Ligne de commande sous Linux

BTS SIO - Bloc 1 - Support et mise à disposition des services informatiques

U4 - 1.2 Répondre aux incidents et aux demandes d'assistance et d'évolution

1.	Introduction - La ligne de commande sous Linux	4
	1.1. La ligne de commande	4
	1.2. Définitions	4
	1.3. Types de shells	4
	1.4. Normes	4
	1.5. Bourne Again Shell	5
	1.6. Le shell interactif	5
	1.7. Commande echo	5
	1.8. Commande ls	7
2. Entrer des commandes dans l'invite		
	2.1. Syntaxe des commandes	8
	2.2. Options et arguments	8
	2.3. Commandes hors du PATH	8
	2.4. Entrer des commandes	9
	2.5. Séquences de commandes	9
	2.6. Exécutions conditionnelles	9
	2.6.1. Opérateur &&	9
	2.6.2. Opérateur	10
	2.7. Historique des commandes	10
	2.8. Tabulation	10
	2.9. Substitution de commandes	10
	2.10. Alias	10
3. Filtres sur les fichiers		12
	3.1. Méta-caractères	12
	3.2. Caractères génériques ou jokers	12
	3.2.1. Masque générique *	12
	3.2.2. Masque de caractère ?	12
	3.2.3. Plage de valeurs []	12
	3.2.4. Filtre {chaîne1,chaîne2}	12

4. Configuration des langues, locales et clavier	
4.1. Définition de la langue	13
4.2. Définition du clavier	13
5. Aide sous Linux	
5.1. Commandes less et more	14
5.2. Commande man	14
5.3. Appel d'une page manuel	14
5.4. Recherche d'une page manuel	15
5.5. Se déplacer dans une page manuel	15
5.6. Commandes man -k ou apropos	15
6. Filesystem Hierarchy Standard (FHS)	
6.1. La structure du système de fichier	16
6.2. La commande tree	16
6.3. Partition racine	16
6.4. Contenu du système de fichier	17
6.5. Chemins relatifs et absolus	17
6.6. Se déplacer dans le système de fichiers	18
6.7. Emplacements	18

1. Introduction - La ligne de commande sous Linux

1.1. La ligne de commande

- La ligne de commande est un moyen simple d'interagir avec un ordinateur.
- Le shell interprète les commandes tapées au clavier.
- Le prompt, ou l'invite de commande, qui se termine par un \$ pour un utilisateur standard ou un # pour l'administrateur du système (désigné root), indique que le shell attend les commandes de l'utilisateur.
- Le shell est également un langage de programmation que l'on peut utiliser pour lancer des tâches automatiquement.
- Les programmes shell sont appelés par des scripts.

1.2. Définitions

- Le terminal = l'environnement d'entrée/sortie
- La console = terminal physique

Shell =

- "Interpréteur" de commande : lancer des commandes,
- Environnement : confort de l'utilisateur, sécurité
- Langage de programmation : fonctionnalités
- Traitement du texte
- Interface avec le noyau

1.3. Types de shells

On obtient la liste des shells présents sur le système en affichant le fichier /etc/shells :

```
$ cat /etc/shells
```

Résultat sur une distribution Linux Ubuntu 20.04:

```
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
```

- <u>sh</u>: historique, standard, "portable"
- <u>bash</u>: Bourn Again Shell Linux, le plus utilisé
- <u>rbash</u>: Bash Restreint
- <u>dash</u>: l'un des remplaçants de sh

1.4. Normes

- POSIX
- Single Unix Specification (SUS)

1.5. Bourne Again Shell

- Le projet GNU offre des outils pour l'administration de systèmes de type UNIX qui sont libres et qui respectent les standards UNIX.
- Bash est un Shell compatible avec sh qui incorpore des spécificités utiles du Korn Shell (ksh) et du C Shell (csh). Il est censé se conformer à la norme IEEE POSIX P1003.2/ISO 9945.2 Standards des Shell et Outils. Il offre des améliorations fonctionnelles par rapport à sh pour la programmation et l'utilisation interactive.

1.6. Le shell interactif

Quand on obtient un terminal avec une ligne de commande, on se situe dans un environnement encadré par un programme "shell" qui a créé un processus sur le système. Il permet notamment d'exécuter des commandes.

