



# Cours Linux<sup>v. 2</sup>

« *Les commandes de base* »

Frédéric BURLET

Louis SWINNEN

{fred, louis}@namurlug.org

*5 mars 2003*

# Copyright information

Ces slides sont fournis sous licence **GNU Free Documentation License**  
These slides are provided under the **Free Documentation License**

Copyright (C) 2003 Namur LUG

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts and with no Back-Cover Texts

A copy of the license is included in the file "[fdl.txt](#)".

For more information about *free licenses*, see the GNU project website.

Source of this document can be obtained by mail to the authors.

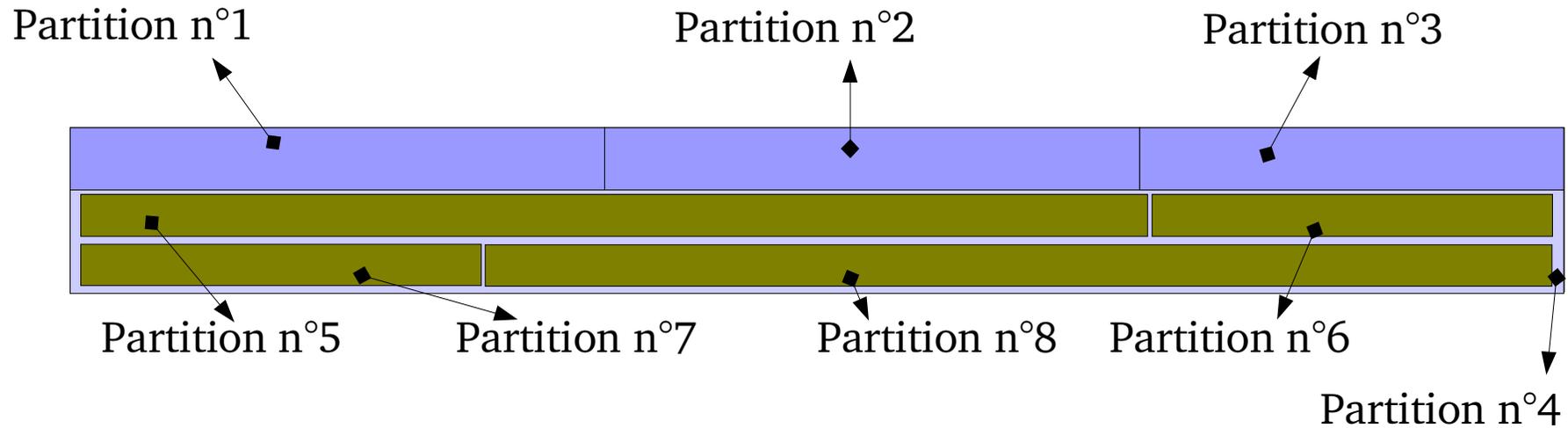
Powered by OpenOffice 1.0.1

# Notion de base (1)

- Disque dur et partition
  - Peut contenir plusieurs partitions
    - Découpage logique du disque pour contenir des données de type différent
      - partition "système"
      - partition "données utilisateurs"
    - Partition principale vs partition étendue
      - Un système peut contenir
        - jusqu'à 4 partitions principales
        - jusqu'à 3 partitions principales et 1 partition étendue
        - Une partition étendue peut contenir un nombre quelconque de *partitions logiques*

# Notions de base (2)

- Numérotation des partitions



- Partition logique
- Partition étendue
- Partition principale

# Notions de base (3)

- Concept de fichier
  - Unité minimale qui permet de contenir des données
    - Un fichier permet à un utilisateur du système de sauvegarder des informations
  - Un fichier a généralement une structure interne, souvent renseignée par l'extension, qui dépend du programme qui l'a créé
    - Une image (.png, .jpg, .gif, ...)
    - Un document (.tex, .sxi, .doc, ...)
    - Une application (.sh, .bin, .exe, ...)

# Notions de base (4)

- Concept de répertoire
  - Un répertoire permet de structurer *logiquement* les fichiers
    - Un répertoire permet de classer les informations de l'utilisateur
    - Ex: Répertoire musical, répertoire des DVDs
  - Il peut contenir des fichiers et des répertoires
    - On place dans un répertoire des informations portant sur un thème précis ou de même nature, ...
    - On obtient une structure sous forme d'arbre

# Notions de base (5)

- Répertoires sous UNIX

- / → Racine de l'arbre
- /etc → Configuration du système
- /usr → Programmes utilisateurs
- /usr/local → Programmes locaux
- /home → Répertoires personnels
- /bin → Répertoire des "applications"
- /dev → Accès aux périphériques
- /sbin → Applications réservées à l'admin.

