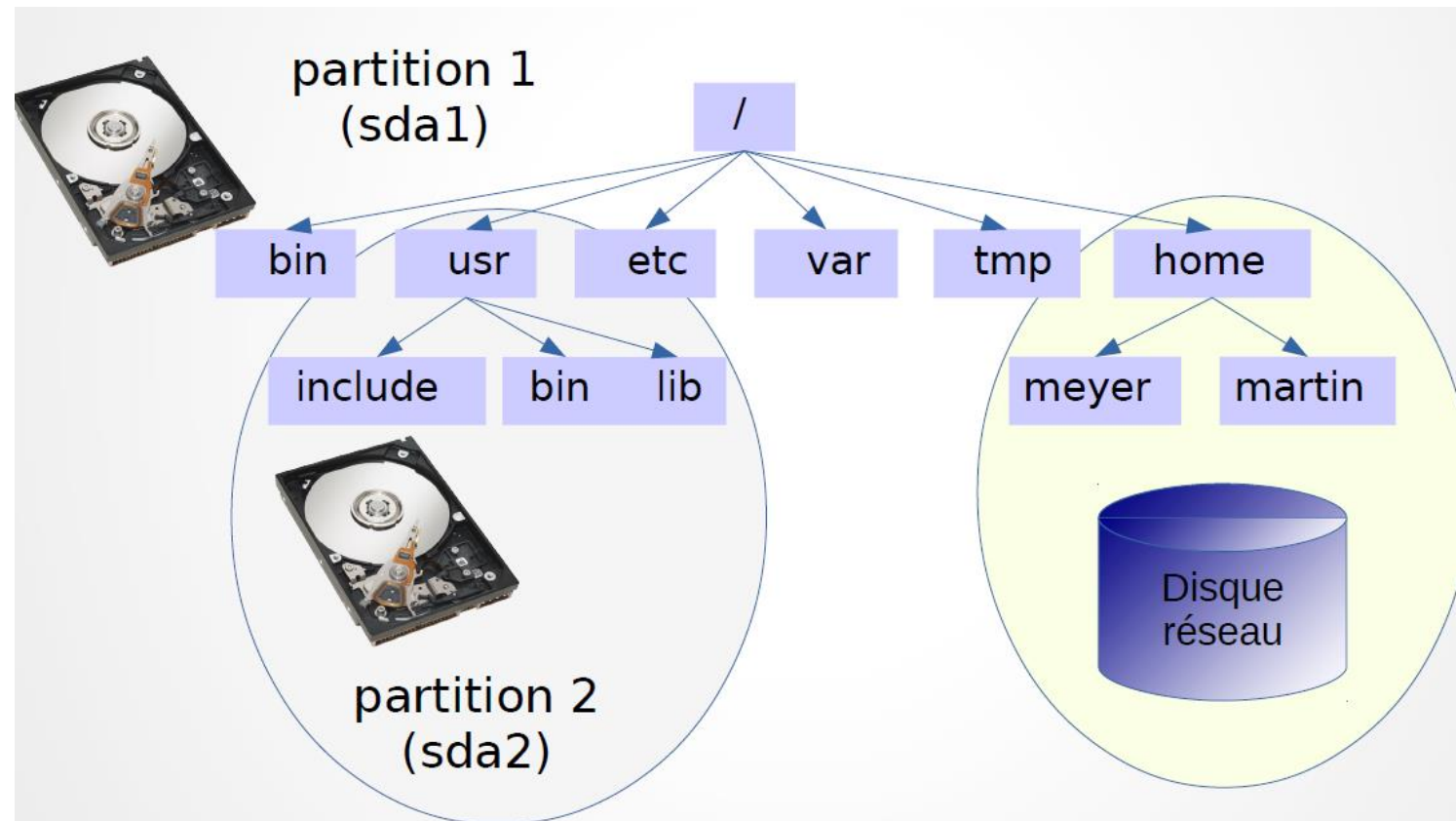
MBR partition1 partition2 partition3 partition4

# Organisation des fichiers

- Système d'exploitation & fichiers :

## DISTRIBUTION

- Corrélation éléments physiques de stockage & forêt de FS :



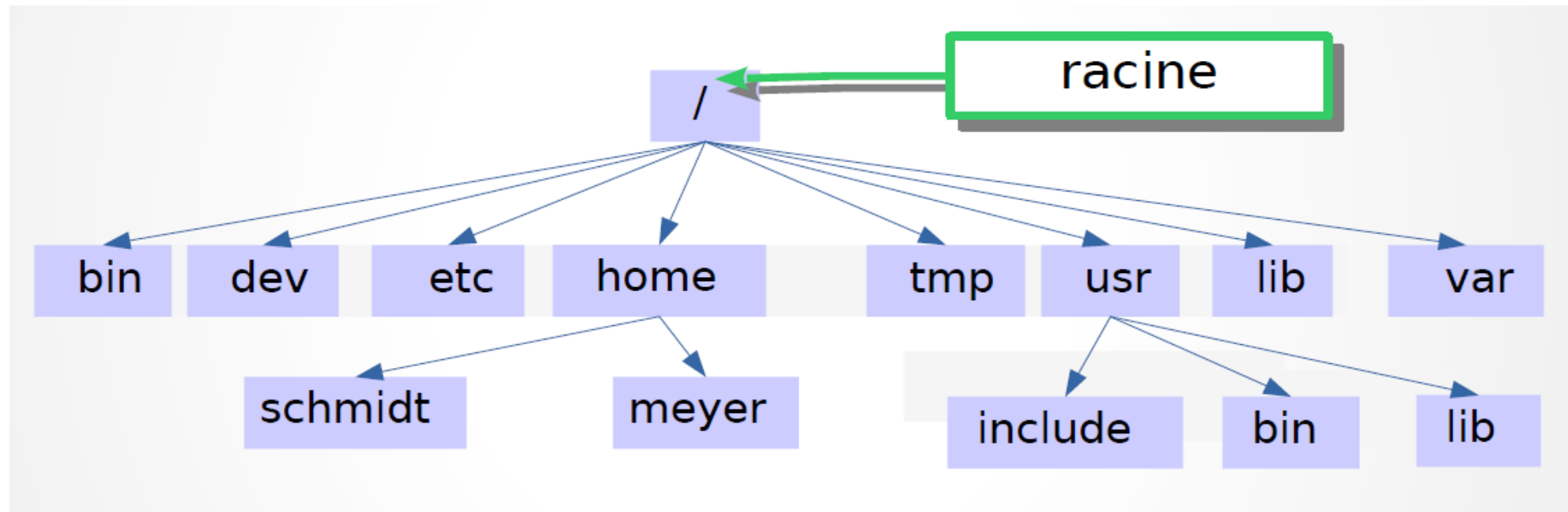
# Organisation des fichiers

- Système d'exploitation & fichiers :

## DISTRIBUTION

- Partition principale sous Linux :

✓ Respecte la *Filesystem Hierarchy Standard* (<http://www.pathname.com/fhs>)





# Organisation des fichiers

---

- Système d'exploitation & fichiers :

## DISTRIBUTION

### ORGANISATION DES FICHIERS

- Il existe de nombreux répertoires usuels dédiés à des tâches spécifiques :
  - ✓ **Fichiers de configuration** : /etc
  - ✓ **Commandes** : /bin, /sbin, /usr/bin, /usr/sbin ...
  - ✓ **Périphériques** : /dev (devices)
  - ✓ **Fichiers de log** : /var
  - ✓ **Applications** : /usr/local
  - ✓ **Librairies** : /lib, /usr/lib, ...
  - ✓ **Fichiers temporaires** : /tmp