1.7. Commande echo

```
ECHO(1)
                                 User Commands
                                                                        ECHO(1)
NAME
       echo - display a line of text
SYNOPSIS
       echo [SHORT-OPTION]... [STRING]...
       echo LONG-OPTION
DESCRIPTION
       Echo the STRING(s) to standard output.
              do not output the trailing newline
       -n
              enable interpretation of backslash escapes
       -E
              disable interpretation of backslash escapes (default)
       --help display this help and exit
       --version
              output version information and exit
       If -e is in effect, the following sequences are recognized:
       \\
              backslash
              alert (BEL)
       \a
       \b
              backspace
              produce no further output
       \c
```

```
\e
               escape
               form feed
       \f
       \n
               new line
       \r
               carriage return
       \t
               horizontal tab
               vertical tab
       \ONNN byte with octal value NNN (1 to 3 digits)
               byte with hexadecimal value HH (1 to 2 digits)
       \xHH
       NOTE: your shell may have its own version of echo, which usually su-
       persedes the version described here. Please refer to your shell's
       documentation for details about the options it supports.
AUTHOR
       Written by Brian Fox and Chet Ramey.
REPORTING BUGS
       GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
       Report echo translation bugs to <a href="https://translationproject.org/team/">https://translationproject.org/team/</a>
COPYRIGHT
       Copyright (c) 2018 Free Software Foundation, Inc. License GPLv3+: GNU
       GPL version 3 or later <a href="https://gnu.org/licenses/gpl.html">https://gnu.org/licenses/gpl.html</a>.
       This is free software: you are free to change and redistribute it.
       There is NO WARRANTY, to the extent permitted by law.
SEE ALSO
       Full documentation at: <https://www.gnu.org/software/coreutils/echo>
       or available locally via: info '(coreutils) echo invocation'
GNU coreutils 8.30
                                  September 2019
                                                                            ECHO(1)
```

La commande echo permet d'afficher du texte à l'écran :

```
roger@predator:~$ echo "Affiche ce texte"
Affiche ce texte
roger@predator:~$
```

On constate que le prompt est composé de :

- roger l'utilisateur connecté
- @ séparateur
- **predator** nom de l'ordinateur
- ~ "titld" qui indique le dossier utilisateur comme dossier courant
- \$ qui indique le type de connexion

Cette configuration de l'environnement est chargée sous forme de script au moment de la connexion de l'utilisateur.

1.8. Commande Is

```
NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILEs (the current directory by default).

Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

....
```

2. Entrer des commandes dans l'invite

Pour entrer des commandes dans le shell, il faut :

- une commande valide (dans le PATH ou précisée par un chemin)
- suivie *éventuellement* d'une ou plusieurs **options** notées
 - o par un "dash", le tiret, " " en **notation abrégée**
 - o un double "dash", double tiret, " -- " en **notation extensive**,
- des arguments,
- et un **retour chariot** qui accepte la ligne entrée.

2.1. Syntaxe des commandes

Chaque commande dispose de sa propre syntaxe :

• sans options:

```
$ 1s
```

• avec une option :

```
$ 1s -1
```

• avec plusieurs options :

```
$ ls -l -a -h -t
$ ls -laht
```

2.2. Options et arguments

• Options double dash:

```
$ ls --all
$ ls --help
```

• Donner un argument :

```
$ 1s -1 /home
```

• Donner plusieurs arguments:

```
$ ls -1 /home /var
```

2.3. Commandes hors du PATH

A titre d'exemple la commande "ls" s'exécute car elle est située dans un des chemins indiqués dans la **variable d'environnement PATH**. On peut afficher ces chemins par défaut pour les fichiers exécutables via cette commande :

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/usr/games:/usr/local/games:/snap/
bin
```

On peut exécuter le logiciel directement à partir de l'emplacement absolu :

```
$ /bin/ls
```

On peut exécuter le logiciel directement à partir de l'emplacement relatif :

```
$ cd /bin
$ ./ls -l ls
```

2.4. Entrer des commandes

• On peut entrer des commandes sur plusieurs lignes :

```
$ ls /var
$ ls /home
$ ls /usr
```

• Ou sur une seule ligne on peut séparer les commandes par un "semicolon", point-virgule, ";" :