# Notions de base (6)

- Unix et les fichiers
  - Sous Unix, tout est fichier !
    - chaque périphérique (ou port de sortie) est associé à un fichier
      - LPT -> 1 fichier
      - COM -> 1 fichier
      - Partition -> 1 fichier
      - ...
    - Un répertoire est également un fichier *spécial*

# Notions de base (7)

- A l'aide !
  - Sur le système même
    - A toute commande est associée une page de manuel  
man <commande>
    - /usr/share/doc/<programme>
    - /usr/doc/<programme>
    - /usr/share/doc/HOWTO
  - Sur le web
    - <http://www.tldp.org>
    - <http://www.lea-linux.org>
    - <http://www.linux-france.org>
  - Un LUG ou un "guru" près de chez soi

# Ligne de commande

## – Le prompt

- se présente généralement sous la forme  
`<user>@<host> :<repertoire> $`
- il est personnalisable
- il n'est que l'effet du shell ou, en français, un interpréteur de commandes

## – Le shell

- attend qu'on lui donne des commandes qu'il peut interpréter
- est de plusieurs type: bash, zsh, tcsh, ksh, sh, ash, ...

# Remarques préliminaires (1)

- Séparer les différents mots par des espaces
  - Commande valide: `cd ..`
  - Commande invalide: `cd..`
- Attention aux majuscules/minuscules
  - Commande valide: `cd`
  - Commande invalide: `Cd`



invalide si cette commande n'existe pas !

# Remarques préliminaires (2)

- Les autres mots ne faisant pas partie de la commande sont des *paramètres*
  - Un paramètre est une chaîne de caractère passée à une commande dans le but d'obtenir un résultat spécifique.
  - S'il commence par un tiret, on parle d'*options*, sinon on parle d'*arguments*
  - Exemple:

```
$ commande -option1 -option2 argument1
```

# Syntaxe

- Dans la suite, nous utiliserons la notation suivante :

<b>Syntaxe</b>	<b>Signification</b>
[x]	optionnel
fichier...	un ou plusieurs fichiers
x   y	x ou y
répertoire...	un ou plusieurs répertoires

# Commandes de base (1)

- Commandes sur les répertoires

- ls (list)

- Permet de **lister** les répertoires et les fichiers

- Commande:

- `ls [-l|-a|-h] [fichier...]`

- Exemples:

- `$ ls`

- `$ ls -l`

- `$ ls -a -l est équivalent à ls -al`

- `$ ls -lah /bin`

- `$ ls -la ~`

- `=> voir page de manuel pour plus d'options`

# Commandes de base (2)

- `pwd` (Print Working Directory)
  - Permet de **connaître le répertoire courant**
  - Commande:  
`pwd`
  - Exemple:  
`$ pwd`

# Commandes de base (3)

- `cd` (Change Directory)
  - Permet de changer le **répertoire courant**
  - Commande:  
`cd [répertoire]`
  - Exemples:

```
$ cd /home/rouge/students
```

```
$ pwd
```

```
$ cd ..
```

```
$ pwd
```

```
$ cd -
```

```
$ pwd
```

```
$ cd
```

```
$ pwd
```

# Commandes de base (4)

## - `mkdir` (Make Directory)

- Permet de **créer un répertoire**

- Commande:

```
mkdir [-m mode|-p] répertoire...
```

- Exemples:

```
$ mkdir toto tata
```

```
$ mkdir -p lulu/tutu/b212 doc/louis/linux
```

# Commandes de base (5)

- `rmdir` (Remove Directory)
  - Permet de **supprimer un répertoire vide**
  - Commande:  
`rmdir [-p] répertoire...`
  - Exemples:  
`$ rmdir tata`  
`$ rmdir -p lulu/tutu/b212`

# Commandes de base (6)

- Commandes sur les fichiers
  - mv (Move)
    - Permet de **changer le nom d'un répertoire/fichier**
    - Permet de **déplacer un répertoire/fichier**
    - Commande:

```
mv fichier_source fichier_destination
```

```
mv [-i] fichier... répertoire
```
    - Exemples:

```
$ mv toto francois
```

```
$ mv francois doc/
```

# Commandes de base (7)

## – touch

- Permet de **créer un fichier vide**
- Permet de modifier **la date** d'un fichier
- Commande:

```
touch fichier...
```

- Exemples:

```
$ touch doc/louis/linux/rapport.txt
```

```
$ touch doc/francois/hello.pas
```

```
$ touch -- -test
```

- Autres:

– voir les pages de manuel

# Commandes de base (8)

## – cp (copy)

- Permet de **copier** un fichier ou un répertoire

- Commande:

```
cp [-i|-r] fichier... répertoire
```

- Exemples:

```
$ cp -i doc/francois/hello.pas hello1.pas
```

```
$ cp -r doc/louis .
```

# Commandes de base (9)

## – rm (remove)

**DANGER**

- Permet de **supprimer** un fichier ou un répertoire

- Commande:

```
rm [-i|-f|-r] fichier...
```

- Exemples:

```
$ rm -i doc/louis/linux/rapport.txt
```

```
$ rm -rf louis
```



**DANGER**

```
$ rm -- -test
```



permet de supprimer  
un fichier commençant  
par -

# Commandes de base (10)

## – file

- Permet de **connaître le type d'un fichier**

- Commandes:

```
file fichier...
```

- Exemples:

```
$ file /usr/bin/man
```

```
$ file hello1.pas
```

```
$ file doc
```

# Commandes de base (11)

## – more

- Permet de **voir** le contenu d'un fichier texte ASCII  
**page par page**

- Commande:

more fichier

- Exemple:

```
$ more /usr/bin/sambamontage
```

Page suivante: espace

Page précédente: b

Ligne suivante: entrée

Quitter: q

Rechercher: /

# Commandes de base (12)

## – less

- Idem *more* mais plus facile à utiliser

- Commande:

```
less fichier
```

- Exemple:

```
$ less /usr/bin/sambamontage
```

Page suivante: espace ou PgDn

Page précédente: b ou PgUp

Ligne suivante: entrée ou flèches

Quitter: q

Rechercher: /

Aide: h

# Commandes de base (14)

## – cat (concatenate)

- Permet de **fusionner** des fichiers et afficher le résultat sur la sortie standard (écran par défaut)

- Commande:

```
cat fichier...
```

- Exemples:

```
$ cat .bashrc .profile
```

# Commandes de base (13)

## – du (disk usage)

- permet de connaître **l'espace occupé** par un répertoire ou un fichier

- commandes:

`du [-s|-h] fichier...`

- Exemples:

`$ du` ← résultat en kilobytes

`$ du -sh /bin /sbin` ← total avec unité

# Commandes de base (14)

## - tr (translate)

- Permet de faire **une substitution d'un ou plusieurs caractères** à partir de l'entrée standard (*clavier par défaut*)

- Commande:

```
tr [-d] groupe1 [groupe2]
```

- Exemple:

```
$ tr -d "abc" ← supprime les lettres abc
```

```
l'alphabet commence par 'abc'
```

```
$ tr cou hey ← remplace c par h, o par e et u par y
```

```
coucou ça va ?
```

```
$ tr a-mn-z n-za-m ← remplace a par n, b par o,  
n par a et z par m
```

# Commandes de base (15)

- Commandes additionnelles
  - echo
    - Permet d'afficher une ligne de texte
    - Commande:

```
echo [-n] [ligne...]
```
    - Exemple:

```
$ echo "Bonjour le monde"  
$ echo -n "Hello"
```

# Commandes de base (16)

- Commandes de recherche

- find

- Permet de **chercher** un fichier

- Commande:

- `find répertoire... expression`

- Exemple:

- `$ find / -name man`

- `$ find / -iname MaN -type d`

- `$ find ~ -iname linux -type f`

- `$ find / -name emacs -exec file {} \;`



exécute la commande *file* sur les fichiers trouvés portant le nom *emacs*

# Commandes de base (17)

## – which

- Permet de **localiser** un fichier s'il se trouve dans le PATH.
- C'est une commande interne au shell.
- PATH=chemin de recherche par défaut
- Commande:  
`which fichier...`
- Exemples:  
`$ echo $PATH`  
`$ which emacs`  
`$ which acroread`

# Commandes de base (18)

## – locate et updatedb

- Permet de **localiser** un fichier sur le disque
- Utilise une base de données construite par l'administrateur avec la commande *updatedb*

- Commande:

```
locate fichier...
```

- Exemple:

```
$ locate which
```

```
$ locate find
```

Si l'administrateur ne construit pas la base de données régulièrement (automatiquement), cette commande est inutile !

# Commandes de base (19)

## – grep

- Permet de faire une recherche à l'intérieur d'un fichier.
- Elle affiche les lignes contenant une occurrence.
- Commande:

```
grep [-r] [-i] "chaîne" fichiers...
```

- Exemples:

```
$ grep -r "cryptography" /usr/share/doc/*
```

```
$ grep -i "leda" /usr/bin/sambamontage
```