# Organisation des fichiers

---

- Système d'exploitation & fichiers :

## DISTRIBUTION

### ORGANISATION DES FICHIERS

- Il existe de nombreux fichiers usuels dédiés à des tâches spécifiques :
  - ✓ **Démarrage** : /etc/inittab, /etc/init.d, /etc/rcX.d
  - ✓ **Utilisateurs** : /etc/passwd, /etc/shadow, /etc/group
  - ✓ **Système de fichiers** : /etc/fstab, /etc/mtab
  - ✓ **Réseaux** : /etc/network/interfaces, /etc/services, /etc/inetd.conf, /etc/resolv.conf
  - ✓ **Librairies dynamiques** : /etc/ld.so.conf.d/\*

# Organisation des fichiers

---

- Système d'exploitation & fichiers :

## DISTRIBUTION

### ORGANISATION DES FICHIERS

- Il existe de nombreux fichiers usuels dédiés à des tâches spécifiques :
  - ✓ **Fichiers de démarrage** : /boot
  - ✓ **Fichiers utilisateurs** : /home/{user}
  - ✓ **Fichiers administrateur** : /root
  - ✓ **Fichiers descriptifs du système** : /proc

# Organisation des utilisateurs

- Système d'exploitation & utilisateurs :

## UTILISATEUR

### DEFINITION UTILISATEUR

- Identification unique : **uid**
- Caractéristiques
  - ✓ Fichiers **/etc/passwd** et **/etc/shadow**
  - ✓ Authentification : login/passwd
  - ✓ Répertoire de travail : **/home/{user}**
  - ✓ Shell associé
  - ✓ Personnages humains ou non
  - ✓ Variables propres à chaque compte

- Ce sont les propriétaires des fichiers et des processus
- root (super-utilisateur)
  - ✓ Administrateur du système
  - ✓ Tous les droits partout
- Les autres
  - ✓ Tous les droits dans leur répertoire personnel
    - ✓ Home directory (ex: /home/dupont)
  - ✓ Droits restreints ailleurs
  - ✓ Nécessité pour l'utilisation courante du système

# Organisation des utilisateurs

- Système d'exploitation & utilisateurs :

## UTILISATEUR

### OPERATION SUR UTILISATEUR

- Ajout/gestion/suppression d'un utilisateur
  - ✓ useradd / usermod / userdel
- Gestion des mots de passe
  - ✓ passwd
- Ajout/gestion/suppression des groupes
  - ✓ groupadd / groupmod / groupdel
- Contrôle de cohérence
  - ✓ pwck (vérification : /etc/passwd et /etc/shadow)
  - ✓ grpck (vérification : /etc/group)
- Chaque utilisateur a un UID unique
  - ✓ Défini dans **/etc/passwd**
  - ✓ Commande id permet d'obtenir l'information
- Chaque utilisateur possède un groupe
  - ✓ Chaque groupe a un GID unique
  - ✓ Défini dans **/etc/group**
- Chaque fichier possède un UID et un GID

# Organisation des utilisateurs

- Système d'exploitation & utilisateurs :

## UTILISATEUR

### DROITS SUR FICHIERS

- Pour chaque fichier
  - ✓ 3 types d'utilisateurs avec des droits distincts
  - ✓ le propriétaire, le groupe, les autres
- Pour chaque type d'utilisateur
  - ✓ 3 droits
  - ✓ read (r), write (w), execute (x)

```
[localhost:~/pictures] gaston% ls -l
total 26352
-rw-r--r--  1 gaston  wheel  1067510 Oct  2 14:19 DSCN0607.JPG
-rw-r--r--  1 gaston  wheel  1098203 Oct  2 14:21 DSCN0608.JPG
```

1	2	3	4	5	6	7
-rw-r--r--	1	gaston	wheel	1067510	Oct 2 14:19	DSCN0607.JPG
-rw-r--r--	1	gaston	wheel	1098203	Oct 2 14:21	DSCN0608.JPG

# Organisation des utilisateurs

- Système d'exploitation & utilisateurs :

## UTILISATEUR

### DROITS SUR FICHIERS

Droits	Signification
Général	
r	Lecture
w	Écriture
x	Exécution
Fichier normal	
r	Contenu fichier peut être lu, visualisé, recopié
w	Contenu du fichier peut être modifié
x	Le fichier peut être exécuté si c'est un binaire ou un script
Répertoire	
r	Contenu du répertoire peut être listé
w	Contenu du répertoire peut être modifié. Fichiers créés, modifiés, <b>supprimés</b>
x	Le répertoire peut être accédé, traversé (sinon verrouillé)

# Processus

---

- Système d'exploitation & processus:

## PROCESSUS

- Chaque processus est lancé par un processus père
  - ✓ Premier processus : /sbin/init (pid : 1)
- Un processus a également un propriétaire
- Arborescence de processus (*ps tree*)
- Répertoire des processus à lancer au démarrage : /etc/rcX.d/
- Les processus résident en mémoire centrale
- Les processus peuvent prendre différents états au cours de leurs exécutions



# Programmes

---

- Système d'exploitation & programmes :

## PROGRAMMES

### INSTALLATION MANUELLE

- Récupération des sources (.tgz)
- Localiser dans /usr/local
- Compilation
  - ✓ ./configure [options]
  - ✓ make
  - ✓ make install

### INSTALLATION AUTOMATIQUE

- Par le réseau ou via un support alternatif (cdrom, clef usb)
- Gestionnaire de packets
  - ✓ Debian : dpk, apt, aptitude
  - ✓ Fedora : rpm, yum
  - ✓ Mandriva : rpm, urpmi
  - ✓ OpenSuse : rpm, yast
  - ✓ Archlinux : pacman, yaourt

# Programmes

---

- Système d'exploitation & programmes :

## PACKAGE

### COMPOSITION

- ✓ Programme (exécutable)
- ✓ Documentation
- ✓ Scripts de pré-installation
- ✓ Scripts de post-installation
- ✓ Fichiers de configuration
- ✓ Liste des dépendances

Extension **.deb** pour Debian

Extension **.rpm** pour Red Hat, OpenSuse, Fedora etc.

- Conversion possible grâce à l'outil **alien**

# Programmes

---

- Système d'exploitation & programmes :

## PACKAGE

### UTILISATION

- **dpkg**

- ✓ /var/lib/dpkg/status
  - ✓ intégralité des packages connus par dpkg et leur état
- Options
  - ✓ -i installation d'un package
  - ✓ -r suppression d'un package
  - ✓ -P suppression d'un package avec les fichiers de configuration
  - ✓ -l liste les paquets debian connus du système
  - ✓ -S nom\_fichier permet de retrouver le package d'origine du fichier

# Programmes

---

- Système d'exploitation & programmes :

## PACKAGE

### UTILISATION

- **APT (aptitude packaging tool) ou aptitude**

- ✓ Intégralité des packages connus par dpkg et leur état dpkg ou rpm ne gèrent pas les dépendances
- ✓ Prend en charge des dépôts situés sur des supports divers : cdrom, répertoire local, internet
- ✓ Fichier de configuration des dépôts : **/etc/apt/sources.list**
  - ✓ deb <http://ftp.fr.debian.org/debian> squeeze main contrib non-free

# Services

---

- Système d'exploitation & services :

