

**DEV1 – ENVL – Laboratoires d'environnement****TD 2 – Introduction (partie II)****Table des matières**

<b>1</b>	<b>Le système de fichiers</b>	<b>2</b>
1.1	La racine du système de fichiers . . . . .	2
1.2	Chemin absolu et relatif . . . . .	2
1.3	Raccourcis pour des chemins . . . . .	3
<b>2</b>	<b>La ligne de commandes</b>	<b>4</b>
2.1	Compléter une commande . . . . .	4
2.2	Les jokers . . . . .	4
2.3	Historique . . . . .	4
2.4	Les options . . . . .	5
<b>3</b>	<b>Se documenter</b>	<b>5</b>
3.1	Recherche d'informations . . . . .	5
<b>4</b>	<b>Conclusion</b>	<b>6</b>

# 1 Le système de fichiers

## 1.1 La racine du système de fichiers

Rappelez-vous que les fichiers sont organisés en hiérarchie : un dossier contient d'autres dossiers qui lui-même. . .

On a vu que chaque utilisateur dispose d'un dossier personnel. Où se trouve-il ? Dans un dossier appelé **home**. Et où se trouve ce dossier **home** ? Dans un dossier appelé « / » (la barre oblique). Et où se trouve ce dossier « / » ? Nulle part ;) C'est le dossier principal (la **racine**).

### La racine du système de fichiers

La **racine** est le dossier principal, tout en haut de la hiérarchie des dossiers.

## 1.2 Chemin absolu et relatif

*« Tout est relatif, et cela seul est absolu. » – Auguste Comte*

Régulièrement, il faut indiquer un endroit du système de fichiers (par exemple pour y aller). Désigner simplement le nom du dossier ne suffit pas. Nous avons déjà abordé la notion de *chemin* ; Précisons.

### Chemin absolu et relatif

#### Chemin

Un **chemin** est une suite de dossiers à traverser pour arriver au dossier ou au fichier qui nous intéresse. On les sépare par « / ».

Par exemple : **home/g12345/dev1/td1/test** indique que dans le dossier **home** il y a un dossier **g12345** qui contient un dossier **dev1** qui contient un dossier **td1** qui contient un fichier (ou un dossier) **test**.

#### Chemin absolu

Un chemin est **absolu** s'il commence à partir de la **racine**. Il commence donc par « / ». Par exemple : **/home/g12345/td1/test**

#### Chemin relatif

Dans le cas contraire il est **relatif**. Il s'agit d'un chemin à suivre à partir du **dossier courant**.

Un chemin **absolu** désigne toujours le même endroit. Un chemin **relatif** non ! ça dépend d'où on est, du dossier courant.

### Exercice 1 pwd

Entrez la commande `pwd` (que fait-elle encore ?). Comprenez-vous la notation qu'elle utilise pour la réponse ? Est-ce un chemin absolu ou relatif ?

Rappel : dans une commande, un nom de fichier (ou de dossier) désigne toujours un fichier (ou un dossier) se trouvant dans le dossier courant  
Pour indiquer un **fichier** (ou un dossier) se trouvant **ailleurs**, il faut donner son **chemin** (absolu ou relatif).

Par exemple : `ls /home/g12345` permet de visualiser le contenu de la home de g12345 (quel que soit l'endroit où on se trouve puisque c'est un chemin absolu).

## Exercice 2

### Utilisation d'un chemin

Supposons que vous êtes dans votre home et que vous ne pouvez pas vous déplacer (la commande `cd` est interdite!).

1. Affichez le contenu du fichier `test` (qui se trouve dans le dossier `td1`) en utilisant un chemin **relatif**.
2. Faites la même chose en utilisant un chemin **absolu**.
3. Créez une copie du fichier `welcome` dans le dossier `td1`.
4. Auriez-vous pu faire cette copie sans utiliser de chemin, en vous déplaçant (`cd`) ?

## 1.3 Raccourcis pour des chemins

### Raccourcis

Dans un chemin :

- ▷ `~g12345` : désigne la home de l'utilisateur g12345 ;
- ▷ `~` : désigne la home de l'utilisateur qui entre la commande ;
- ▷ `..` : désigne le dossier parent du dossier courant ;
- ▷ `.` : désigne le dossier courant.

## Exercice 3

### Se familiariser avec les raccourcis

Placez-vous dans votre dossier personnel. En utilisant les raccourcis afin de trouver la solution la plus courte possible :

1. Affichez le contenu de votre dossier personnel en utilisant un chemin absolu.
2. Affichez le contenu de la home de votre professeur en utilisant un chemin absolu.
3. Refaites la même chose mais en utilisant un chemin relatif.

## Exercice 4

### Comprendre un chemin

Que désigne le chemin suivant : `~mcd/../../home` ?

## Exercice 5

### Chemins absolus et relatifs

Parmi tous les chemins suivants, quels sont ceux qui sont **relatifs** ?

