

TD - DÉCOUVERTE DE LINUX

1. Préliminaires

- Installer WSL.

Dans la barre de recherche de windows, taper « panneau de configuration » et sélectionner « Programmes et fonctionnalités ».

Dans la partie gauche de la fenêtre : « Activer ou désactiver des fonctionnalités Windows ». Cocher « Sous-système Windows pour Linux ». Il est possible qu'il vous soit demandé une confirmation.

La « plateforme de machine virtuelle » doit également être activée.

- Télécharger Ubuntu sur le Windows Store.
- Puis initialiser la distribution. Pour cela, lancer Ubuntu, le premier lancement prend un peu de temps. Créer un compte en donnant nom et mot de passe.

Je vous conseille de mettre un mot de passe très simple, du style « linux », car vous risquez de ne pas vous en servir très souvent. Ce conseil

Remarque 0 : en cas d'erreur, voir <https://learn.microsoft.com/fr-fr/windows/wsl/troubleshooting> (détails ci après pour le problème particulier du passage de Windows 10 à Windows 11)

Remarque 1 : si vous êtes passés de Windows 10 à à Windows 11, il peut y avoir une erreur lors de l'installation (erreur 0x80070003 ou l'erreur 0x80370102), il faut activer la virtualisation dans le BIOS. Vous pouvez essayer ce lien : https://support.hp.com/fr-fr/document/ish_5637148-5698278-16 pour les ordinateurs HP. Sinon sur un moteur de recherche, tapez « activer virtualisation bios marque_ordinateur » ou mieux, en anglais « activate bios virtualisation ». Vous pouvez aussi consulter <https://support.microsoft.com/fr-fr/windows/activer-la-virtualisation-sur-windows-11-pc-c5578302-6e43-4b4b-a449-8ced115f58e1>.

Remarque 2 : Erreur 0x800701bc. Dans l'invite de commande Windows (taper cmd dans la barre de recherche puis démarrer en mode administrateur), taper `wsl --update`

Remarque 3 : WSL donne souvent lieu à des erreurs difficiles à corriger lors du lancement, qui devient alors impossible. Dans ce cas, vous pouvez faire une bonne partie de ce TD (partie 2 ci-dessous jusqu'à « Supprimer avec la commande : `rm mon_fichier3.txt` », questions 3a et 3b, et tester les solutions de vos exercices) de ce TD sur un simulateur en ligne. Par contre vous ne pourrez pas faire les trucs rigolos de ce TD !

Trois possibilités parmi d'autres :

- <https://bellard.org/jslinux/> puis Fedora 33 Console (startup link)
- <http://gervaisprof.free.fr/weblinux/> (démarrage un peu lent)
- <https://www.webminal.org/> gratuit le 1^{er} mois. L'inscription avec un compte permet de conserver son travail.

2. Premières commandes

Remarques :

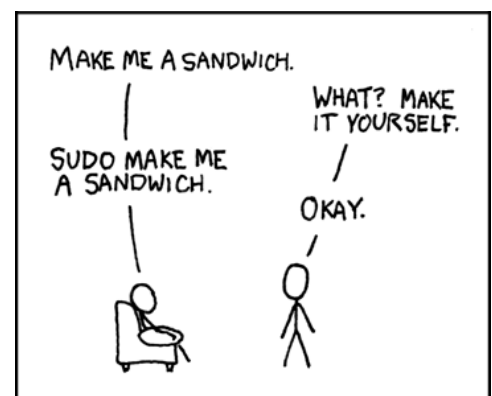
- chaque fois qu'une commande est refusée/ne semble rien faire, rajouter `sudo` devant et rentrer éventuellement son mot de passe. Cela signifie substitute user do, et non super utilisateur do, ce dont on a l'impression ceci dit.

Exemple de l'efficacité de `sudo` (tiré de XKCD) :

- si vous êtes bloqués sur un écran, essayez `ctrl-c` ou `q`. Si ça ne marche pas, fermez la fenêtre...
- toutes les commandes sont des abréviations faciles à retenir, cf. le cours.

- les commandes s'écrivent sous la forme :

`nom_de_la commande -option1 -option2 ... paramètre1 paramètre2 ...` : des espaces entre les options/paramètres, qui peuvent être dans un ordre différent de ce modèle, le tiret est le signe « moins ».