## SERVICE / DAEMON

### UTILISATION

- **Il s'agit d'une application en tâche de fond**
- Exécuté par un processus
  - ✓ IE : réseau : NFS, httpd, samba
  - ✓ IE : système : cron, syslogd
- Mise à jour des paquets déjà installés
  - ✓ Init.d : au démarrage, manuellement (sysV, systemd)
  - ✓ Inetd : pour des problématiques réseau généralement

# Services

- Système d'exploitation & services :

## SERVICE / DAEMON

### UTILISATION

- **INIT.D**

- ✓ Scripts shell au sein du répertoire :
  - /etc/init.d/ (debian)
  - /etc/rc.d/ (centos)
  - Au démarrage par le processus init
- ✓ Usage manuel via les commandes :
  - ✓ start, stop, reload, status

- **INET.D**

- ✓ Super-daemon :
  - /etc/inetd.conf
  - Réservé aux services réseaux
  - Lancer un service à la demande
- ✓ Limite la consommation des ressources
- ✓ **xinetd** version plus sécurisée
- ✓ Exemple de configuration :

```
cat /etc/inetd.conf :  
ftp stream tcp nowait root usr/libexec/ftpd ftpd -l
```

# Services

---

- Système d'exploitation & services :

## SERVICE / DAEMON

### UTILISATION

- **Redémarrage**

- ✓ Potentiellement nécessaire après un changement de configuration
- ✓ Plusieurs méthodes :
  - `/etc/init.d/{nom_service} restart` (ou `reload` si disponible)
  - `service {nom_service} restart` (ou `reload` si disponible)
  - `systemctl {restart|reload|stop|start} {nom_service}`
  - Commande `kill -9 {PID}`

- Il est toujours possible et/ou indispensable d'éviter le reboot !

# Sommaire

---

- I. Fondamentaux
  - II. Système linux
  - III. Notions élémentaires
  - IV. Utilisation du shell
  - V. Essentiel
  - VI. Conclusion
-



# Utilisation

---

- Système d'exploitation & utilisation :

## EXPLOITATION

### UTILISATION

- **Commandes**

- ✓ Programmes exécutables
- ✓ Permet aux utilisateurs de manipuler les fichiers et les processus
- ✓ Catégorisé en deux sortes :
  - Interne : interprétées par le shell
  - Externe : application tierces

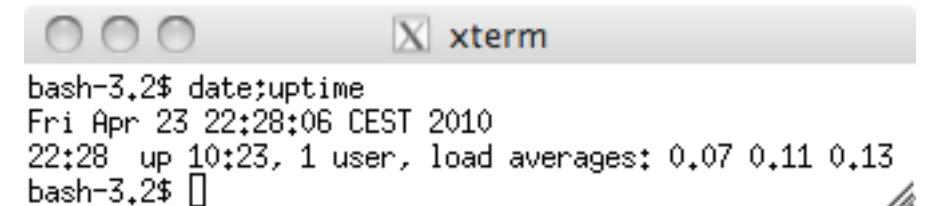
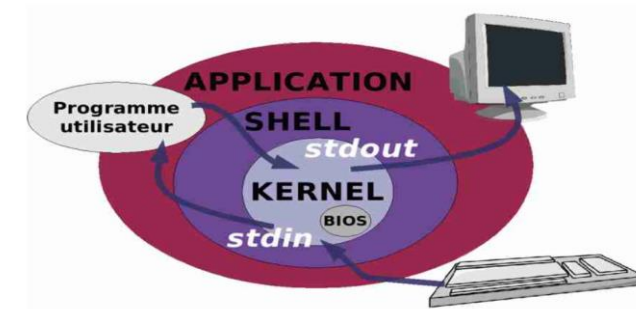
# Utilisation

- Système d'exploitation & shell :

## EXPLOITATION

### UTILISATION

- **Interpréteur de commandes**
- « Couteau suisse » pour utilisateur
- Partie la plus externe du SE
  - ✓ Interface utilisateur
- Il existe différents types d'interpréteur :
  - ✓ bash, csh, ksh
- L'invite de commande :
  - ✓ Syntaxe : commande [OPTIONS] [ARGUMENTS]
    - ✓ IE : rm -f test.txt
    - ✓ Séparation de commande par un « ; »



# Utilisation

---

- Aide et documentation shell :

## EXPLOITATION

### UTILISATION

- Pour les commandes externes au shell : --help
  - ✓ IE : apt-get --help
- Pour les commandes internes au shell : help COMMANDE
  - ✓ IE : help passwd
- Présence d'un manuel : man COMMANDE
  - ✓ Recherche par correspondance / section / commande spécifique
  - ✓ Présent au sein du répertoire : **/usr/share/doc**

# Utilisation

- Aide et documentation shell :

## EXPLOITATION

### UTILISATION

Section	Contenu
1	Instructions exécutables ou commande du shell
2	Appels système (API du noyau)
3	Appel des bibliothèques (fonctions C ...)
4	Fichiers spéciaux (contenu de /dev comme sd, hd, ..)
5	Format des fichiers (/etc/passwd, /etc/hosts, etc)
6	Les jeux, économiseurs d'écrans, gadgets,...
7	Divers, commandes non standard
8	Commande d'administration du système Linux
9	Sous-programmes du noyau (souvent vide)

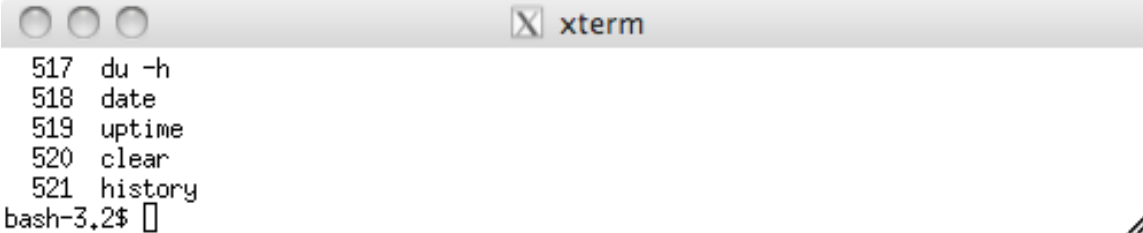
# Utilisation

- Historique des commandes :

## EXPLOITATION

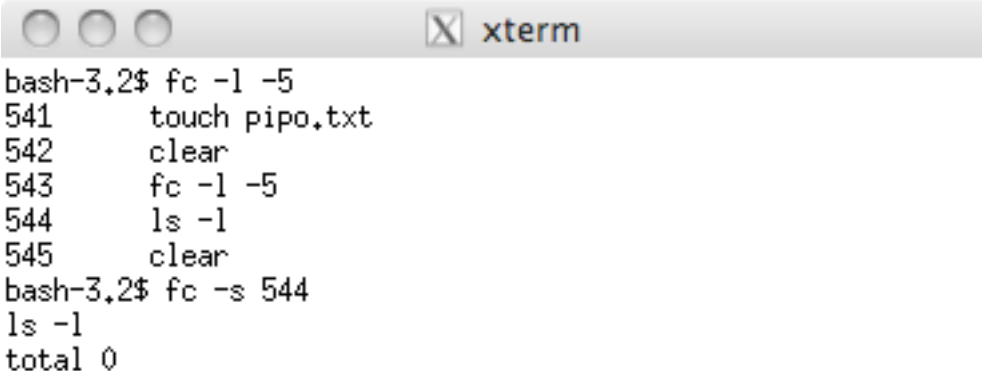
### UTILISATION

- Commande **history** : *(lié au fichier .history du shell)*



```
517 du -h
518 date
519 uptime
520 clear
521 history
bash-3.2$
```

