Université Paris 13 L1
Institut Galilée Année 2014-2015
Licence 1 - 1er semestre

Pré-rentrée : commandes linux et inititation à la programmation

L'objectif de ce TD est de vous familiariser avec le système d'exploitation Linux, et plus particulièrement le Terminal, qui permet d'effectuer des commandes usuelles (copier des fichiers, créer des dossiers, ...) sans passer par une interface graphique. Vous verrez ensuite des petits exercices simples de programmation, à terminer chez vous.

1 Quelques commandes Linux

Dans l'exercice suivant, vous utiliserez le Terminal de Linux afin de réaliser des opérations sur des fichiers. Afin de voir que ces opérations ont un impact dans vos fichiers, vous ouvrirez un explorateur de fichier pour visualiser ces derniers.

1. Commencez par ouvrir **un Terminal Linux**. Vous obtenez une fenêtre ressemblant à ce qui se présentait à l'écran des vieux ordinateurs dans les années 80.
2. Ouvrez aussi un **explorateur de fichier** (Menu Raccourci > Mes fichiers) : vous obtenez une fenêtre ressemblant plus à ce dont vous êtes habitués sous Windows.

Les commandes de base

Grâce au Terminal, vous pourrez effectuer des opérations sur vos fichiers (création de dossier, copie de fichiers, ...) ou lancer des programmes en écrivant quelques commandes au clavier. Vous vous rendrez rapidement compte qu'avec le temps, le Terminal vous permet de faire des opérations plus rapidement qu'avec l'explorateur de fichiers.

1. Toutes les opérations que vous demandez au Terminal sont effectuées dans un répertoire. **Pour connaître dans quel répertoire se situe le Terminal**, il faut taper la commande

   ```
   pwd
   ```

   puis valider avec la touche "Entrée" (cette commande signifie *Print Working Directory*, que l'on peut traduire comme afficher le répertoire de travail). Le Terminal vous affiche alors le répertoire où il se situe.

2. Nous allons maintenant **créer un dossier** avec la commande **mkdir**. Pour créer un dossier dans le répertoire où se situe le Terminal, il suffit de taper la commande suivie du nom du répertoire. Pour créer un dossier nommé *TestP13*, nous allons taper dans le Terminal

   ```
   mkdir TestP13
   ```

   puis valider avec la touche "Entrée" (cette commande signifie *MaKe DIRectory*, que l'on peut traduire comme créer dossier). Vérifiez avec l'explorateur de fichiers que le dossier a bien été créé.

3. Pour **afficher le contenu du répertoire courant**, vous devez utiliser la commande

   ```
   ls
   ```

(cette commande signifie *LiSt*). Vérifiez que ce que vous affiche cette commande correspond avec ce que vous affiche l'explorateur de fichiers. Voyez-vous des éléments en plus ou en moins ?

4. Pour faire **naviguer le Terminal dans un autre répertoire**, on utilise la commande **cd**, signifiant *Change Directory* (que l'on peut traduire comme changer de répertoire), suivie du nom de répertoire où l'on désire se rendre. Par exemple, pour rentrer dans le dossier *TestP13*, on fait

```
cd TestP13
```

Vérifiez ensuite avec la commande **pwd** que le Terminal s'est bien déplacé dans le dossier *TestP13*.

5. Grâce à ce que vous venez d'apprendre, créer, dans le dossier *TestP13*, un dossier *Temp*, et entrez-y avec le Terminal. Vérifiez que le Terminal s'est bien placé dans le bon répertoire avec la commande **pwd**.

6. Le Terminal est maintenant dans le dossier *Temp*, qui est lui-même dans le dossier *TestP13*. Pour faire **revenir le Terminal en arrière, dans le dossier parent**, on utilise toujours la commande **cd**, mais en spécifiant comme répertoire ".." :

```
cd ..
```

Vérifiez ensuite avec la commande **pwd** que le Terminal s'est bien déplacé dans le dossier *TestP13*. Recommencez la commande une seconde fois afin de revenir à votre dossier personnel de départ.

Gestion des fichiers

Nous allons maintenant voir des commandes permettant de copier, déplacer, supprimer ou renommer des fichiers sous Linux.

1. Nous allons commencer par **créer un fichier à l'aide d'un éditeur de texte**. Nous allons démarrer l'éditeur de texte **Gedit** et lui dire de créer un fichier nommé *test.txt*, grâce à la commande

```
gedit test.txt
```

Le programme se lance. Remarquez que votre fenêtre de Terminal reste bloquée : elle le restera jusqu'à ce que le programme Gedit soit fermé. Tapez un peu de texte dans l'éditeur de texte, sauvegardez et fermez Gedit. Vérifiez, de deux façons différentes, le contenu de votre répertoire personnel pour vérifier que le fichier a bien été créé.

2. Nous allons tenter de **déplacer le fichier** dans le dossier *TestP13* précédemment créé. Pour ce faire, on utilise la commande **mv**, qui signifie *MoVe* (déplacer). On spécifie ensuite le nom du fichier à déplacer, et la destination :

```
mv test.txt TestP13
```

Cette commande est l'équivalent, sous Windows, d'un couper/coller. Vérifiez avec l'explorateur de fichier que le fichier *test.txt* a bien été déplacé.