Exécuter les commandes suivantes

Tester la commande « list », qui donne le contenu d'un répertoire : `ls` puis `ls -a` et `ls -l` et enfin `ls -a -l` `ls -al`

Créer un dossier avec la commande « make directory » : `mkdir mon_dossier`

Tester à nouveau `ls`, `ls -a`, `ls -l`, `ls -a -l`

Lire la partie du cours sur les permissions et le détail de `ls -l`.

Aller dans le dossier avec la commande « change directory » : `cd mon_dossier`

Créer un fichier avec la commande « touch » : `touch mon_fichier.txt`

Éditer le fichier avec la commande : `vim mon_fichier.txt` (vim est un éditeur de texte)

Dans vim :

- Aide de vim si nécessaire touche `escape/échap` : `help` (les : font partie de la commande)
- Rentrer du texte touche `insertion` puis écrire
- Quitter avec `escape` puis pour sauver : `x`

Afficher le contenu du fichier avec une des commande « more » ou « less » commande : `more mon_fichier.txt`

Changer les autorisations de `mon_fichier.txt`, commande « change mode » :

- lire la partie du cours sur les autorisations/permissions !
- puis exécuter, après l'avoir comprise, la commande :
`sudo chmod u-w mon_fichier.txt`

Éditer à nouveau le fichier avec `vim mon_fichier.txt`

- Rentrer du texte touche `insertion` puis écrire, que constatez-vous ?
- Quitter avec `escape` puis, sans sauver avec `:q` , en sauvant `:x` , et finalement en force avec « `:q!` »

Copier le fichier avec la commande « copy » : `cp mon_fichier.txt mon_fichier2.txt`

Changer le nom avec la commande « move » : `mv mon_fichier2.txt mon_fichier3.txt`

Supprimer avec la commande : `rm mon_fichier3.txt` (que constatez-vous ?)

Aller dans son dossier Documents étape par étape, vous pouvez faire des `ls -a` pour explorer le contenu des dossiers. Commande « change directory » :

`cd /` `cd mnt/` `cd c/` puis pour aller plus vite `cd Users/fmandon/Documents/`

Vous aurez bien sûr habilement remplacé `fmandon` par votre nom d'utilisateur Windows.

On peut faire ce changement de répertoire en une seule ligne :

`cd /mnt/c/Users/fmandon/Documents/`

racine de
l'ordinateur

Un disque dur branché n'est pas forcément lisible : il faut qu'il soit en plus monté. Ici c'est le disque `c` qui est monté (d'où `mnt`), c'est le disque dur interne de votre ordinateur

Essayer `cd ..` (remonter au répertoire parent «moyen mnémotechnique .. = papa »)

Essayer `cd ~` et `cd /`. Où arrive-t-on dans chaque cas ?

Retourner directement dans votre dossier Documents de Windows

Utiliser la commande `pwd` pour savoir où on est («print working directory »)

Recopier un fichier du dossier documents vers `mon_dossier`, commande :

`cp mon_fichier.suffixe ~/mon_dossier/mon_fichier.usffixe`

`cd /` revient à la racine du disque puis `ls`

Revenir au dossier racine linux `cd~`

Essayer `rmdir mon_dossier` pour supprimer ce qu'on vient de faire (« remove directory »).

Comme ça ne marche pas, alors on utilise la commande « remove » avec l'option récursif `-r` (comme la récurrence, on répète la commande dans tout ce qu'il y a à l'intérieur jusqu'à ce qu'il ne reste plus rien) : `rm -r mon_dossier`

Nota Bene pour la suite :

- on peut revenir aux commandes précédentes sans avoir à les retaper avec les flèches de direction (vers le haut)
- on peut faire des copier-coller dans le shell Linux. En théorie avec shift-ctrl-C et shift-ctrl-V. En pratique pour le coller le clic droit fonctionne nettement mieux.

3. Compléments

a. Les opérateurs de redirection et le « pipe »

Les opérateurs `>` et `>>` permettent d'écrire les sorties d'une commande non pas à l'écran, mais dans un fichier. Placez-vous dans un répertoire non vide et testez `ls -al > liste_fichier.log`. Vous pouvez ensuite ouvrir le fichier créé avec `vim`, ou simplement l'afficher avec `less`. L'opérateur `>` crée le fichier de sortie, éventuellement en écrasant un fichier de même nom préalablement existant. L'opérateur `>>` écrit à la fin du fichier sans l'écraser.

L'opérateur `|` permet d'enchaîner les commandes. Par exemple, `ps -e` (ou `aux`) permet de lister tous les processus (cf. cours de terminale... on va dire que ce sont les programmes) tournant sur le noyau Linux. La commande `wc` permet de compter (word count). `wc -l` compte les lignes (`-c` les octets, `-w` les mots...). Pour compter le nombre de processus, on peut faire `ps aux | wc -l`

b. Aller sur internet.