- Commande **fc** :



```
bash-3.2$ fc -l -5
541      touch pipo.txt
542      clear
543      fc -l -5
544      ls -l
545      clear
bash-3.2$ fc -s 544
ls -l
total 0
```

# Utilisation

---

- Valeur de retour d'une commande :

## EXPLOITATION

### UTILISATION

- Variable : \$?
- 0 si réussi ; 1 ou 2 si échec

```
bash-3.2$ echo $?  
0  
bash-3.2$ rmdir rep-test  
rmdir: rep-test: No such file or directory  
bash-3.2$ echo $?  
1  
bash-3.2$
```

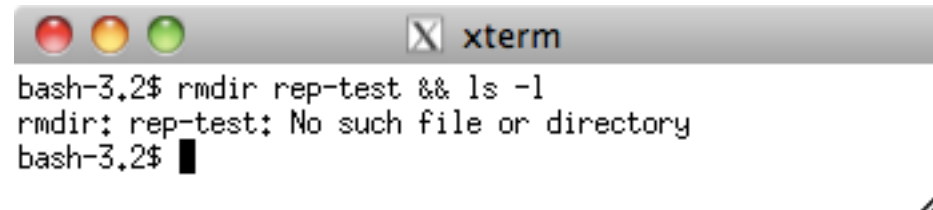
# Utilisation

- Exécution conditionnelle des commandes **&&** et **||** :

## EXPLOITATION

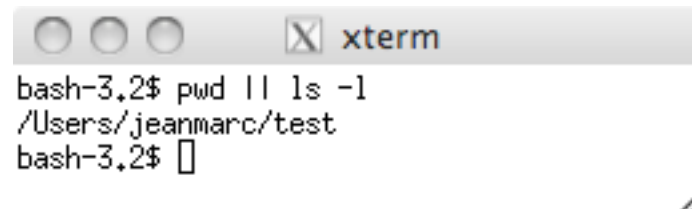
### UTILISATION

- **commande1 && commande2**
  - ✓ commande2 seulement si commande1 retourne 0



```
bash-3.2$ rmdir rep-test && ls -l
rmdir: rep-test: No such file or directory
bash-3.2$
```

- **commande1 || commande2**
  - ✓ commande2 seulement si commande1 ne retourne pas 0



```
bash-3.2$ pwd || ls -l
/Users/jeanmarc/test
bash-3.2$
```

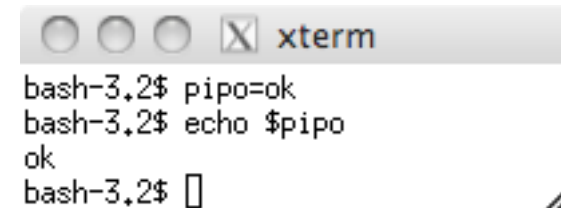
# Utilisation

- Variables d'environnement : (3 types : utilisateur, système, spéciale)

## EXPLOITATION

### UTILISATION

- Nomenclature
  - ✓ Premier caractère pas un chiffre
  - ✓ Variables utilisateur en minuscule, variable système en majuscule
  - ✓ Taille du nom illimité
- Affectation
  - ✓ `nom_variable = valeur`
- Accès
  - ✓ `$nom_variable`
- Affichage
  - ✓ `echo $nom_variable`



```
bash-3.2$ pipo=ok
bash-3.2$ echo $pipo
ok
bash-3.2$
```



# Utilisation

- Variables d'environnement : (3 types : utilisateur, système, spéciale)

## EXPLOITATION

### UTILISATION

- Affichage liste des variables définies

- ✓ env
- ✓ set

- Suppression

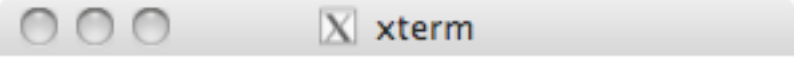
- ✓ unset nom\_variable

- Protection

- ✓ readonly
- ✓ Peut plus modifier
- ✓ Peut pas supprimer (quitter le shell)

- Exposer la variable pour d'autres shells et scripts

- ✓ Par défaut accessible que dans le shell où elle est définie
- ✓ export nom\_variable



```
bash-3.2$ echo $pipo
ok
bash-3.2$ unset pipo
bash-3.2$ echo $pipo

bash-3.2$ pipo=nok
bash-3.2$ readonly pipo
bash-3.2$ unset pipo
bash: unset: pipo: cannot unset: readonly variable
bash-3.2$
```

# Utilisation

- Variables d'environnement : (3 types : utilisateur, système, spéciale)

## EXPLOITATION

### SYSTEME

Variable	Contenu
<b>HOME</b>	Répertoire par défaut de l'utilisateur
<b>PATH</b>	Liste des répertoires, séparés par des : ou le shell va chercher les commandes
<b>P\$1</b>	Chaîne représentant le prompt standard affiché à l'écran
<b>SHELL</b>	Chemin complet du shell en cours d'exécution
<b>LANG</b>	Définition de la langue à utiliser ainsi que du jeu de caractère
<b>USER</b>	Nom de l'utilisateur en cours
<b>LOGNAME</b>	Nom du login utilisé lors de la connexion
<b>PWD</b>	Chemin d'accès courant
<b>RANDOM</b>	Génère et contient un nombre aléatoire entre 0 et 32767

### SPECIALE

Variable	Contenu
<b>\$?</b>	Code retour de la dernière commande exécutée
<b>\$\$</b>	PID du shell actif
<b>\$!</b>	PID du dernier processus lancé en arrière-plan
<b>\$-</b>	Les options du shell

# Utilisation

---

- Raccourci de commandes :

## EXPLOITATION

### UTILISATION

- Commande **alias** : *(souvent définit dans le fichier ~/.bashrc)*

```
muller@aricept:~$ alias
alias ls='ls --color=auto'
muller@aricept:~$ alias deltree='rm -rf'
muller@aricept:~$ alias
alias deltree='rm -rf'
alias ls='ls --color=auto'
muller@aricept:~$
```

# Utilisation

---

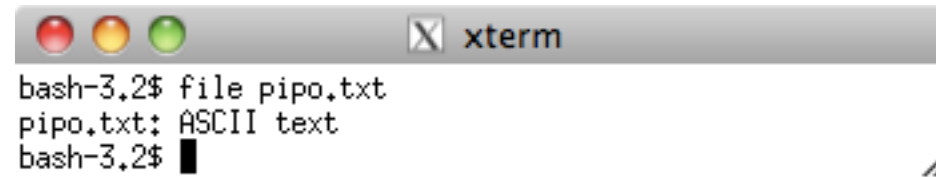
- Typologie d'un fichier :

## EXPLOITATION

### UTILISATION

- Commande *file* :

- ✓ Fichiers
- ✓ Répertoires
- ✓ Fichiers spéciaux
  - ✓ /dev



```
bash-3.2$ file pipo.txt
pipo.txt: ASCII text
bash-3.2$
```