```
$ ls /var; ls /home; ls /usr
```

2.5. Séquences de commandes

• Si les arguments diffèrent pour une même commande, on peut créer une boucle et profiter de variables :

```
$ for arg in /home /var /usr
> do
> echo "visualisation : " $arg
> ls -a $arg
> done
```

• ou encore en une seule ligne

```
$ for arg in /home /usr /var; do ls -la $arg; done
```

2.6. Exécutions conditionnelles

&& et || sont des séparateurs de commandes conditionnels.

2.6.1. Opérateur &&

```
$ commande1 && commande2
```

Exécute commande2 si commande1 est exécuté sans erreur. Par exemple sous Debian/Ubuntu :

```
# apt-get update && apt-get -y upgrade
```

Par exemple sous Centos:

```
# yum -y install epel-release && yum update
```

2.6.2. Opérateur | |

```
$ commande1 || commande2
```

Exécute commande2 si commande1 est exécutée avec erreur.

Par exemple:

```
# apt-get update || yum -y update
```

2.7. Historique des commandes

Pour voir la liste des commandes que vous avez validées, vous pouvez utiliser la commande interne de bash history :

\$ history

- La commande history liste les commandes en cache ainsi que celles sauvées dans ~/.bash_history. Lorsque l'utilisateur quitte le shell, les commandes en cache sont inscrites dans ce fichier.
- Vous pouvez récupérer les commandes tapées en utilisant les flèches directionnelles (haut et bas) de votre clavier.
- history -c efface l'historique de la session courante.

2.8. Tabulation

Selon la distribution la touche de tabulation offre des possibilités d'auto-complétion.

2.9. Substitution de commandes

• La commande uname permet de connaître la version du noyau courant. Comment la substituer?

```
$ uname -a
Linux predator 5.4.0-52-generic #57-Ubuntu SMP Thu Oct 15 10:57:00 UTC 2020 x86_64 x86_64
x86_64 GNU/Linux
```

Peut se substituer par :

```
$ system=$(uname -a)
$ echo $system
Linux predator 5.4.0-52-generic #57-Ubuntu SMP Thu Oct 15 10:57:00 UTC 2020 x86_64 x86_64
x86_64 GNU/Linux
```

2.10. Alias

- Un alias est une autre manière de substituer des commandes.
- Afficher la liste des alias :

```
$ alias
...
alias l='ls -CF'
alias la='ls -A'
```

```
alias ll='ls -alF'
alias ls='ls --color=auto'
```

• Créer un alias :

```
$ alias zozo='ls -a'
$ alias
$ zozo
```

3. Filtres sur les fichiers

3.1. Méta-caractères

- Les méta caractères ont un sens spécial pour le shell. Ils sont la plupart du temps utilisés comme jokers, pour correspondre à plusieurs noms de fichiers ou de dossiers en utilisant un minimum de lettres.
- Les caractères d'entrée (<), de sortie (>) et le tube (|) sont également des caractères spéciaux ainsi que le dollar (\$) utilisé pour les variables. Notez que ces caractères sont rarement utilisés pour nommer des fichiers standards.

3.2. Caractères génériques ou jokers

3.2.1. Masque générique *

Le caractère * remplace n'importe quel nombre de caractères :

```
$ ls /usr/bin/b*
```

Liste tous les programmes commençant par "b".

3.2.2. Masque de caractère?

Le caractère ? remplace n'importe quel caractère unique :

```
$ ls /usr/bin/?b*
```

Liste tous les programmes ayant un "b" pour seconde lettre.

3.2.3. Plage de valeurs []

[] est utilisé pour définir une plage de valeurs :

```
$ ls /usr/bin/linux[0-9][0-9]
```

liste tous les fichiers commençant par "linux" suivis de deux chiffres.

```
$ ls /usr/bin/[!Aa-Yy]*
```

liste tous les fichiers qui ne commencent pas par un "a" ni par un "A" jusqu'à "y"/"Y".

3.2.4. Filtre {chaîne1,chaîne2}

{chaîne1,chaîne2} même si ce n'est pas simplement un joker de nom de fichiers, on peut l'utiliser pour filtrer des noms de fichiers :

```
ls index.{htm,html}
```

4. Configuration des langues, locales et clavier

4.1. Définition de la langue

Comment connaître la langue actuelle?

