

# Le système d'exploitation GNU/Linux

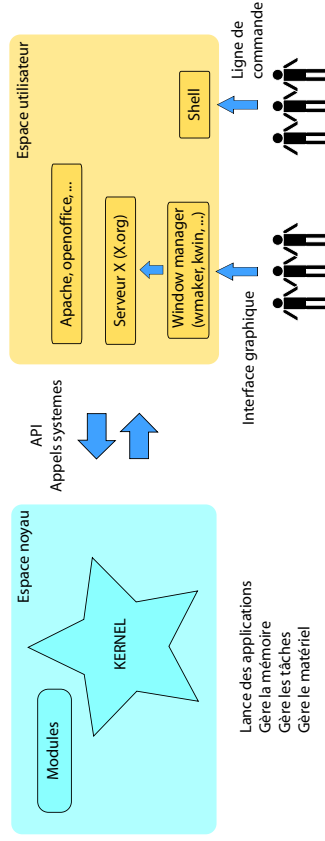
Nicolas Burrus <nicolas.burrus@laposte.net>



- Structure du système
- Initiation au shell

16 Avril 2005

## Overview



## Contenu du cours

- 1 Structure générale du système
- 2 Utilisation de base du shell
- 3 TP

## Le noyau Linux

- Ne fournit que des appels systèmes
- API en C
- Pas d'interaction directe avec l'utilisateur physique



## Arborescence standard des répertoires

- **FHS (Filesystem Hierarchy Standard)**

<http://www.pathname.com/fhs/>

- /bin
- /dev
- /etc
- /home
- /lib
- /mnt
- /proc
- /root
- /sbin
- /tmp
- /usr
- /var

## Détails de l'arborescence

### Hiéarchie habituelle de /usr

- /usr
  - /usr/X11R6 : serveur X
  - /usr/bin, /usr/lib : binaires et bibliothèques non essentielles
  - /usr/include : fichiers en-tête C, C++
  - /usr/local : programmes locaux à la machine
  - /usr/share : données indépendantes de l'architecture
- /var
  - /var/cache : données de cache
  - /var/log : journaux
  - /var/spool : fichiers en attente (imprimante, mails, ...)

## Détails de l'arborescence

- /bin : binaires essentiels au système
- /dev : fichiers de périphériques
- /etc : fichiers de configuration
- /home : répertoire des utilisateurs
- /lib : bibliothèques essentielles au système
- /mnt : points de "montage"
- /proc : paramètres dynamiques du noyau
- /root : répertoire du superutilisateur
- /sbin : binaires essentiels réservés au "root"
- /tmp : fichiers temporaires
- /usr : hiérarchie partageable entre ordinateurs
- /var : fichiers variables

## Fichiers de périphériques

### Quelques exemples de périphériques

- /dev/hdaN : 1er disque dur IDE. N : numéro de partition.
- /dev/fd0 : lecteur de disquettes
- /dev/psaux : port PS2
- /dev/dsp : carte son
- /dev/tty\* : terminaux

## Montage de systèmes de fichier

- Abstraction du support physique → répertoire
- /mnt : accueille les points de montage

### **mount** device directory

### Monte un périphérique dans un répertoire

Ex: mount /dev/hda1 /mnt/win-drive

### D'autres choses peuvent être montées

- Image ISO
- Répertoire ftp
- Lecteur réseau (NFS)

## Initiation au shell

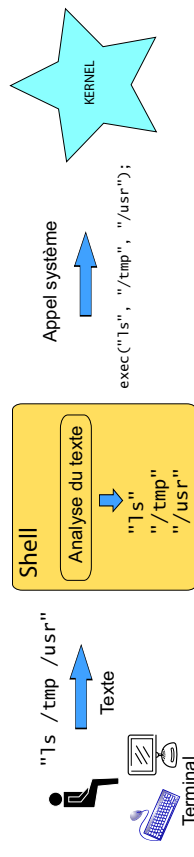
## Notion de terminal

### Périphérique d'entrée/sortie

- Les programmes peuvent être attachés à un "terminal"
- Un programme peut lire l'entrée du terminal, et peut écrire dessus
- Terminal classique : clavier + souris
- Terminal exotique : par le port série
- Terminal "virtuel" : accès distant par TCP/IP
- 6 terminaux principaux en général  
Ctrl + Alt + F[1-6]

## Fonctionnement du shell

- Principe d'un shell minimaliste



## Syntaxe

### Familles de shell

- Bourne shells (sh, bash, zsh)
- Ksh, csh

### Format usuel d'une commande

- ```
command [arg1] [arg2] ...
```
- Commandes séparées par un retour à la ligne ou un ','
  - Si arguments avec espaces, utiliser des quotes
  - ls "repertoire qui contient des espaces"

### Conventions Unix

- Options commencent par "-" ou "\_"
- Exemple : ls -l "/home"
- Les binaires sont recherchés dans le "PATH"