On n'ouvre pas un navigateur, seuls de rares adresses peuvent être lisibles/accessibles, comme celle-ci pour la météo : `curl wttr.in/montpellier`. La commande `curl` (client url) est puissante, elle permet notamment de récupérer l'en-tête, le code html etc. d'une page (cf. challenges root-me).

c. Installer des programmes

Exemple : installation de `cowsay`

```
sudo apt-get update
sudo apt-get install cowsay
```

Puis test du programme en tapant

```
cowsay hello beau(belle) gosse
```

Options à tester après `cowsay` et avant le texte :

```
-bdgsty (une seule lettre)
-e 88
-T U
-f type_de_vache (par exemple dragon ou vador-koala)
```

Exemple : installation de `fortunes`

```
sudo apt-get install fortunes
```

Puis test de `fortunes` avec la commande `fortune` (parfois très geek et incompréhensible, souvent drôle)

Linux permet d'enchaîner des instructions, en envoyant le résultat d'une commande dans une autre. On appelle cela un tuyau (pipe en anglais). Essayer :

```
fortune /usr/share/games/fortunes | cowsay -n
```

Pour avoir des informations sur son système, le programme `inxi` est très complet (`sudo apt-get install inxi`). Plein d'options possibles.

d. Écrire un script de démarrage.

Éditer son fichier de configuration avec `vim`. Si vous avez installé Ubuntu, il est dans votre répertoire de base et s'appelle `.bashrc`.

Faire `vim .bashrc` et taper en dernière ligne la commande suivante, en faisant attention aux différents types d'apostrophes (c'est une seule ligne de commande, il n'y a pas de passage à la ligne suivante) :

```
cowsay -f `ls /usr/share/cowsay/cows |shuf -n1|cut -d'.' -f1`  
" `fortune` "
```

e. *Splitter (séparer) la fenêtre de Linux en plusieurs sous-fenêtres :*

sudo screen puis :

- ctrl a puis | pour une séparation verticale ;
- ctrl a puis S pour une séparation horizontale ;
- ctrl a puis Q pour une supprimer les séparations ;
- ctrl a puis tab pour passer d'une fenêtre à une autre.
- Pour démarrer une session dans la nouvelle fenêtre, ctrl a puis c.

Utiliser ces commandes pour séparer l'écran en deux, travailler dans une partie et lancer dans l'autre :

```
telnet towel.blinkenlights.nl
```

Telnet est un protocole de communication par texte, abandonné sauf pour faire des tests.

f. *Autres bêtises et amusements :*

On peut rajouter des fichiers de citations dans fortunes :

- sous Windows, exemple compliqué :
 - Télécharger les Chuck Norris facts (par exemple)
 - Deux adresses (on télécharge avec le petit bouton de téléchargement à droite au dessus de la fenêtre avec le texte)
 - <https://github.com/henkealg/ChuckNorrisSlackBot/blob/master/facts.txt>
 - <https://github.com/LudwigThePig/Chuck-Norris-Facts/blob/master/facts.txt>
 - Dans un éditeur de textes, changer chaque retour à la ligne en « retour à la ligne % retour à la ligne ». Ca se fait assez bien sous Notepad++
 - menu Affichage > Symboles Spéciaux > fin de ligne
 - menu rechercher > remplacer
 - mode de recherche étendu
 - Le caractère LF est codé par \n, CR par \r
 - En général, on remplace \n par \n%\n
- Sous Windows, exemple simple :
 - télécharger le fichier citations en français sur www.maths-info-lycee.fr , page 1ère NSI. Le nom du fichier se comporte bizarrement parfois, changez-le éventuellement en francais.txt .
- Sous Ubuntu :
 - aller dans le dossier /usr/share/games/fortunes
 - y copier le fichier provenant des téléchargements Windows (une seule ligne de code) :
 - cp /mnt/c/Users/<votre nom d'utilisateur>/Downloads/<le nom du fichier>/nom_du_fichier_telechargé.extension
nom_du_fichier_telechargé.extension
 - Si la permission vous est refusée, débrouillez-vous !
 - Créer un fichier .dat utilisable par fortunes (ainsi que les autres qui semblent avoir également une utilité)

```
strfile facts.txt chucknorris.dat  
strfile facts.txt chucknorris.u8  
cp facts.txt chucknorris
```
- Sous Ubuntu :
 - retourner dans le dossier d'origine (cd ~) et essayer fortune francais ou fortune chucknorris, suivant ce que vous avez fait.
- Installer un magnifique aquarium : voir <https://doc.ubuntu-fr.org/asciiquarium>.
Des commandes fonctionnelles et simples en 2025 :

```
sudo add-apt-repository ppa:ytdvld/asciiquarium
sudo apt-get update && sudo apt-get install asciiquarium
Puis simplement exécuter : asciiquarium
```

Les commandes classiques, au cas où les précédentes ne fonctionneraient pas : voir <https://doc.ubuntu-fr.org/asciiquarium>.