# Utilisation

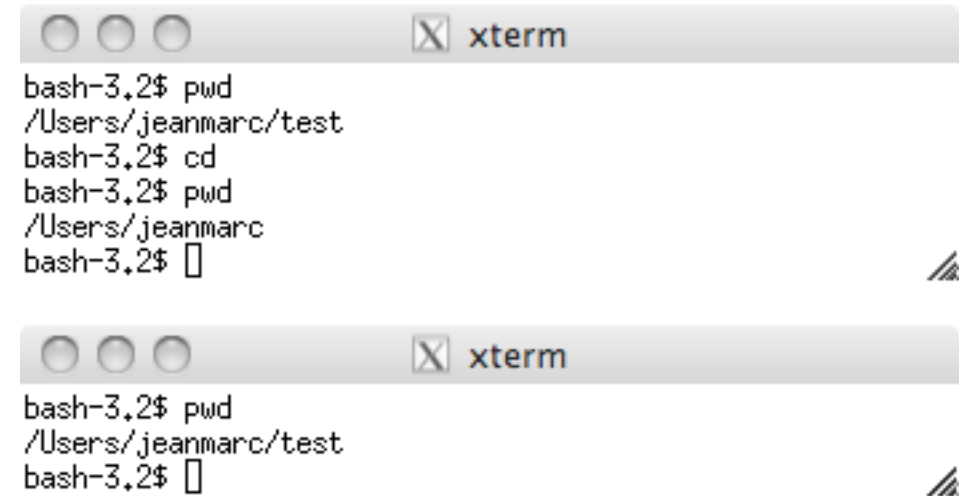
- Navigation et consultation des fichiers:

## EXPLOITATION

### UTILISATION

- Commande ***cd*** :
  - ✓ Permet de changer de répertoire
- Commande ***pwd*** :
  - ✓ Indique le chemin absolu du répertoire courant

Variable	Contenu
.	Répertoire courant
..	Répertoire parent
~	Répertoire personnel « home »
/	Répertoire racine /



```
bash-3.2$ pwd
/Users/jeanmarc/test
bash-3.2$ cd
bash-3.2$ pwd
/Users/jeanmarc
bash-3.2$
```

```
bash-3.2$ pwd
/Users/jeanmarc/test
bash-3.2$
```

# Utilisation

- Navigation et consultation des fichiers:

## EXPLOITATION

### UTILISATION

- Commande **ls** :

- ✓ Permet de lister les fichiers d'un répertoire
- ✓ 1 : type de fichier
  - ✓ - : ordinaire, d : répertoire, l : lien symbolique
- ✓ 2 : nombre de liens
- ✓ 3 : user
- ✓ 4 : group
- ✓ 5 : taille en octets
- ✓ 6 : date de dernière modification
- ✓ 7 : nom du fichier

```
[localhost:~/pictures] gaston% ls -l
total 26352
-rw-r--r--  1 gaston  wheel  1067510 Oct  2 14:19 DSCN0607.JPG
-rw-r--r--  1 gaston  wheel  1098203 Oct  2 14:21 DSCN0608.JPG
```

1	2	3	4	5	6	7
-rw-r--r--	1	gaston	wheel	1067510	Oct 2 14:19	DSCN0607.JPG
-rw-r--r--	1	gaston	wheel	1098203	Oct 2 14:21	DSCN0608.JPG

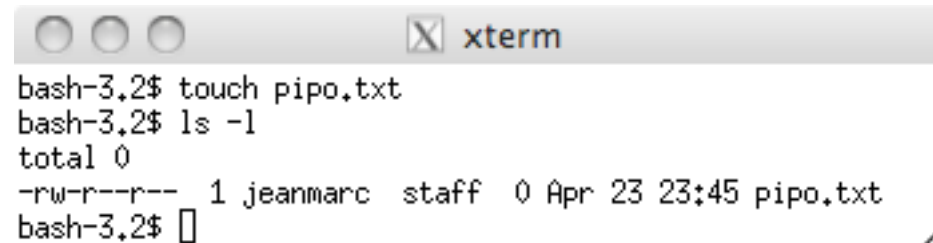
# Utilisation

- Création de répertoires & fichiers:

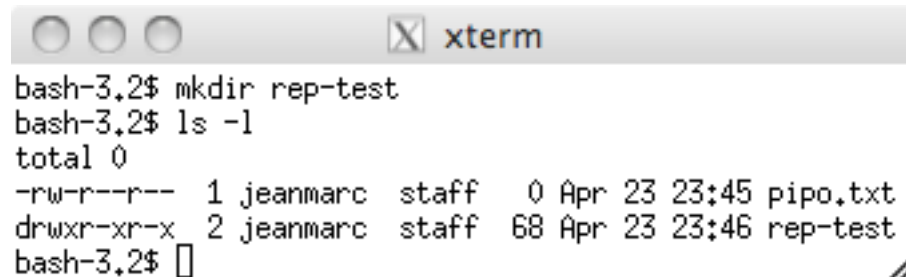
## EXPLOITATION

### UTILISATION

- Commande ***touch*** :
  - ✓ Permet de créer un fichier
- Commande ***mkdir*** :
  - ✓ Permet de créer un répertoire



```
bash-3.2$ touch pipo.txt
bash-3.2$ ls -l
total 0
-rw-r--r-- 1 jeanmarc  staff  0 Apr 23 23:45 pipo.txt
bash-3.2$
```



```
bash-3.2$ mkdir rep-test
bash-3.2$ ls -l
total 0
-rw-r--r-- 1 jeanmarc  staff  0 Apr 23 23:45 pipo.txt
drwxr-xr-x 2 jeanmarc  staff 68 Apr 23 23:46 rep-test
bash-3.2$
```

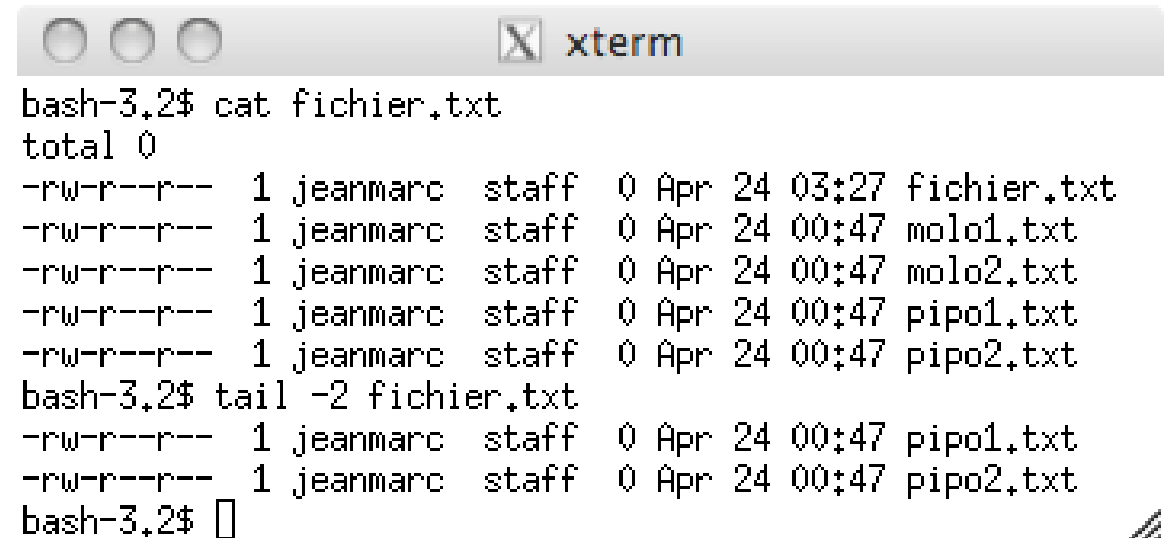
# Utilisation

- Consultation de fichiers:

## EXPLOITATION

### UTILISATION

- Commandes ***pg, more, less*** :
  - ✓ Consultation en mode page
- Commande ***cat*** :
  - ✓ Consultation en mode bloc
- Commandes ***head tail***:
  - ✓ Consultation par troncature (début/fin)



```
bash-3.2$ cat fichier.txt
total 0
-rw-r--r--  1 jeanmarc  staff   0 Apr 24 03:27 fichier.txt
-rw-r--r--  1 jeanmarc  staff   0 Apr 24 00:47 molo1.txt
-rw-r--r--  1 jeanmarc  staff   0 Apr 24 00:47 molo2.txt
-rw-r--r--  1 jeanmarc  staff   0 Apr 24 00:47 pipo1.txt
-rw-r--r--  1 jeanmarc  staff   0 Apr 24 00:47 pipo2.txt
bash-3.2$ tail -2 fichier.txt
-rw-r--r--  1 jeanmarc  staff   0 Apr 24 00:47 pipo1.txt
-rw-r--r--  1 jeanmarc  staff   0 Apr 24 00:47 pipo2.txt
bash-3.2$
```



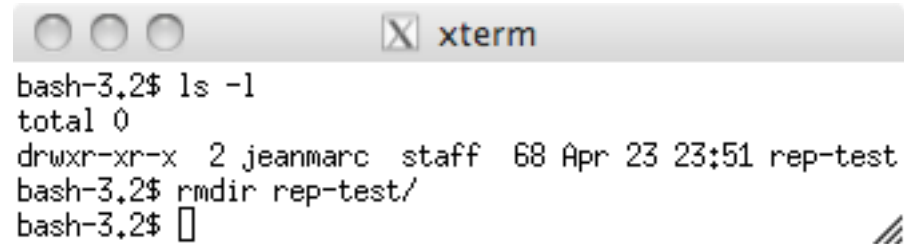
# Utilisation

- Suppression de répertoires & fichiers:

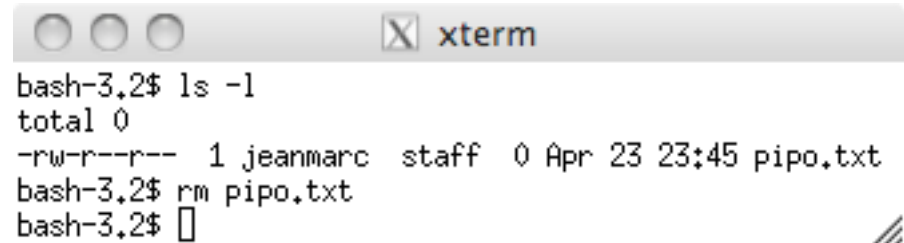
## EXPLOITATION

### UTILISATION

- Commande ***rm*** :
  - ✓ Permet de supprimer un fichier
  - ✓ option ***-rf***
- Commande ***rmdir*** :
  - ✓ Permet de supprimer un répertoire vide



```
bash-3.2$ ls -l
total 0
drwxr-xr-x  2 jeanmarc  staff  68 Apr 23 23:51 rep-test
bash-3.2$ rmdir rep-test/
bash-3.2$
```



```
bash-3.2$ ls -l
total 0
-rw-r--r--  1 jeanmarc  staff   0 Apr 23 23:45 pipo.txt
bash-3.2$ rm pipo.txt
bash-3.2$
```

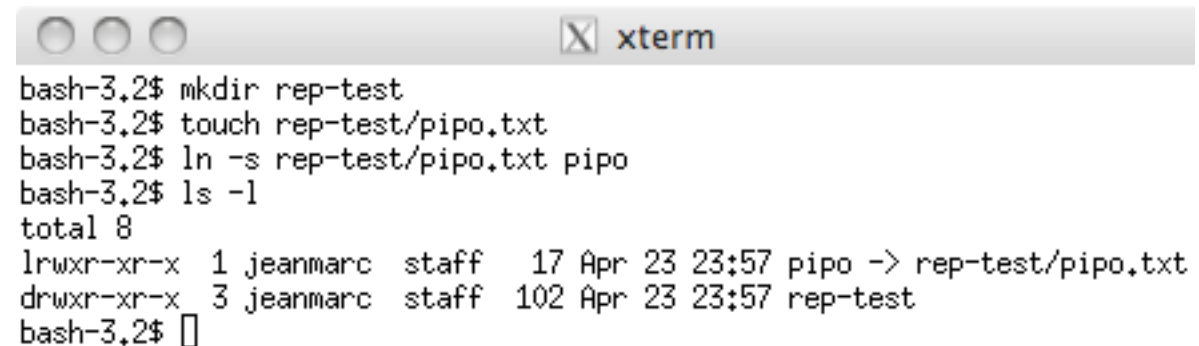
# Utilisation

- Liens symboliques sur répertoires & fichiers:

## EXPLOITATION

### UTILISATION

- Commande ***ln*** :
  - ✓ Permet de réaliser un « raccourci » sur un fichier ou dossier
    - ✓ *option -s*
  - ✓ *Répercution sur le fichier « pointé »*
  - ✓ *Suppression « safe »*



```
bash-3.2$ mkdir rep-test
bash-3.2$ touch rep-test/pipo.txt
bash-3.2$ ln -s rep-test/pipo.txt pipo
bash-3.2$ ls -l
total 8
lrwxr-xr-x 1 jeanmarc staff 17 Apr 23 23:57 pipo -> rep-test/pipo.txt
drwxr-xr-x 3 jeanmarc staff 102 Apr 23 23:57 rep-test
bash-3.2$
```

# Utilisation

---

- Droits & accès aux fichiers :

## EXPLOITATION

### UTILISATION

- Chaque utilisateur a un **UID** (User Identification) unique
  - ✓ Précisé au sein du fichier **/etc/passwd**
  - ✓ Commande **id** permet d'obtenir l'information
- Chaque utilisateur rattaché à un groupe
  - ✓ Chaque groupe a un **GID** (Group Identification) unique
  - ✓ Précisé au sein du fichier **/etc/group**
- À chaque fichier est associé un UID et un GID
  - ✓ Si UID utilisateur identique alors celui-ci propriétaire fichier
  - ✓ Sinon si GID utilisateur identique alors dans groupe fichier
  - ✓ Sinon utilisateur fait partie des autres par rapport au fichier

# Utilisation

- Droits & accès aux fichiers :

## EXPLOITATION

### UTILISATION

Droits	Signification
<b>Général</b>	
r	Lecture
w	Écriture
x	Exécution
<b>Fichier normal</b>	
r	Contenu fichier peut être lu, visualisé, recopié
w	Contenu du fichier peut être modifié (suppression pas forcément liée à ce droit)
x	Le fichier peut être exécuté si c'est un binaire ou un script
<b>Répertoire</b>	
r	Contenu du répertoire peut être listé
w	Contenu du répertoire peut être modifié. Fichiers créés, modifiés, <b>supprimés</b>
x	Le répertoire peut être accédé, traversé (sinon verrouillé)

