

GNU/Linux

Ligne de commande

Nicolas Delanoue

Université d'Angers - Polytech Angers



POLYTECH[°]
ANGERS



- 1 Shell
- 2 Utilisation du système de fichiers
- 3 Comptes utilisateurs
- 4 Permissions et utilisateurs

Définition

Le *shell* est un programme qui interprète des commandes.

Remarques

- Il existe plusieurs shells. `Bash` en est un, mais il en existe d'autre comme par exemple `sh`.
- Un shell s'exécute dans une *console* aussi appelée *terminal*.

Définition

Une *ligne de commande* est une instruction donnée à un shell.

A quoi cela sert ?

A tout faire !

Définition

Le *prompt* est un message invitant à taper une commande.

```
nico@pc:~$ ls
adminstratif2      gurobi.lic        perso
Animation_scientifique gurobi.log        perso2
articles           images            recherche
Bureau             liste_pour_tim.txt~ Steam
congres_seminar   livre_de_vadim.txt Téléchargements
dosbox             livres            tmp
dosbox.conf       monlivre          tout.pdf
eclipse           monter_partage    video
nico@pc:~$
```

Les touches clavier

Tab complète le nom de fichier/de répertoire s'il n'y en a qu'un seul qui correspond

Tab Tab dans le cas où un seul appui de tab n'a rien affiché, affiche la liste de toutes les possibilités,

Flèche_Haut rappelle la commande précédente,

Shift Page_Up remonte d'une page dans la console,

Ctrl C arrête le processus en cours d'exécution dans la console (envoi au processus du signal SIGINT),

Ctrl Z suspend le processus en cours d'exécution dans la console (envoi au processus du signal SIGSTOP),

Ctrl L efface l'écran du terminal,

Sélection sélectionne le texte (=copier)

Clic Milieu l'insère à la position actuelle du curseur de texte (=coller)

Organisation

- Système de fichiers hiérarchique, c'est à dire qu'on a une arborescence.
- Fichiers et répertoires sont des entrées

Contraintes sous ext4 Linux

- Les noms sont sensibles à la casse
- Les noms de fichiers n'ont pas nécessairement d'extension

Note pour l'enseignant : construire avec les étudiants un système de fichiers fictif.

Organisation du système de fichiers de Linux

- `/bin` commandes exécutables de base du système
- `/boot` fichiers de démarrage du noyau Linux
- `/dev` périphériques de la machine sous la forme de fichiers
- `/etc` fichiers de configuration du système
- `/home` répertoires de base des utilisateurs pour qu'ils sauvent leurs documents,
- `/lib` bibliothèques partagées et modules du noyau
- `/media` point de montage pour les périphériques amovibles
- `/mnt` point de montage standard des systèmes externes
- `opt` applications supplémentaires
- `/proc` informations sur les ressources système
- `/root` répertoire de base du super-utilisateur (root)
- `/sbin` commandes exécutables réservées au root
- `/tmp` répertoire temporaire
- `/usr` contient les applications, bibliothèques partagées, documentation
- `/var` la partie "variable" du système : fichiers de log, files d'attente mail, imprimante ...

Définition

Un *chemin* permet d'identifier un fichier ou un répertoire.

Remarques

- Un chemin
 - énumère tous les répertoires traversés,
 - aboutit à la destination.
- Les répertoires traversés sont séparés par des /

Définition

Un chemin qui part de la racine / est appelé *absolu*.

Exemples de chemins absolus :

- /home/etudiant/
- /bin/find
- /root/doc/presentation.ppt

Chemins spéciaux

Le répertoire **courant**, noté `.`, est celui dans lequel on travaille.

Le répertoire **parent**, noté `..`, est celui au-dessus du répertoire courant,

Le répertoire **de base**, accessible via `~`, est le répertoire des documents de l'utilisateur connecté.

Remarques

Les répertoires `.` et `..` sont des chemins relatifs par rapport au répertoire courant.

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`

solution : `/home/etudiant/polytech/rapportTP.doc`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`
solution : `/usr/bin/firefox`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`
solution : `/usr/bin/firefox`
- 4 `./jeux/stupides/pileouface`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`
solution : `/usr/bin/firefox`
- 4 `./jeux/stupides/pileouface`
solution : `/home/etudiant/jeux/stupides/pileouface`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`
solution : `/usr/bin/firefox`
- 4 `./jeux/stupides/pileouface`
solution : `/home/etudiant/jeux/stupides/pileouface`
- 5 `..`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`
solution : `/usr/bin/firefox`
- 4 `./jeux/stupides/pileouface`
solution : `/home/etudiant/jeux/stupides/pileouface`
- 5 `..`
solution : `/home/`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`
solution : `/usr/bin/firefox`
- 4 `./jeux/stupides/pileouface`
solution : `/home/etudiant/jeux/stupides/pileouface`
- 5 `..`
solution : `/home/`
- 6 `.`