```
$ localectl status
System Locale: LANG=fr_BE.UTF-8
      VC Keymap: n/a
      X11 Layout: fr
      X11 Model: pc105
      X11 Variant: latin9
```

Valeurs possibles:

```
$ localectl list-locales | grep fr_
fr_BE.UTF-8
fr_CA.UTF-8
fr_CH.UTF-8
fr_FR.UTF-8
fr_LU.UTF-8
```

Modification:

```
$ localect1 set-locale LANG=fr_FR.utf8
$ localect1 status
System Locale: LANG=fr_FR.utf8
VC Keymap: be-oss
X11 Layout: be
X11 Variant: oss
```

Il faudra probablement quitter votre session et l'ouvrir à nouveau afin que les paramètres s'appliquent.

4.2. Définition du clavier

En console texte:

```
sudo loadkeys fr
Loading fr
sudo loadkeys be
Loading be
```

En console graphique:

```
setxkbmap fr
setxkbmap fr
```

De manière permanente :

```
sudo dpkg-reconfigure keyboard-configuration
```

5. Aide sous Linux

5.1. Commandes less et more

- less est une commande Unix permettant de visualiser un fichier texte page par page (sans le modifier). Sa fonction est similaire à la commande more, mais permet en plus de revenir en arrière ou de rechercher une chaîne. Contrairement à vi (qui permet aussi de visualiser des fichiers), less n'a pas besoin de charger entièrement le fichier en mémoire et s'ouvre donc très rapidement même pour consulter de gros fichiers.
- Raccourcis dans less:
 - o h ou help pour l'aide
 - o / suivi d'une occurrence pour effectuer une recherche
 - o Barre d'espace : pour avancer d'une page
- Exemples:

\$ more /var/log/dmesg

5.2. Commande man

man est une commande Unix. Elle permet de visionner le manuel d'une commande du shell. Elle utilise la commande less. Les raccourcis de navigation sont donc identiques.

Chaque page de manuel fait partie d'une section :

- -1. Commandes utilisateur
- -2 Appels système
- -3 Fonctions de bibliothèque
- -4 Fichiers spéciaux
- -5 Formats de fichier
- -6 Jeux
- -7 Divers
- -8 Administration système
- -9 Interface du noyau Linux

Chaque section possède une page d'introduction qui présente la section, disponible en tapant man <section> intro

Pour installer les pages de manuel en français :

```
apt-get install man-pages-fr
```

5.3. Appel d'une page manuel

Pour appeler une page de manuel, tout simplement :

```
$ man [-s<section>] <nom_de_commande>
```

Par exemple:

```
$ man man
$ man ls
```

\$ man 5 passwd

5.4. Recherche d'une page manuel

Une page de manuel peut avoir le même nom et faire partie d'une section différente (la portée de la page est différente). Par exemple :

```
$ man -f passwd
passwd (1) - Modifier le mot de passe d'un utilisateur

passwd (5) - fichier des mots de passe

passwd (1ssl) - compute password hashes
```

L'option man -f passwd permet d'effectuer une recherche sur le nom des pages man. La commande whatis passwd aurait eu le même effet.

5.5. Se déplacer dans une page manuel

- G pour arriver à la fin du document
- gg pour revenir au début du document

5.6. Commandes man -k ou apropos

```
$ man -k password
```

Recherche la description courte et le nom des pages de manuel comportant le mot-clé, utilisé comme une expression rationnelle, puis affiche tout ce qui a été trouvé.

La commande apropos donne l'équivalent :