# Utilisation

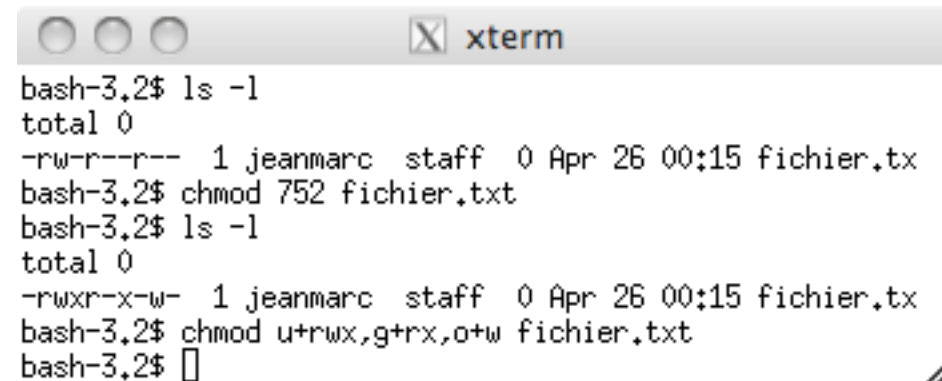
- Droits & accès aux fichiers :

## EXPLOITATION

### UTILISATION

- Commande **chmod** : (modification des droits)

Propriétaire			Groupe			Autres		
r	w	x	r	w	x	r	w	x
400	200	100	40	20	10	4	2	1



```

bash-3.2$ ls -l
total 0
-rw-r--r-- 1 jeanmarc staff 0 Apr 26 00:15 fichier.tx
bash-3.2$ chmod 752 fichier.txt
bash-3.2$ ls -l
total 0
-rwxr-x-w- 1 jeanmarc staff 0 Apr 26 00:15 fichier.tx
bash-3.2$ chmod u+rwx,g+rx,o+w fichier.txt
bash-3.2$
  
```

# Utilisation

---

- Droits & accès aux fichiers :

## EXPLOITATION

- UTILISATION ■ Commande ***umask*** : (modification des droits appliqués)
  - ✓ La valeur du masque est retirée des droits par défaut des fichiers et répertoires
    - ✓ Fichier **666** (rw-rw-rw-)
    - ✓ Répertoire **777** (rwxrwxrwx)
    - ✓ Puis masque est appliqué

# Utilisation

---

- Droits & accès aux fichiers :

## EXPLOITATION

### UTILISATION

- Commande ***chown*** : (modification du propriétaire)
  - ✓ La valeur du propriétaire du fichier est changé
- Commande ***chgrp*** : (modification du groupe)
  - ✓ La valeur du groupe du fichier est changé
- ✓ Seul l'utilisateur **root** est capable de changer le propriétaire d'un fichier
- ✓ Utilisateur peut changer le groupe s'il appartient au groupe destination

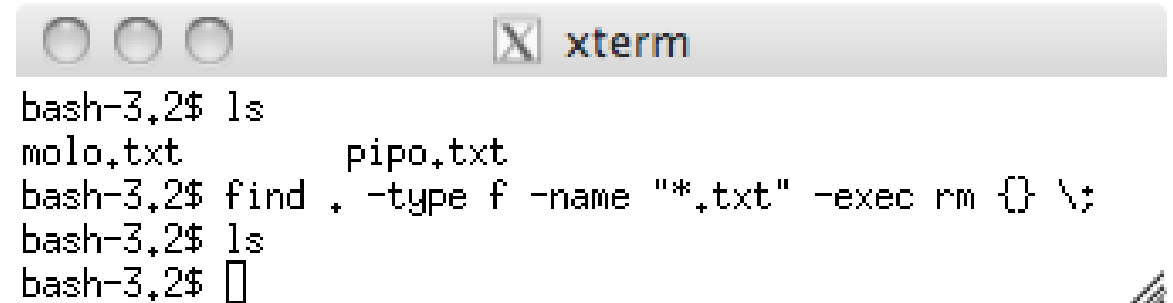
# Utilisation

- Recherche de fichiers :

## EXPLOITATION

### UTILISATION

- Commande ***find*** :
  - ✓ + [CHEMIN] [CRITERES] [COMMANDE]
  - ✓ Critères : -name, -type, -user, -group, -size, -atime, -mtime, -perm
  - ✓ Commande : -print, -ls, -exec
- ✓ La recherche est menée de façon récursive



```
bash-3.2$ ls
molo.txt      pipo.txt
bash-3.2$ find . -type f -name "*.txt" -exec rm {} \;
bash-3.2$ ls
bash-3.2$
```



# Utilisation

---

- Redirection & flux :

## EXPLOITATION

- UTILISATION ■ Les entrées/sorties passe par des **flux** :
  - ✓ 0 : entrée standard
  - ✓ 1 : sortie standard
  - ✓ 2 : sortie d'erreur
- Il est possible de dévier un canal afin de réaliser des redirections :
  - ✓ Affichage => un fichier (**1>** nom\_fichier)
  - ✓ Messages d'erreur => un autre fichier (**2>** nom\_fichier)
  - ✓ Substituer la saisie clavier par le contenu d'un fichier (**<** nom\_fichier)
  - ✓ Rediriger le résultat d'une commande vers l'entrée d'une autre ( **|** )
  - ✓ Possibilité de dévier un canal ( **x>&y** )

# Utilisation

## ■ Redirection & flux :

### EXPLOITATION

#### UTILISATION

```

bash-3.2$ ls
molo1.txt      molo2.txt      pipo1.txt      pipo2.txt
bash-3.2$ ls -l > liste.txt
bash-3.2$ cat liste.txt
total 0
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 01:31 liste.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 molo1.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 molo2.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 pipo1.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 pipo2.txt
bash-3.2$ 

```

```

bash-3.2$ cat liste.txt
total 0
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 01:31 liste.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 molo1.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 molo2.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 pipo1.txt
-rw-r--r--  1 jeanmarc  staff  0 Apr 24 00:47 pipo2.txt
bash-3.2$ wc -l < liste.txt
6
bash-3.2$ 

```

```

bash-3.2$ rmdir rep1
rmdir: rep1: No such file or directory
bash-3.2$ rmdir rep1 2>fichier-erreur
bash-3.2$ cat fichier-erreur
rmdir: rep1: No such file or directory
bash-3.2$ 

```

```

bash-3.2$ ls >> resultat.txt;rmdir rep >> resultat.txt 2>&1
bash-3.2$ cat resultat.txt
molo1.txt
molo2.txt
pipo1.txt
pipo2.txt
resultat.txt
rmdir: rep: No such file or directory
bash-3.2$ 

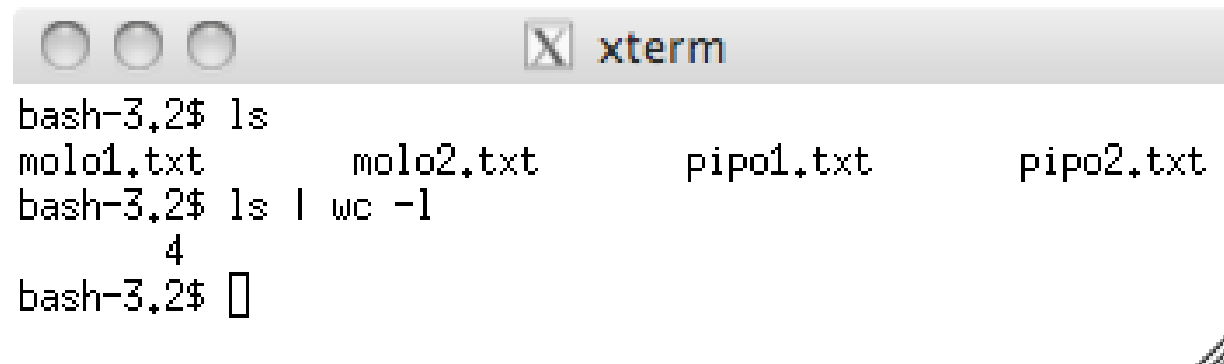
```