Exercice

En supposant que le répertoire en cours est `/home/etudiant`, donnez le chemin absolu des répertoires suivants :

- 1 `polytech/rapportTP.doc`
solution : `/home/etudiant/polytech/rapportTP.doc`
- 2 `../prof/texteTP.pdf`
solution : `/home/prof/texteTP.pdf`
- 3 `/usr/bin/firefox`
solution : `/usr/bin/firefox`
- 4 `./jeux/stupides/pileouface`
solution : `/home/etudiant/jeux/stupides/pileouface`
- 5 `..`
solution : `/home/`
- 6 `.`
solution : `/home/etudiant`

Quelques commandes à connaître

En supposant que le répertoire en cours est `/home/etudiant` :

Où suis-je ?	<code>pwd</code>	print working dir	<code>\$ pwd</code> <code>/home/etudiant/</code>
Qu'y a-t-il ici ?	<code>ls</code>	list	<code>\$ ls</code> Bureau rapportTP.odp cv.pdf
Changer de répertoire	<code>cd</code>	change directory	<code>\$ cd /home</code> <code>/home/\$</code>
Afficher le contenu d'un fichier	<code>cat</code>	catalog	<code>\$ cat /etc/passwd</code>

Copier	cp	copy	<pre>\$ cp rapportTP.doc ../prof \$ cp Bureau/linux.pdf . \$ cp -r /tmp/dossier .</pre>
Déplacer / re-nommer	mv	move	<pre>\$ mv comptes.txt compta.txt \$ mv Bureau/linux.pdf . \$ mv dossier/ Bureau/</pre>
Supprimer	rm	remove	<pre>\$ rm comptes.txt \$ rm Bureau/linux.pdf . \$ rm -rf /tmp/dossier/</pre>

Créer un fichier vide	touch	<pre>\$ touch test0.txt \$ ls Bureau test0.txt</pre>
Créer un répertoire	mkdir	<pre>\$ mkdir photos</pre>
Rechercher un fichier	find	<pre>\$ find / -name "compt*"</pre>

puis :

diff, locate, whereis, ...

Définition

Un *compte d'utilisateur* est l'ensemble des ressources informatiques attribuées à un utilisateur ou à un appareil (ordinateur, périphérique...).

Remarques

- Un utilisateur peut être une personne ou bien un rôle,
- une personne peut utiliser plusieurs comptes, par exemple j'utilise le compte `nico` et le compte `root`,
- un compte peut être utilisé par plusieurs personnes.

Pourquoi plusieurs comptes ?

- Chaque personne veut protéger ses données,
- chacun ne peut pas tout faire sur la machine.

Mécanismes de sécurité

- On protège les fichiers.
- Les permissions sur les fichiers sont accordées à des utilisateurs.

Localisation de la liste des comptes utilisateurs

- Les comptes sont listés dans le fichier `/etc/passwd`.
- Les mots de passe sont dans le fichier `/etc/shadow`.

```
user@pc:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
...
gdm:x:108:118:gdm:/var/lib/gdm:/bin/false
nico:x:1000:1000:nicolas,,,:/home/nico:/bin/bash
```

Explication de la dernière ligne de ce fichier

- `nico` est le nom du compte utilisateur,
- `1000` est le identifiant unique de l'utilisateur, aussi appelé iud,
- `nicolas,,,` correspond à la description de l'utilisateur,
- `/home/nico` est son répertoire personnel,
- `/home/nico:/bin/bash` est le programme qui s'exécute à l'ouverture de session.

Les comptes : créer/supprimer

- avec l'interface graphique : facile
- en ligne de commande :
 - useradd pour ajouter,
 - usermod pour modifier,
 - userdel pour supprimer,
 - passwd pour changer le mot de passe.

Remarque

Tous ces modifications impactent les fichiers passwd, shadow du répertoire /etc.

Les comptes : changer d'utilisateur

Une même personne peut, pour remplir différents rôles, devoir changer de compte

- graphiquement : se déconnecter ou se reconnecter,
- en ligne de commande : avec la commande `su` et `exit`.

Exemple :

```
nico@pc1:~$ su xavier
```

```
Mot de passe: *****
```

Résultat, on est connecté en tant que xavier :

```
xavier@pc1:/home/nico$
```

Pour s'en assurer, utiliser `whoami` :

```
xavier@pc1:/home/nico$ whoami
```

```
xavier
```

Les comptes : changer d'utilisateur.

```
xavier $ su reivax
```

```
reivax $ ...  
reivax $ ...  
reivax $ su root -
```

```
xavier $ su reivax
```

```
reivax $ su root -  
root # ...  
root # ...  
root # exit
```

```
xavier $ su reivax  
xavier $
```

```
xavier $ su reivax
```

```
reivax $ su root -  
reivax $ ...  
reivax $ ...  
reivax $ exit
```

Deux possibilités pour devenir le super utilisateur, i.e. root

- 1 Changer d'utilisateur avec la commande `su root` -
 - nécessite le mot de passe du root.
- 2 Utiliser le mécanisme de `sudo`
 - permet d'exécuter une commande sous root, (juste le temps de la commande)
 - nécessite d'y être autorisé ...