3. Essayez, avec les commandes apprises précédemment, de déplacer le fichier *test.txt* dans le dossier *Temp*. Une fois que vous avez réussi (vérifiez avec l'explorateur de fichiers que c'est le cas), revenez à votre répertoire personnel.

4. Pour **renommer un fichier**, on utilise aussi la commande **mv**, en spécifiant cette fois-ci le nom du fichier à renommer et le nouveau nom. Pour renommer le fichier test.txt en test2.txt, on fait

```
mv test.txt test2.txt
```

Vérifiez avec l'explorateur de fichiers que cette commande a bien renommé votre fichier.

5. Pour **copier un fichier**, on utilise la commande **cp**, qui signifie *CoPy*. On spécifie ensuite le nom du fichier à copier, et le nom de la copie. Pour copier le fichier *test2.txt* dans un fichier *test3.txt*, on fait

```
cp test2.txt test3.txt
```

Vérifiez avec l'explorateur de fichiers, ainsi qu'avec la commande **ls**, que le fichier a bien été copié. Ouvrez le fichier *test3.txt* avec Gedit afin de vérifier que le contenu est identique au fichier de départ :

```
gedit test3.txt
```

Refermez le programme Gedit.

6. Pour **supprimer un fichier**, on utilise la commande **rm**, qui signifie *ReMove* (supprimer). Pour supprimer le fichier test3.txt, on fait

```
rm test3.txt
```

Vérifiez avec l'explorateur de fichiers ainsi que la commande **ls** que le fichier a bien été supprimé.

7. Pour **supprimer un dossier**, la commande est presque la même. Pour supprimer le dossier TestP13, on fait

```
rm -r TestP13
```

Vérifiez avec l'explorateur de fichiers ainsi que la commande **ls** que le dossier a bien été supprimé.

L'autocomplétion

Ecrire des commandes avec le Terminal peut parfois être long : il faut taper les bons noms de fichier sans se tromper, sinon la commande échouera. Heureusement, un système existe afin d'accélérer la saisie des noms de fichiers ou de dossiers : il s'agit de l'autocomplétion.

1. Commencez en créant un dossier avec un nom assez long et compliqué (une quinzaine de lettres entrées au hasard), en faisant par exemple

```
mkdir fehuihvreiuzaoidapvnjve
```

Attention : vous n'avez pas besoin d'entrer le même nom de fichier que sur l'exemple ; choisissez votre propre nom de dossier en tapant des lettres au hasard. Vérifiez que le dossier a bien été créé.

2. Pour entrer dans le dossier avec le Terminal, c'est un peu compliqué ! Il faut utiliser la commande **cd** et recopier après exactement le bon nom de dossier :

```
cd fehuihvreiuzaoidapvnjve
```

Cependant, une technique plus simple existe. Il suffit de saisir seulement quelques premières lettres du dossier :

```
cd feh
```

puis d'appuyer sur la touche Tabulation (la touche à gauche du clavier, avec une flèche pointant vers la gauche et une flèche pointant vers la droite) : le reste du nom du dossier apparait et il vous suffit alors d'appuyer sur la touche Entrée pour valider la commande.

L'autocomplétion permet, grâce à la touche Tabulation (aussi appelée Tab), de compléter automatiquement une commande avec un nom de fichier dont on ne tape que le début. Elle peut être utilisée avec n'importe laquelle des commandes vues précédemment. N'hésitez pas à l'utiliser le plus souvent possible.

3. Maintenant, revenez à votre dossier personnel et utilisez l'autcomplétion pour supprimer le dossier précédemement créé. Faîtes de même pour supprimer le fichier *test2.txt* : même lorsque le nom du fichier est assez court, on peut utiliser l'autocomplétion pour aller plus vite.

4. Pour terminer, sachez qu'en utilisant les flèches Haut et Bas de votre clavier, vous pouvez retrouver l'historique des commandes précédemment entrées sur le Terminal.

2 Blockly : une introduction à la programmation

La programmation consiste en général à faire résoudre un problème à un ordinateur : les problèmes peuvent être variés (trouver le chemin le plus court entre deux endroits, détecter des visgaes sur une image, ...) mais les opérations que peut effectuer l'ordinateur sont assez limitées (additionner des nombres, les soustraire, comparer des valeurs, ...). La programmation consiste à trouver comment combiner ces opérations afin de faire effectuer une tâche complexe à la machine.

Tout le long de ce premier semestre, vous verrez le langage C. Cependant, pour commencer, nous allons regarder un langage visuel, nommé Blockly, consistant à assembler des blocs : rendez-vous à l'adresse `https://blockly-demo.appspot.com/static/apps/index.html`. Vérifiez, une fois sur la page, que le langage Français est bien sélectionné en haut à droite.

1. Pour commencer, réalisez le défi "Puzzle". Ce dernier est très simple, et vous permettra de vous familiariser avec la façon d'assembler des blocs sur l'interface.

2. Réalisez maintenant le défi "Labyrinthe" : ce dernier est en 10 étapes. A chacune des étapes, il faudra assembler les blocs de façon à faire se déplacer Pegman (le petit bonhomme jaune) vers le point rouge. Attention, dans certains défis, vous ne pourrez vous servir que d'un nombre limité de blocs. Dans tous les cas, il ne faut pas que Pegman heurte un mur avant d'atteindre sa destination.