

TD Shell

L'objectif de ce TD est de découvrir en ligne de code quelques commandes de base sous linux. En attendant que Linux ne soit installé au lycée on va utiliser un émulateur en ligne.

Connectez-vous sur <https://www.offidocs.com/jor1k/demos/main.html?>

```
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
rtc-lpc32xx 99000000.rtc: rtc core: registered rtc-lpc32xx as rtc0
sd 0:0:0:0: [sda] Attached SCSI disk
NET: Registered protocol family 17
9pnet: Installing 9P2000 support
rtc-lpc32xx 99000000.rtc: setting system clock to 2021-05-24 09:26:30 UTC (1621848390)
ALSA device list:
 #0: Dummy 1
VFS: Mounted root (9p filesystem) readonly on device 0:12.
devtmpfs: mounted
Freeing unused kernel memory: 136K (c054e000 - c0570000)
udhcpd: started, v1.26.2
Setting IP address 0.0.0.0 on eth0
udhcpd: sending discover
udhcpd: sending discover
udhcpd: sending discover
udhcpd: no lease, failing
~login[72]: root login on 'ttyS1'
Note, you can exit most programs by pressing CTRL+C
~ $
```



1) Commandes "arborescences des dossiers et fichiers"

mkdir reptest	Crée le répertoire reptest
touch fichier1.txt	Crée le fichier fichier1.txt
ls	Affiche le contenu du repertoire
cd reptest	Change de répertoire pour aller dans le répertoire fils reptest

Exercice 1 :

Ajouter deux répertoires `essai1` et `essai2` et deux fichiers `test1.txt` et `test2.txt` dans `essai2`.

cd ..	Change de répertoire pour aller dans le répertoire père
cd ~	Revient au répertoire personnel
rmdir <code>essai1</code>	Supprime un répertoire vide
rm -r <code>essai2</code>	Supprime un répertoire non vide
cp <code>fichier1.txt</code> <code>fichier2.txt</code>	Copier le fichier1 et crée le fichier2 identique
rm <code>fichier1.txt</code>	Supprime un fichier
mv <code>fichier2.txt</code> <code>test</code>	Deplace <code>fichier2.txt</code> dans le répertoire <code>test</code>

Exercice 2 :

Créer un fichier `texte.txt` et deux répertoires `livre1` et `livre2`

Dupliquer le fichier texte.txt en deux fichiers texte1.txt et texte2.txt
 Déplacer texte1.txt dans le répertoire livre1
 Se déplacer dans chaque répertoire et vérifier que le contenu est correct.
 Effacer les deux répertoires livre1 et livre2

2) Commandes "gestion des droits"

ls -s	Affiche le contenu du répertoire de manière détaillé.
ls -al	Affiche les répertoires et fichiers cachés

```
-rw-r--r--.    1 root      root          9 Mar  7 09:32 .magic_string.txt
drwxrwxr-x.   2 sebbanville sebbanville 40 Mar  7 20:11 test
-rw-rw-r--.   1 sebbanville sebbanville  0 Mar  7 20:09 test2.txt
```

On a de gauche à droite :

Droits / nb liens / Nom propriétaire / Nom groupe / Taille en octet / Date / Heure / Nom fichier

Les systèmes de type "UNIX" sont des systèmes multi-utilisateurs. Plusieurs utilisateurs peuvent donc partager un même ordinateur. Chaque utilisateur possédant un environnement de travail qui lui est propre.

Chaque utilisateur possède certains droits lui permettant d'effectuer certaines opérations et pas d'autres. Le système d'exploitation permet de gérer ces droits très finement.

Un utilisateur un peu particulier est autorisé à modifier tous les droits : ce "super utilisateur" est appelé "administrateur" ou "root".

Au lieu de gérer les utilisateurs un par un, il est possible de créer des groupes d'utilisateurs.

L'administrateur attribue des droits à un groupe au lieu d'attribuer des droits particuliers à chaque utilisateur.

Les fichiers et les répertoires possèdent 3 types de droits :

- les droits en lecture : "r" signifie "lecture autorisée".
- les droits en écriture : "w" signifie "écriture autorisée".
- les droits en exécution : "x" signifie "exécution autorisée" pour un fichier et "accès autorisé" pour un répertoire.
- le caractère "-" à la place d'un des trois précédents interdit le droit en question.

Pour le fichier test2.txt on a : `-rw-rw-r--`

- : c'est un fichier, pour un répertoire on a d
- rw- : Le propriétaire peut lire et écrire sur ce fichier, il ne peut pas l'exécuter.
- rw- : Tous les utilisateurs du groupe peuvent lire et écrire sur ce fichier, mais pas l'exécuter.
- : Toutes les autres utilisateurs seulement peuvent lire le fichier

On peut changer les droits, il suffit de savoir compter en binaire :

rw- correspondra au nombre binaire 111 donc au nombre entier 7

r-- correspondra au nombre binaire 100 donc au nombre entier 4

rw- correspondra au nombre binaire 110 donc au nombre entier 6 ect ...

chmod 777 test2.txt	Je donne tous les droits sur le fichier test2.txt à tous les utilisateurs !
---------------------	---

Avec webminal nous ne sommes pas administrateur donc nous ne pouvons pas :

Changer le propriétaire et le groupe d'un fichier avec `chown`

Ajouter un utilisateur avec `adduser`

Créer un groupe avec `addgroup`

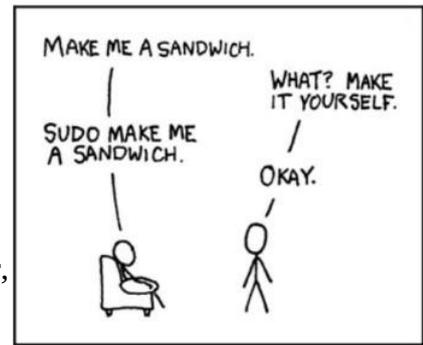
Modifier un mot de passe avec `passwd`

....

Lorsqu'on est connecté en root on a le contrôle de tous.

Par exemple `shutdown -h now` éteint l'ordinateur.

La commande `sudo` signifie « super utilisateur do » permet d'installer, lancer, effacer, ...



3) Commande des processus

La commande `top` permet d'afficher les processus en cours. (q pour sortir)

On constate que le système d'exploitation gère en parallèle plusieurs processus.

C'est l'ordonnanceur qui gère les priorités.

Pour exécuter un fichier python on tape par exemple: `python essai.py`

Si on est connecté en root la commande `kill -9 1234` envoie le signal détruire au processus 1234 ...

4) Commande de fichier

Nous avons vu que la commande `touch` permettait de construire un fichier vide.

On peut réutiliser cette commande pour modifier la date et l'heure (le timestamp) du fichier.

La commande `cat` permet de compléter un fichier et de l'afficher.

<code>cat</code>	Permet de saisir plusieurs lignes au clavier. Ctrl+d pour finir
<code>cat essai.txt</code>	Affiche le contenu du fichier <code>essai.txt</code>
<code>cat > essai.txt</code>	Ecrit ce qui est tapé dans le fichier

Exercice :

Afficher la liste des processus, noter le nombre processus devant `running`, `sleeping`, `stopped`, `zombi`.

Construire un dossier Python et y insérer un fichier `essai.py`.

Dans ce fichier rentrer le programme :

```
a=2
a=a+2
print(a)
```

Exécuter ce programme, (au besoin changer les droits)

Ré-afficher la liste des processus, noter le nombre processus devant `running`, `sleeping`, `stopped`, `zombi`. Comparer .