

Introduction aux systèmes Unix-like et aux commandes Shell



12 septembre 2016

Licence Creative Commons Attribution-ShareAlike 4.0 International



1 De Unix à GNU/Linux

- Les débuts avec Unix
- Le projet GNU
- Le projet Linux
- Le système d'exploitation GNU/Linux

2 Quelques commandes Shell utiles

- L'arborescence de fichiers sous Unix et notion de chemin
- Option vs argument
- Commandes de manipulation de répertoires et fichiers
- Commandes de lecture de fichiers
- Commandes d'écriture dans des fichiers
- Commandes diverses

Table des matières

- 1 De Unix à GNU/Linux
 - Les débuts avec Unix
 - Le projet GNU
 - Le projet Linux
 - Le système d'exploitation GNU/Linux
- 2 Quelques commandes Shell utiles

De Unix à GNU/Linux

Les débuts avec Unix

- Unix est l'un des premiers systèmes d'exploitation
- Début de la conception et implémentation en 1969 par Ken Thompson, Dennis Ritchie, Douglas McIlloy, Joe Ossanna qui travaillent alors pour les laboratoires Bell.
- Sorti en 1971
- 1975 : distribution du code source aux universités à des fins éducatives, moyennant l'achat d'une licence.

De Unix à GNU/Linux

Les débuts avec Unix

- 1977 : Apparition d'une des premières variantes, BSD (Berkeley Software Distribution).
- Les Unix-like se répandent rapidement dans les milieux académiques.
- 1987 : Apparition de Minix, destiné à l'éducation.
- 1999 : Apparition de Mac OS X, un système dérivé de FreeBSD (un Unix-like)

Table des matières

- 1 De Unix à GNU/Linux
 - Les débuts avec Unix
 - **Le projet GNU**
 - Le projet Linux
 - Le système d'exploitation GNU/Linux
- 2 Quelques commandes Shell utiles

De Unix à GNU/Linux

Le projet GNU

- 1983 : Richard Stallman commence le projet GNU avec pour but de créer un système libre compatible Unix.
- 1985 : Création de la *Free Software Foundation* qui a pour but de promouvoir le logiciel « libre ».

De Unix à GNU/Linux

Les quatre libertés fondamentales

- n°0 : la liberté d'exécuter le programme, pour tous les usages ;
- n°1 : la liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
- n°2 : la liberté de redistribuer des copies du programme ;
- n°3 : la liberté d'améliorer le programme et de distribuer ces améliorations au public, pour en faire profiter toute la communauté.

“Je peux expliquer le logiciel libre en 3 mots : Liberté, Égalité, Fraternité.”

—Richard Matthew Stallman, introduction de ses conférences en France

Table des matières

- 1 De Unix à GNU/Linux
 - Les débuts avec Unix
 - Le projet GNU
 - **Le projet Linux**
 - Le système d'exploitation GNU/Linux

- 2 Quelques commandes Shell utiles

De Unix à GNU/Linux

Le projet Linux

- 1991 : Linus Torvalds, alors étudiant à l'université d'Helsinki, commence à travailler sur un noyau qu'il appellera plus tard « Linux ».
- Association de Linux et des applications de Minix pour créer un système d'exploitation complet.

"I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu)..."

— Linus Torvalds, *annonçant Linux sur un forum*

Table des matières

- 1 De Unix à GNU/Linux
 - Les débuts avec Unix
 - Le projet GNU
 - Le projet Linux
 - Le système d'exploitation GNU/Linux
- 2 Quelques commandes Shell utiles

De Unix à GNU/Linux

GNU/Linux

- Au final, GNU était composé de beaucoup de logiciels mais un noyau manquait pour en faire un système d'exploitation.
- Linus Torvalds décide alors de faire de Linux un logiciel libre afin de pouvoir intégrer les logiciels du projet GNU : GNU/Linux est né.
- De nombreuses distributions sont alors apparues : Debian (1993), Archlinux (2002), Fedora (2003), Ubuntu(2004), ...