# Utilisation

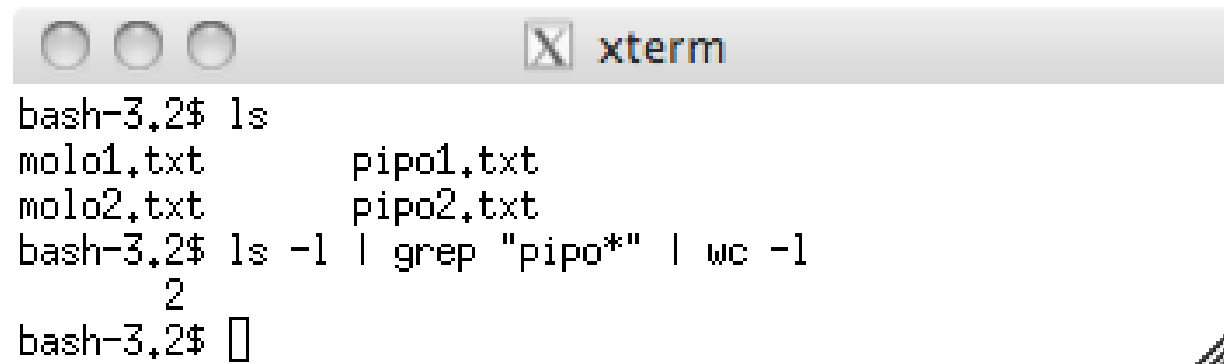
- Redirection & flux :

## EXPLOITATION

### UTILISATION



```
bash-3.2$ ls
molo1.txt      molo2.txt      pipo1.txt      pipo2.txt
bash-3.2$ ls | wc -l
4
bash-3.2$
```



```
bash-3.2$ ls
molo1.txt      pipo1.txt
molo2.txt      pipo2.txt
bash-3.2$ ls -l | grep "pipo*" | wc -l
2
bash-3.2$
```

# Utilisation

---

- Commandes & processus :

## EXPLOITATION

### UTILISATION

- A chaque terminal sont associés des processus :
  - ✓ Le shell lui-même est un processus
  - ✓ Une commande interne s'exécute dans le cadre du processus du shell
  - ✓ Une commande externe s'exécute sous la forme d'un processus fils
    - ✓ Le shell crée un processus enfant
  - ✓ **&** - permet de lancer un processus en tâche de fond
- A chaque processus sont associés :
  - ✓ Des ressources (CPU, RAM etc.)
  - ✓ Des descripteurs
  - ✓ Des comportements vis à vis de signaux

# Utilisation

- Explorer les processus :

## EXPLOITATION

### UTILISATION

- Commande **ps** :
  - ✓ -f pour plus d'information
  - ✓ -e info sur tous les processus en cours
  - ✓ -u info sur les processus d'un utilisateur donné
  - ✓ -g info sur les processus d'un groupe donné
  - ✓ -l plus d'informations techniques

```
bash-3.2$ ps -a -f
  UID   PID  PPID  C   STIME TTY          TIME CMD
    502  1706  1551  0   0:00.80 ttys000    0:01.10 bash
      0   3027   1706  0   0:00.00 ttys000    0:00.00 ps -a -f
bash-3.2$
```

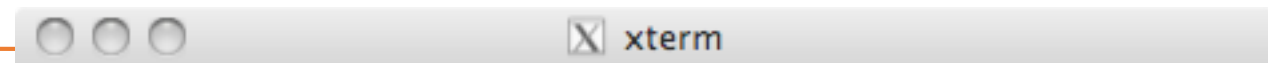
# Utilisation

- Consommation des processus :

## EXPLOITATION

### UTILISATION

- Commande **time** :
  - ✓ Affiche l'usage CPU d'un processus
  - ✓ **Real** : durée totale d'exécution
  - ✓ **User** : temps CPU pour l'exécution
  - ✓ **System** : temps CPU pour exécuter les commandes de l'OS
- Commande **top** :
  - ✓ Affiche l'usage CPU/RAM d'un processus



```
bash-3.2$ time ls -l
total 16
-rw-r--r-- 1 jeanmarc staff 40 Apr 24 03:54 fichier.txt
-rw-r--r-- 1 jeanmarc staff 40 Apr 24 03:54 fichier2.txt
-rw-r--r-- 1 jeanmarc staff 0 Apr 24 00:47 molo1.txt
-rw-r--r-- 1 jeanmarc staff 0 Apr 24 00:47 molo2.txt
-rw-r--r-- 1 jeanmarc staff 0 Apr 24 00:47 pipo1.txt
-rw-r--r-- 1 jeanmarc staff 0 Apr 24 00:47 pipo2.txt

real    0m0.010s
user    0m0.001s
sys     0m0.004s
bash-3.2$
```

```
Processes: 76 total, 2 running, 74 sleeping, 284 threads 12:14:53
Load Avg: 0.38, 0.20, 0.17 CPU usage: 7.54% user, 7.7% sys, 85.37% idle
SharedLibs: 8260K resident, 9276K data, 0B linkedit.
MemRegions: 8914 total, 439M resident, 20M private, 311M shared.
PhysMem: 466M wired, 669M active, 306M inactive, 1441M used, 2655M free.
VM: 160G vsize, 1041M framework vsize, 59237(6) pageins, 0(0) pageouts.
Networks: packets: 119050/4500K in, 5362/379K out.
Disks: 24524/967M read, 13105/148M written.
```

PID	COMMAND	%CPU	TIME	#TH	#WQ	#POR	#MRE	RPRVT	RSHRD	RSIZE	VPRVT
539	cvmsComp_x86	0.0	00:00.04	1	0	18	33	1616K	9020K	6132K	18M
534	Grab	11.5	00:00.42	6	4	108+	170+	1792K-	33M+	12M+	-24M+
533	quicklookd	0.0	00:00.09	6	2	81	70	2236K	6924K	5584K	542M
525	mdworker	0.0	00:00.08	3	1	48	59	1612K	8160K	3504K	31M

# Utilisation

- Signaux & processus :

## EXPLOITATION

### UTILISATION

- Commande ***kill*** :
  - ✓ Emet un signal vers un processus : **kill [-l] [-NUM\_SIG] PID**
  - ✓ **Forme de communication mono directionnelle**

Signal	Rôle
1 (SIGHUP)	Envoyé par le père à tous ses enfants lorsqu'il se termine
2 (SIGINT)	Interruption du processus demandé
3 (SIGQUIT)	Même que SIGINT, mais avec génération d'un coredump
9 (SIGKILL)	Ne pas être ignoré. Force le processus à finir "brutalement"
15 (SIGTERM)	Demande d'arrêt normal, signal par défaut
19 (SIGSTOP)	Demande de suspension

# Sommaire

---

I. Fondamentaux

---

II. Système linux

---

III. Notions élémentaires

---

IV. Utilisation du shell

---

V. Essentiel

---

VI. Conclusion

---



# Essentiel

---

- Toutes les notions abordées dans ce chapitre sont fondamentales



# Conclusion



# Annexes