

TP n°1 - Découverte de bash

Le but de ce TP est d'apprendre à se servir de la ligne de commande linux. Bien sûr vous pouvez utiliser la souris et l'interface pour la majorité des choses qu'on va voir, mais c'est bien plus rapide au clavier.

Généralités

- Pour copier-coller dans le terminal, utiliser  +  +  et  +  + . En dehors du terminal, utilisez  +  et  + .
 - Vous pouvez utiliser  (la touche avec les flèches) pour auto-compléter la commande ou le nom de dossier/fichier que vous êtes en train d'écrire. Si une seule commande commence par les lettres que vous avez écrites, la commande en entier s'écrit à l'écran. Sinon appuyez deux fois et vous verrez toutes les possibilités commençant par ces lettres.
 - En appuyant sur , vous pouvez revenir aux commandes précédentes, pour les réutiliser.
 - Pour forcer la fin d'une commande dans le terminal, utiliser  + .
 - Ne mettez pas d'espaces ou d'accents dans vos noms de fichiers ou de dossiers
1. Démarrer un terminal (ou console) en appuyant sur  +  + 
 2. Utiliser la commande `uname -a`. `uname` est une commande et `-a` une option. Des informations s'affichent, en reconnaissez-vous certaines ?
 3. Utiliser la commande `man uname` pour savoir ce que faisait la commande précédente. À tout moment vous pouvez utiliser `man` plus le nom d'une commande pour savoir ce qu'elle fait et connaître ses options.
 4. Vous pouvez démarrer n'importe quel logiciel à partir de la ligne de commande (à condition de connaître la commande associée). Essayez `firefox` ou `libreoffice`.

1 Navigation dans l'arborescence

Pour naviguer dans les fichiers, on utilise :

- `cd`, pour ouvrir un dossier, ou un chemin (par ex. `cd Mon/Chemin/Vers/Mon/Dossier`).
 - Chaque dossier possède un nom et il y a plusieurs noms spéciaux : `~` est le dossier personnel, `\` est la racine du système (aucun dossier ne la contient), `.` (un point) est le dossier actuel et `..` (deux points) est le dossier parent du dossier actuel.
 - `ls`, pour voir ce que contient le dossier actuel. La variante `ls -l` permet de voir toutes les informations sur les fichiers/dossiers.
5. Tester la commande `ls` pour voir ce que contient le répertoire actuel. Normalement lorsque vous démarrez la console elle se place dans votre dossier personnel, aussi appelé `~`. Ce dossier est compris dans et contient d'autres dossiers.
 6. Déplacez vous dans vos documents avec `cd`.
 7. Revenez en arrière avec `cd ..`. Revenez encore en arrière. Faites `ls` pour voir ; vous vous trouvez là où sont rangés tous les dossiers personnels des utilisateurs du labo info.
 8. Essayez d'aller dans le dossier personnel de votre voisin de table.
 9. Pour finir, écrivez `cd /`, exécutez et regardez le contenu avec `ls`. Vous vous trouvez au point le plus "haut" de l'arborescence. C'est ici que sont stockés les fichiers qui font fonctionner linux.

2 Modifier l'arborescence des fichiers

- `mkdir` permet de créer un dossier ("make a directory"). Ex : `mkdir dossier`
 - `touch` permet de créer un fichier vide (ce n'est pas son utilisation première). Pour indiquer quel type de fichier c'est, on utilise les extensions. Ex : `touch fichier.txt` crée un fichier de texte brut.
 - On peut modifier les permissions des dossiers et des fichiers avec `chmod`.
10. Retournez dans votre dossier personnel, parce que vous n'avez pas le droit de créer de dossier ailleurs. Allez dans vos documents puis créez un dossier `tp01`.
 11. Allez dans ce dossier, puis créez un fichier `test.c` (code C) et un fichier `test.ml` (code Caml).
 12. Affichez toutes les informations sur ces fichiers avec `ls -l`.

Lire les permissions

Vous vous trouvez devant des lignes de la forme :

```
prof@hazel:~/Documents/tp01$ ls -l
total 36
-rw----- 1 prof prof    4  1 sept. 16:41 fichier_secret.txt
-rw-rw-r-- 1 prof prof   21  1 sept. 16:41 hello_world.py
drwxr-xr-x 2 prof prof 4096  1 sept. 16:40 Le_Dossier
-rw-r--r-- 1 prof prof   54  1 sept. 16:39 test.c
-rwxr-xr-x 1 prof prof 15832  1 sept. 16:39 test.c.out
-rw-r--r-- 1 prof prof    82  1 sept. 16:43 test.ml
```

Le premier groupe de lettres se réfère aux permissions. Il est constitué de 10 caractères. Le premier est **-** pour un fichier, **d** pour un dossier. Les autres sont en trois groupes de trois : les trois premiers sont les permissions du propriétaire du fichier, les trois suivants sont les permissions du groupe du propriétaire (par exemple le groupe des élèves) et les trois derniers sont les permissions des autres.

Les trois lettres de chaque groupe correspondent respectivement au droit de lire **r**, d'écrire **w** et d'exécuter **x**. Si la lettre est présente, le droit est présent, sinon il y a un **-**.

13. Qu'avez vous le droit de faire sur vos fichiers ? À quel groupe appartenez-vous ? Quels droits ont les autres utilisateurs ?

Modifier les permissions

On peut modifier les 3 groupes séparément : **u** (comme user) désigne les permissions du propriétaire, **g** (comme group) celles du groupe, **o** (comme other) celles des autres et **a** (comme all) désigne les permissions de tout le monde.

Il faut utiliser la commande **chmod**, suivie d'une ou plusieurs lettres pour indiquer qui, puis **+**, **-**, **=** pour indiquer si on ajoute, on retire ou on fixe les permissions et enfin **r**, **w**, **x** pour indiquer quelles permissions sont changées. On termine en indiquant le(s) fichier(s), le(s) dossier(s) affectés par le changement.

Par exemple **chmod u=wx fichier.odt** change les permissions du propriétaire à "pas de lecture, écriture possible, exécution possible".

14. Que fait la commande suivante : **chmod u+x,g+w fichier.c** ? (la virgule sert à séparer deux modifications à faire)
15. Que fait la commande suivante : **chmod ug-x fichier.ml** ?
16. Modifiez les permissions de vos fichiers **test.c** et **test.ml** pour que tout le monde puisse les lire et écrire dedans.

Remarque : votre dossier personnel est normalement protégé contre l'accès par un autre utilisateur, sauf les administrateurs.

3 Environnement de programmation VScode

Cette année je vous conseille d'utiliser l'environnement de code VScode (pour C au moins, on verra plus tard pour Ocaml).

17. Ouvrir Vscode en utilisant la commande **code**. Ouvrir le fichier **test.c** en utilisant les menus.
18. Écrire **#include <stdio.h>** puis **int main(){printf("hello world\n");}** dans le fichier. VScode doit reconnaître qu'il s'agit d'un fichier C : des couleurs devraient apparaître.
19. Pour compiler le code, on a besoin d'un terminal. On peut ouvrir un terminal dans VScode en utilisant le menu ou le raccourci :  + .
20. Compiler le code c en écrivant **gcc -o test.out test.c** dans le terminal. Exécuter le code avec **./test.out**, toujours dans le terminal.

Si vous avez fini, vous pouvez commencer le TP suivant : Introduction au C.