Table des matières

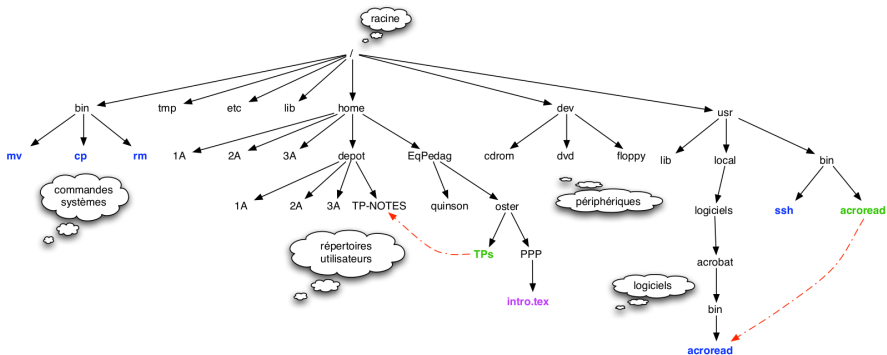
1 De Unix à GNU/Linux

2 Quelques commandes Shell utiles

- L'arborescence de fichiers sous Unix et notion de chemin
- Option vs argument
- Commandes de manipulation de répertoires et fichiers
- Commandes de lecture de fichiers
- Commandes d'écriture dans des fichiers
- Commandes diverses

Quelques commandes Shell utiles

L'arborescence de fichiers avec Unix



Quelques commandes Shell utiles

Notion de chemin

Un chemin est la représentation textuelle décrivant l'emplacement d'un fichier ou répertoire dans un système de fichiers. Le caractère « / » est à la fois le caractère représentant la racine et le séparateur de répertoire.

Il existe deux types de chemins : absolu et relatif.

La différence fondamentale entre ces deux types est le point de départ du chemin :

- la racine (/) pour un chemin absolu ;
- le répertoire courant pour un chemin relatif.

Quelques commandes Shell utiles

Notion de chemin

`/home/jpeisenbarth/intro_gnu_linux_shell.pdf` est un chemin absolu.

Un chemin relatif peut utiliser deux représentations pour indiquer son caractère relatif :

- `<< .. >>` représente le répertoire parent
- `<< . >>` représente le répertoire courant

Quelques commandes Shell utiles

Notion de chemin

Exemple de chemins relatifs :

- `../../intro_gnu_linux_shell.pdf`
(le fichier est situé deux répertoires parents au dessus)
- `./intro_gnu_linux_shell.pdf`
(le fichier est situé dans le répertoire courant)

Quelques commandes Shell utiles

Notion de chemin

Il existe un autre caractère spécial : `~` (tilde)

Bash permet de l'utiliser comme un raccourci (*Tilde Expansion*) :

`~` : le chemin absolu du *home* de l'utilisateur courant

`~user` : le chemin absolu du *home* de l'utilisateur *user*

Table des matières

1 De Unix à GNU/Linux

2 Quelques commandes Shell utiles

- L'arborescence de fichiers sous Unix et notion de chemin
- **Option vs argument**
- Commandes de manipulation de répertoires et fichiers
- Commandes de lecture de fichiers
- Commandes d'écriture dans des fichiers
- Commandes diverses

Quelques commandes Shell utiles

Différence entre option et argument

- Une option est une valeur que prend la commande pour modifier son effet/son exécution.
- Un argument est une valeur sur laquelle la commande s'exécute (une entrée du programme).

Table des matières

1 De Unix à GNU/Linux

2 Quelques commandes Shell utiles

- L'arborescence de fichiers sous Unix et notion de chemin
- Option vs argument
- **Commandes de manipulation de répertoires et fichiers**
- Commandes de lecture de fichiers
- Commandes d'écriture dans des fichiers
- Commandes diverses

Quelques commandes Shell utiles

`pwd` - print name of current/working directory

La commande `pwd` permet de récupérer le chemin complet du répertoire courant.

Quelques commandes Shell utiles

ls - list directory contents

La commande `ls` permet de lister les détails des fichiers passés en arguments ou les détails de l'ensemble des fichiers du répertoire courant si aucun argument n'est renseigné.

Options courantes :

- l : affiche selon un formatage « long »
- a : affiche également les fichiers cachés

Quelques commandes Shell utiles

cd - change the shell working directory

La commande cd change le répertoire courant.

Quelques commandes Shell utiles

`touch` - change file timestamps

La commande `touch` permet de changer la date de dernier accès ou de dernière modification d'un fichier. Si le fichier n'existe pas alors il est créé.