FIGURE – Et là, je suis le roi !

Démonstration du mécanisme sudo

```
nico@pc:~$ ls /dossier_sensible
```

```
ls: impossible d'ouvrir le répertoire '/root':  
Permission non accordée
```

```
nico@pc:~$ sudo ls /dossier_sensible
```

```
[sudo] Mot de passe de nico :  
fichier_1_important.pdf truc_sensible.txt
```

Ici, il faut taper le mot de passe de nico et non pas celui de root.

Permissions et utilisateurs

Définition

Un *utilisateur* est « quelqu'un » de `/etc/passwd`.

Définition

Un *groupe* est « quelque chose » de `/etc/group`. Un groupe rassemble des utilisateurs.

Propriétés du système de fichier `ext4`

- Un utilisateur appartient forcément à un groupe (qui porte son nom, en général),
- un fichier (ou un répertoire) possède un utilisateur et un groupe (propriétaires).

Définition

Un fichier ou répertoire possède des *permissions*

Permissions en ext4

Il y a 3 jeux de permissions :

- Concernant le propriétaire noté *u*
- Concernant le groupe noté *g*
- Concernant les autres noté *o*

Chaque jeu de permissions autorise (ou non) :

- La lecture (ou le listing) notée *r*
- L'écriture (ou la création) notée *w*
- L'exécution (ou l'entrée) notée *x*

```
user@pc:~/ $ ls -l
drwxr-x--- 27 nico g1      4096 nov.   5 09:25 articles
-rw-rw-r--  1 nico nico   1619 oct.   2 2017 monip.txt
```

Explication

Le fichier articles

- est un dossier (à cause du premier d),
- appartient à l'utilisateur `nico` qui peut
 - lister les fichiers qu'il contient (`r`)
 - créer des fichiers à l'intérieur de ce dossier (`w`)
 - rentrer dans ce dossier (`x`)
- est rattaché au groupe `g1`, les utilisateurs du groupe peuvent
 - lister les fichiers qu'il contient (`r`)
 - pas créer des fichiers à l'intérieur de ce dossier (`w`)
 - rentrer dans ce dossier (`x`)
- et le reste du monde ne peut rien faire.

```
user@pc:~/ $ ls -l
drwxr-x--- 27 nico g1      4096 nov.   5 09:25 articles
-rw-rw-r--  1 nico nico   1619 oct.   2 2017 monip.txt
```

Explication

Le fichier `monip.txt`

- est un fichier régulier (à cause du premier `-`),
- appartient à l'utilisateur `nico` qui peut
 - ouvrir le fichier (`r`)
 - modifier ce fichier (`w`)
 - pas exécuter ce fichier (`x`)
- est rattaché au groupe `nico`, les utilisateurs du groupe peuvent
 - ouvrir le fichier (`r`)
 - pas modifier ce fichier (`w`)
 - pas exécuter ce fichier (`x`)
- et le reste du monde peut seulement lire le fichier.

Notation

Les permissions peuvent aussi s'écrire en octal. En effet, chaque permission correspond à une valeur :

- La lecture, notée *r*, correspond à la valeur 4,
- L'écriture, notée *w*, correspond à la valeur 2,
- L'exécution, notée *x*, correspond à la valeur 1,

Exemple

La permission *rx* *r-x* *r--* correspond donc à 7 5 4, effets

- *rx* correspond à $4+2+1$, i.e. 7.
- *r-x* correspond à $4+0+1$, i.e. 5.
- *r--* correspond à $4+0+0$, i.e. 4.

Changer les permissions

La commande pour changer les permissions est `chmod`.

Remarque

On peut changer les permissions si :

- On est le propriétaire,
- on est root.

Exemples de syntaxe

```
nico@pc:~/ $ chmod 750 rapport.doc
```

```
nico@pc:~/ $ chmod ug+w rapport.doc
```

On pourra utiliser l'option `-R` appeler récursivement.

Changer le propriétaire

La commande pour changer le propriétaire est `chown`.

Remarque

On peut changer le propriétaire si :

- On est le propriétaire,
- on est root.

Exemple

```
nico@pc1:~/ $ chmod nico:group document.txt
```

On pourra aussi utiliser l'option `-R`.

Les commandes à connaître

