Ligne de commande Linux

Les systèmes d'exploitation actuels ont une *interface utilisateur graphique* (ou GUI, de l'anglais *Graphical User Interface*) composée de fenêtres, de menus déroulants, d'icônes, de boutons, … Ce type d'interface utilisateur est généralement assez intuitive et simple d'utilisation, permettant l'utilisation du système par des personnes n'ayant aucune connaissance technique. Ce type d'interface date du début des années 70, mais il faudrait attendre la fin des années 80 pour qu'il se démocratise, avec les premiers Macintosh, puis les premières versions de Windows. Avant l'avènement des interfaces graphique, la communication entre l'utilisateur et la machine se faisait en mode texte, en tapant des commandes. L'utilisation d'un système informatique demandait alors plus de connaissances, et était moins facile d'accès. Ce type d'interface existe toujours (sous le nom de *ligne de commande*) car il permet de faire certaines actions impossibles (ou très compliquées) avec l'interface graphique habituelle. Dans la suite de ce document nous parlerons surtout de la ligne de commande Linux (cf. activité précédente sur les systèmes d'exploitation).

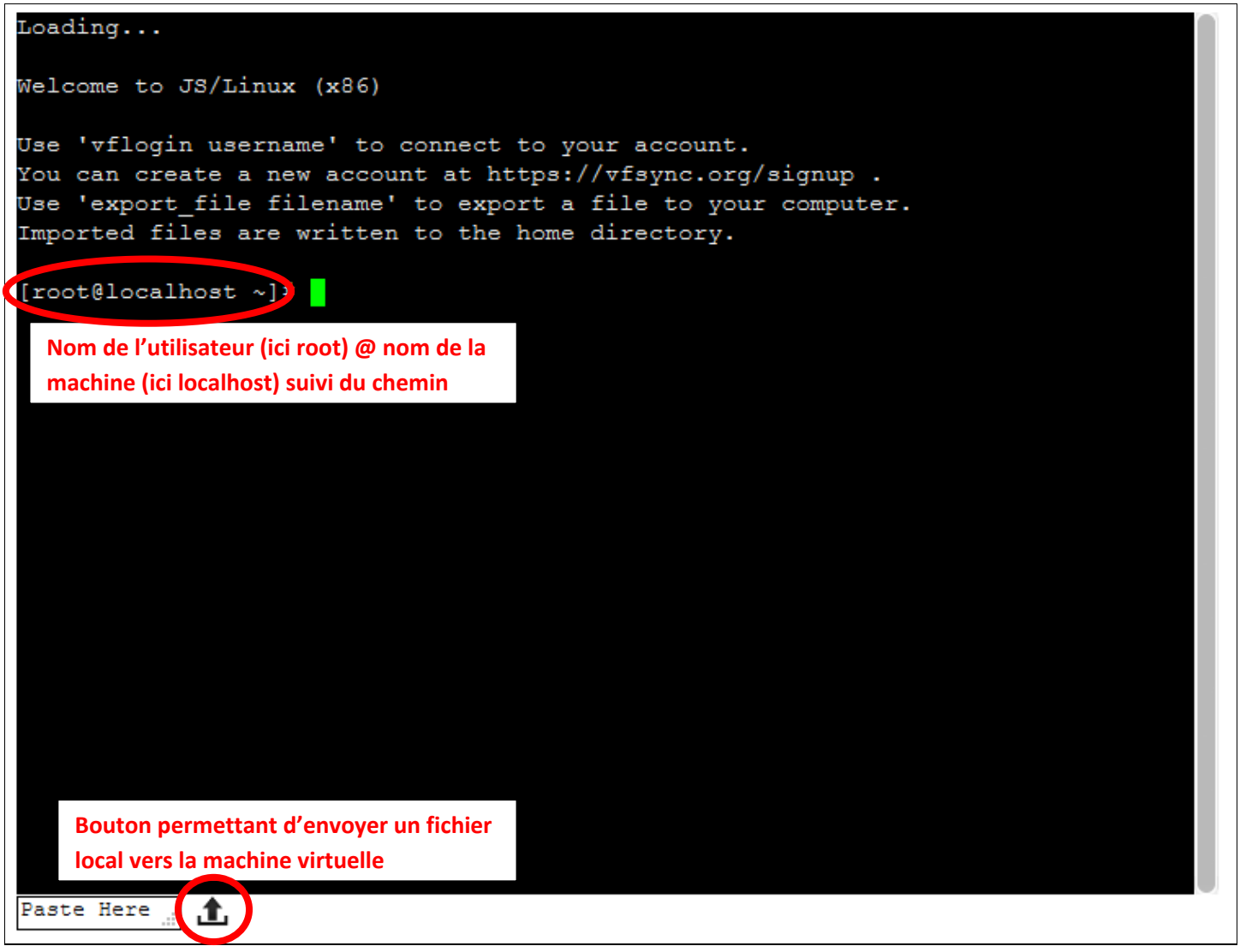
I./ Simulation d'un PC tournant sous Linux :

Tout le monde ne possédant pas un ordinateur tournant sous Linux, nous allons utiliser un simulateur en ligne, programmé en Javascript. Si vous avez un système Linux, vous pourrez évidemment faire directement les manipulations indiquées ici. Une autre méthode est de booter un PC Windows sur un live DVD Linux.