Quelques commandes Shell utiles

`mkdir` - make directories

La commande `mkdir` permet de créer un répertoire.

Une option courante est « `-p` » qui permet de créer les répertoires parents si nécessaire.

Quelques commandes Shell utiles

`mv` - move (rename) files

La commande `mv` permet de déplacer un fichier.

Prend deux arguments (chemins) :

- la source, qui sera déplacée
- la destination, là où sera copiée la source

Si le chemin de la source et de la destination est le même (i.e. seul le nom de fichier change) alors la commande renomme le fichier.

Quelques commandes Shell utiles

cp - copy files and directories

La commande `cp` permet de copier des fichiers ou des répertoires.

Prend deux arguments (chemins) :

- la source, qui sera copiée
- la destination, là où sera copiée la source

Quelques commandes Shell utiles

rm - remove files or directories

La commande `rm` permet d'effacer des fichiers ou des répertoires.

Options courantes :

- r : efface un répertoire et tout son contenu ;
- i : demande confirmation pour chaque fichier ;

Quelques commandes Shell utiles

chmod - change file mode bits

La commande `chmod` permet de changer les différents droits attribués à un fichier ou un répertoire.

Des explications détaillées et un exercice complet vous sera présenté dans le sujet du TP.

Table des matières

1 De Unix à GNU/Linux

2 Quelques commandes Shell utiles

- L'arborescence de fichiers sous Unix et notion de chemin
- Option vs argument
- Commandes de manipulation de répertoires et fichiers
- **Commandes de lecture de fichiers**
- Commandes d'écriture dans des fichiers
- Commandes diverses

Quelques commandes Shell utiles

cat - concatenate files and print on the standard output

La commande cat permet d'afficher entièrement le contenu d'un fichier dans la sortie standard.

Quelques commandes Shell utiles

less - opposite of more

La commande `less` permet de lire le contenu d'un fichier.

Table des matières

1 De Unix à GNU/Linux

2 Quelques commandes Shell utiles

- L'arborescence de fichiers sous Unix et notion de chemin
- Option vs argument
- Commandes de manipulation de répertoires et fichiers
- Commandes de lecture de fichiers
- **Commandes d'écriture dans des fichiers**
- Commandes diverses

Quelques commandes Shell utiles

echo - display a line of text

La commande `echo` permet d'afficher l'argument passé sur la sortie standard.

Elle s'utilise généralement avec une redirection de sortie.

Quelques commandes Shell utiles

echo et redirections de sortie

Exemple : `echo "bonjour" > file.txt`

Cette commande redirige le résultat de la commande « echo » dans le fichier « file.txt » en écrasant ce qui s'y trouvait précédemment.

Pour rediriger sans écraser (i.e. ajout à la suite du fichier), il faut utiliser « >> ».

Table des matières

1 De Unix à GNU/Linux

2 Quelques commandes Shell utiles

- L'arborescence de fichiers sous Unix et notion de chemin
- Option vs argument
- Commandes de manipulation de répertoires et fichiers
- Commandes de lecture de fichiers
- Commandes d'écriture dans des fichiers
- Commandes diverses

Quelques commandes Shell utiles

`man` - format and display the on-line manual pages

La commande `man` permet de consulter un manuel de référence.

Dans son utilisation courante cette commande attend un argument qui est la commande dont on veut lire la documentation.

Quelques commandes Shell utiles

man - format and display the on-line manual pages

Man possède plusieurs sections numérotées.

En passant un numéro compris entre 1 et 9 (impérativement en 1^{er} argument), `man` affichera la documentation de la commande correspondante à la section voulue.

Sinon il recherchera dans toutes les sections de manière croissante jusqu'à trouver une section qui documente l'argument.

Quelques commandes Shell utiles

man - format and display the on-line manual pages

Exemple : `man 1 ls`

Pour voir l'ensemble des sections, il suffit de consulter la page `man` de `man` en tapant `man man`.

Quelques commandes Shell utiles

help - display information about builtin commands

Il existe aussi une autre commande d'aide pour les commandes intégrées (*shell builtins*) : `help`

Elle attend un argument qui correspond à la commande dont on veut lire la documentation

Exemple : `help cd`