En cas de problème indiqué avec l'animation, voir cette page :

<https://askubuntu.com/questions/927441/how-do-i-install-asciiquarium/>

à la réponse Install term-animation

- Si vous faites comme de nombreux linuxiens, et que pour lister le contenu d'un répertoire vous tapez souvent `sl` à la place de `ls`, alors essayez :

```
sudo apt-get install sl
```

EXERCICES

Exercice 1.

Créer un dossier *programmes*, et dedans un fichier *bonjour.py*.

Le code du script Python peut être (ce n'est pas une obligation, vous pouvez faire plus sophistiqué si vous le souhaitez) :

```
from time import sleep
nom = input("Quel est votre nom ? ")
print("Bonjour ", nom)
sleep(5)
```

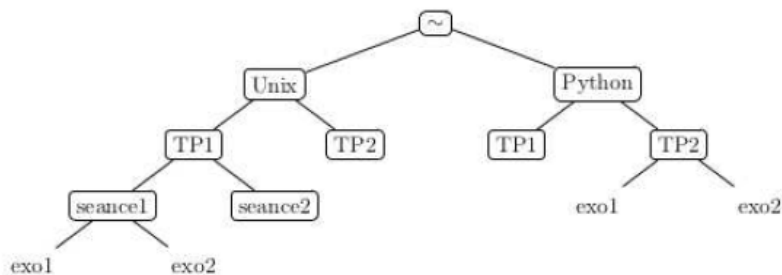
Rendre le fichier exécutable pour vous.

Vérifier quelle est la version de Python dans `/usr/bin`. Vous constaterez qu'un « `ls` » donne trop de résultats. On utilise alors `ls py*` pour trouver tous les fichiers dont le nom commence par `py` (et est suivi de n'importe quoi d'autre).

Lancer le programme avec la commande : `bonne_version_de_python bonjour.py`

Exercice 2.

Créer l'arborescence suivante de dossiers et fichiers dans votre répertoire de travail.



On précisera toutes les commandes utilisées.

A l'aide d'un « pipe », lister à partir du répertoire courant tous les répertoires triés par taille décroissante. On pourra consulter le manuel pour les commandes `du` et `sort`.

Exercice 3.

Dans le répertoire `./Unix/TP2`, créer deux fichiers `public.txt` et `prive.txt`.

Donner les commandes permettant de mettre les permissions demandées, quelles que soient les permissions initiales sur les répertoires ou fichiers.

1. Le répertoire personnel possède tous les droits pour l'utilisateur, et uniquement le droit d'exécution pour le groupe et les autres.
2. Les répertoires `Unix` et `Unix/TP2` possèdent tous les droits pour l'utilisateur et les droits de lecture et d'exécution pour le groupe et les autres

3. Le fichier `public.txt` du répertoire `Unix/TP2` possède les droits de lecture et d'écriture pour l'utilisateur et uniquement les droits de lecture pour le groupe et les autres.
4. Le fichier `prive.txt` du répertoire `Unix/TP2` possède les droits de lecture et d'écriture pour l'utilisateur et aucun droit pour le groupe et les autres.

Exercice 4.

Sans tester ces commandes dans un terminal, dessiner ci-dessous l'arborescence finale des fichiers et des répertoires.

On utilisera `TEST` comme racine de l'arborescence.

1. `mkdir a b c d`
2. `touch a/t.txt d/foo.txt`
3. `cd c`
4. `mkdir ../b/e f g`
5. `cd ..`
6. `cp */*.txt c/g` # le caractère spécial "*" remplace n'importe quelle chaîne de caractère
7. `rm -r f d`

Cours de Frédéric Mandon, licence Creative Commons BY NC SA, <https://creativecommons.org/licenses/by-nc-sa/3.0/fr/>

Merci aux collègues de la liste NSI pour leurs idées et leurs exercices, dont je n'ai pas noté la provenance :-/ .