- ☐ `~/../g12345/td2`
- ☐ `/home/g12345/../../g54321/Hello.java`
- ☐ `./tds/td2`
- ☐ `tds/td2/Hello.java`
- ☐ `~g12345/tds/td2`

## 2 La ligne de commandes

### 2.1 Compléter une commande

Parfois, vous devez entrer une commande assez longue parce que les noms de fichiers sont longs et/ou nombreux. Linux offre plusieurs facilités pour simplifier l'entrée de longues commandes.

#### Experience 1

#### La complétion de la commande

Supposons que vous ne vous rappeliez plus très bien de la commande qui permet de modifier le mot de passe. Vous vous rappelez juste qu'elle commence par `pas`.

- ✍ Tapez `pas` sans appuyer sur `ENTER`.
- ✍ Appuyez 2× sur la touche `TAB`. Le shell affiche la liste des commandes commençant ainsi.
- ✍ Tapez sur `s` puis à nouveau 2× sur la touche `TAB`. Comme il n'y a plus qu'une seule possibilité, le shell complète la commande.

#### Exercice 6

#### La complétion des noms de fichiers

La touche de tabulation permet également de compléter un nom de fichier.

1. Copiez chez vous le fichier  
`monfichier aunom tellement long qu'il me paraît peu probable de taper 2x sans erreur`  
qui se trouve dans le dossier `/eCours/java/td/td3`.
2. Affichez le contenu de ce fichier en évitant de retaper son nom.

### 2.2 Les jokers

TODO (motivation, expérience, exercice...)

#### Les jokers

Dans un nom de fichier/dossier

- ▷ « `?` » remplace **un et un seul** caractère quelconque.
- ▷ « `*` » remplace **0, 1 ou plusieurs** caractères quelconques.

### 2.3 Historique

Il arrive souvent qu'on oublie une commande qu'on a déjà utilisée avant. Linux retient un historique des commandes tapées par un utilisateur et il est possible de l'exploiter.

### Historique

- ▷ Les flèches *haut* et *bas* permettent de se déplacer dans les commandes déjà entrées.
- ▷ `history` affiche la liste (numérotée) des dernières commandes tapées.
- ▷ `!n` réexécute la commande numéro *n*.
- ▷ `!mot` réexécute la commande la plus récente **commençant** par *mot*.
- ▷ `!?mot` réexécute la commande la plus récente **contenant** par *mot*.
- ▷ `^chaine1^chaine2^` réexécute la dernière commande en remplaçant *chaine1* par *chaine2*.

## 2.4 Les options

La plupart des commandes possèdent des **options**.

### Les options

Une **option modifie le sens** d'une commande ;

- ▷ Elle commence par le signe - suivi d'une seule lettre ;
- ▷ ou encore par le double tiret -- suivi d'un nom d'option.
- ▷ Elle est placée n'importe où après le nom de la commande.

#### Experience 2

### Expérimenter les options

- ✍ Tapez la commande `ls -l`. Vous constatez que le résultat obtenu est beaucoup plus verbeux que celui obtenu sans l'option.
- ✍ Placez vous dans le dossier `td1` et tapez la commande `cat -n test`. L'option demande de numéroté les lignes
- ✍ Essayez `cat --number test`, la version *longue* équivalente à la précédente.

## 3 Se documenter

### 3.1 Recherche d'informations

Non seulement, il y a beaucoup de commandes à connaître mais, en plus, chacune dispose d'une multitude d'options. Impossible de tout retenir ! Comment faire ?

### Information

- ▷ `nomCommande --help` affiche une aide succincte sur la commande.
- ▷ `man nomCommande` affiche une documentation plus complète. (*q* pour quitter).

Si vous ne connaissez pas le nom de la commande, consultez les documents que l'on met à votre disposition, notamment le *Linux Cheat Sheet*.

#### Exercice 7

### Trouver la bonne option

La commande `ls -l` affiche le contenu du dossier en format *long*. La 5<sup>e</sup> colonne donne la taille du fichier (en octets). Lorsque les nombres sont grands, ce n'est pas très lisible. Trouvez l'option qui permet d'afficher cette taille sous un format plus lisible.

De grâce, **cherchez** la réponse, ne la **demandez pas** à votre voisin. Le but de cet exercice n'est pas de connaître l'option (elle n'est pas si utile que ça) mais d'apprendre à trouver soi-même l'information.

#### Exercice 8

#### Trouver la bonne commande

Quelle commande permet de

- ▷ nettoyer l'écran ?
- ▷ afficher la date et l'heure ?

## 4 Conclusion

### Notions importantes de ce TD

Voici les notions importantes que vous devez avoir assimilées à la fin de ce TD.

- ▷ Comprendre la notion de chemin absolu et relatif. Savoir les utiliser dans les commandes.
- ▷ Connaître la signification dans un chemin de : « ~ », « . » et « .. ».
- ▷ Pouvoir demander plus d'informations sur une commande, ses options. Comprendre les informations fournies.

#### Exercice 9

#### Exercice récapitulatif

TODO...

### Félicitations !

Vous êtes arrivé au bout de ce premier TD. Avant de quitter le laboratoire, n'oubliez pas de quitter proprement la connexion avec `linux1` (`exit`) et d'éteindre l'ordinateur (ou au moins de vous déloguer).

À la semaine prochaine et soyez à l'heure !