```
$ apropos password
```

La commande whatis -r donne également le même résultat.

Man utilise une base donnée pour consulter les descriptions des pages. En cas de pages ou de logiciels ajoutés, il est indiqué de mettre à jour la base de données mandb :

\$ mandb

6. Filesystem Hierarchy Standard (FHS)

https://fr.wikipedia.org/wiki/Filesystem Hierarchy Standard

6.1. La structure du système de fichier

- Un système de fichiers est similaire à une arborescence, avec une racine qui se scinde en branches et sous-branches, soit en répertoires et sous-répertoires.
- On commence par le tronc principal, la racine (root) : / . C'est un peu comme le C:\ sous DOS, sauf que C:\ est également le premier périphérique de stockage, alors que la racine peut correspondre à n'importe quel disque (partition) de votre système (point de montage).
- La racine contient différents répertoires et sous-répertoires contenant eux-mêmes des fichiers.

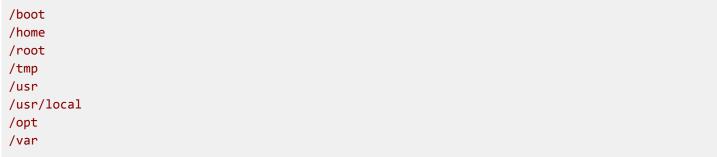
6.2. La commande tree

La commande tree liste le contenu des répertoires sous forme d'arborescence.

```
$ tree -L 1 /
   backup
   - bin -> usr/bin
   - boot
    - cdrom
   - data
   - dev
   - etc
   - home
   - lib -> usr/lib
   - lib32 -> usr/lib32
   - lib64 -> usr/lib64
   - libx32 -> usr/libx32
   - lost+found
   - media
   - mnt
   - opt
   - proc
    - root
   - run
   - sbin -> usr/sbin
   - snap
   - srv
    - swapfile
   - sys
    tmp
     usr
    var
```

6.3. Partition racine

Les répertoires suivants peuvent être montés sur d'autres partitions que la celle de la racine :



Les répertoires /dev, /bin, /sbin, /etc et /lib doivent être montés sur la partition racine.

De plus, la racine doit contenir un répertoire /proc vide. Il est utilisé par le noyau pour informer sur le statut du système d'exploitation (processus, statistiques d'utilisation de la mémoire, etc.).

6.4. Contenu du système de fichier

- /bin et /sbin : contiennent les binaires nécessaires au démarrage et les commandes essentielles
- /dev : fichiers périphériques ou fichiers spécifiques
- /etc : fichiers et répertoires de configuration spécifiques à la machine
- /lib et /lib64 : bibliothèques partagées pour les binaires de /bin et /sbin . Contient également les modules du noyau.
- /mnt ou /media : points de montage pour les systèmes de fichiers externes
- /proc : informations du noyau. En lecture seule sauf pour /proc/sys
- /boot : contient le noyau Linux, le System.map (carte des symboles du noyau) et les chargeurs d'amorçage secondaires.
- /home (facultatif) : répertoires utilisateurs, avec, en général, une copie du contenu de /etc/skel
- /root : répertoire de l'utilisateur root.
- /sys : export d'information du noyau, à la manière de /proc
- /tmp : fichiers temporaires.
- /usr: User Specific ressources. Contenu essentiellement statique et partageable. /usr est composé de sous-répertoires bin, sbin, lib et autres qui contiennent des programmes et bibliothèques de votre système non essentielles ni nécessaires au démarrage.
- lost+found : est un dossier spécial de récupération des données du système de fichiers.
- /usr/local ou /opt : programmes et bibliothèques supplémentaires. En général, c'est dans ces répertoires que l'on place les programmes qui ne font pas partie des paquets des distributions.
- /var : données variables comme les spool ou les journaux. Les sous-répertoires peuvent être soit partageables (comme /var/spool/mail) soit non partageables (comme /var/log).
- /var/www, /var/ftp ou /srv : pages web ou fichiers ftp anonymes.

6.5. Chemins relatifs et absolus

- On peut accéder à un répertoire ou un fichier en donnant son chemin complet, qui commence à la racine (/), ou en donnant son chemin relatif partant du répertoire courant.
- Chemin absolu:
 - o indépendant du répertoire de travail de l'utilisateur
 - o commence par /
- Chemin relatif:

- o dépend de l'endroit où se trouve l'utilisateur
- o ne commence pas par /

6.6. Se déplacer dans le système de fichiers

Comme pour tout système de fichiers structuré, un certain nombre d'outils aident à parcourir le système. Les deux commandes suivantes sont des commandes internes du shell :

- pwd : (Print Working Directory) affiche le répertoire actuel en chemin absolu
- cd : la commande pour changer de répertoire (Change Directory)

6.7. Emplacements

- L'emplacement courant est représenté par un point .
- L'emplacement parent est représenté par deux points ..
- Le répertoire utilisateur courant est représenté par le tild ~