## Documentation

- Souvent : `command --help` ou `command -h`

### man command

Affiche le manuel de la commande

/ : rechercher une chaîne de caractère  
q : sortir

### info command

Affiche les pages "info" de la commande (plus complet)

s ou / : rechercher une chaîne de caractère  
t, u, p, n, q : top, up, previous, next, quit

Organisation hiérarchique de la documentation

<http://google.com>

## Commandes de base

```
echo [text1] [text2] ...
```

Affiche les arguments sur la sortie standard

### exit

Quitte le shell

```
ls [-l] [dir1] [dir2] ...
```

Liste le contenu du répertoire courant ou des arguments

-l : donne des détails

```
cd [path]
```

Change le répertoire courant

Sans argument : va dans le répertoire home

ex: `cd /usr/bin`; `cd ../lib`

### pwd

Affiche le nom du répertoire courant

## Lire le contenu d'un fichier

```
cat fichier1 [fichier2] ...
```

Affiche le contenu des fichiers en argument

```
less fichier1 [fichier2] ...
```

Visualise les arguments

'/ ' : rechercher une chaîne

q : quitter

```
more fichier1 [fichier2] ...
```

Visualize le contenu des arguments

[space] : avance d'un écran

[return] : avance d'une ligne

## Manipuler les fichiers

**cp** [-r] src dst

Copie le fichier src dans dst

[-r] : copie un répertoire entier

**mv** src dst

Déplace un fichier de src vers dst

**rm** [-r] file1 [file2] ...

Supprime les fileN

[-r] : supprime un répertoire

**mkdir** directory

Crée le répertoire directory

**ln** [-s] src dst

Crée un lien dst qui pointe vers src

[-s] : lien symbolique, pour les répertoires

## Autres commandes utiles sur les fichiers

**du** [-h] [directory]

Donne la taille occupée par directory

[-h] : taille sous forme agréable, kilo/mega/giga octets

**file** file1

Donne le type de contenu de file1

**df** [-h]

Donne l'espace libre sur tous les périphériques montés

[-h] : idem du

**touch** file1 [file2] ...

Crée les fichiers vides fileN

## Archivage

**gzip** file

Crée une version compressée de file dans file.gz

**gunzip** file.gz

Crée une version décompressée de file.gz dans file

**tar** cvf archive.tar file1 [file2] ...

Regroupe tous les fileN dans archive.tar

**tar** xvf archive.tar

Extrait le contenu de archive.tar

**tar** xvfz archive.tar.gz

Exécute gunzip puis extrait le contenu de archive.tar

ldem : tar cvfz archive.tar.gz file1 [file2] ...

## Wildcards et expansion

### Wildcards

- Cherche les fichiers "matchant" une certaine expression
- '\*': 1 ou plusieurs caractères
- '?': remplace un caractère
- '[abcz]': peut être remplacé par a, b, c ou z
- Exemple : `cat *.c` affiche le contenu de tous les fichiers terminant par `.c`
- `ls file?.c` listera `file1.c`, `files.c`, ...

### Expansion

- `~` : remplacé par le répertoire home
- `ls ~/dir1` : liste le contenu de `dir1` dans le répertoire home de l'utilisateur actif

## Variables d'environnement

A chaque programme sont transmises des variables spéciales

- USER : nom de l'utilisateur
- HOST : nom de la machine
- HOME : répertoire home
- PATH : liste de chemins où chercher les programmes

### En shell

- Accès aux variables : préfixe '\$'
- ls \$HOME : liste le contenu du home

### export

Liste les variables d'environnement

**export** VAR1=value

Définit la variable VAR1 contenant value

26 / 29

Nicolas Burrus

Module GNU/Linux

## Quotes

### 3 types différents

- Simple quotes ('') : aucun remplacement entre les quotes
- Doubles quotes ("") : remplacement des variables
- Backquotes (`) : exécute l'expression et se remplace par sa sortie
- ls '\$HOME' : No such file or directory
- ls "\$HOME" : contenu du répertoire home
- echo `ls` : affiche la sortie de ls

27 / 29

Nicolas Burrus

Module GNU/Linux

## Astuces

- Interrompt un programme : Ctrl + C
- Quitter le shell : Ctrl + D
- Complétion : [tab]
- Historique avec haut/bas
- Recherche dans l'historique : Ctrl + R
- Fichier commençant par '' : fichiers cachés
- Scroller dans la sortie : [Shift] + [PgUp / PgDown]
- Alias : alias ls='ls -l'
  - Taper ls devient équivalent à taper ls -l
  - unalias ls pour supprimer l'alias

28 / 29

Nicolas Burrus

Module GNU/Linux

## TP

<http://nburrus.objectis.net/teaching/linux/>

29 / 29

Nicolas Burrus

Module GNU/Linux