Le simulateur se trouve à l'adresse suivante : https://bellard.org/jslinux/vm.html?url=buildroot-x86.cfg

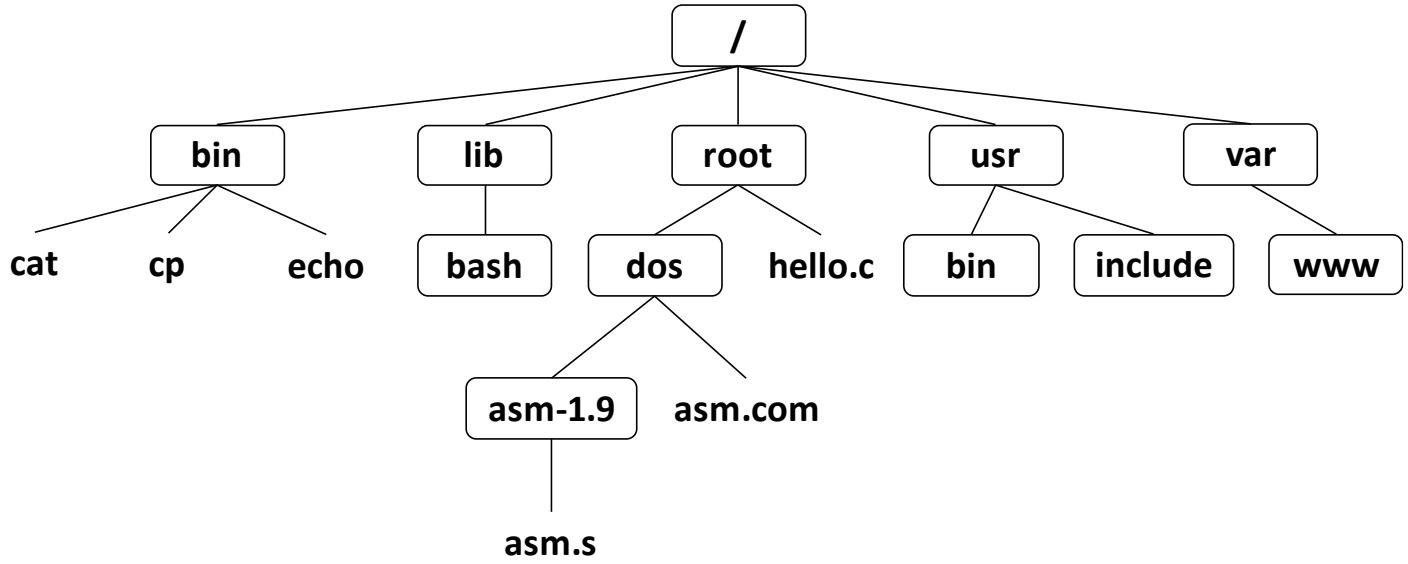
Attention, le chargement peut être plus ou moins long selon la puissance de votre ordinateur (on rappelle qu'un code Javascript inclus dans une page web est exécuté par votre machine).

Après que la machine virtuelle ait démarré, on obtient un écran de ce type :



II./ Système de fichiers :

Nous allons voir comment manipuler les fichiers avec la ligne de commande. Il faut savoir que Linux utilise un *système de fichiers* en *arborescence*. Un *répertoire* est un objet permettant stocker un ou plusieurs *fichiers*. Un répertoire peut aussi contenir un ou plusieurs répertoires qui peuvent à leur tour contenir d'autres répertoires, etc… Le tout premier répertoire est appelée le *répertoire racine*, et il est noté « / ». On peut représenter l'arborescence des fichiers dans un système de type Linux sous la forme :



Les noms encadrés correspondent aux répertoires, ceux qui ne le sont pas correspondent aux fichiers. Le chemin (ou *path*) d'un fichier est l'enchainement de répertoires qu'il faut parcourir pour l'atteindre en partant de la racine. Lorsqu'on écrit le chemin d'un fichier, on sépare les noms des différents répertoires par le caractère « / ».

Dans notre exemple, le chemin du fichier « asm.s » est : « /root/dos/asm-1.9/asm.s ».

Cette première façon d'adresser un fichier s'appelle le *chemin absolu*. Si on se trouve déjà à l'intérieur d'un répertoire, on peut décrire le chemin jusqu'au fichier à partir de ce répertoire, et non en partant de la racine. On parle alors de *chemin relatif*.

Dans notre exemple, si on se trouve dans le répertoire root, le chemin du fichier « asm.s » est : « dos/asm-1.9/asm.s ».

Attention : pour un même fichier, il peut y avoir plusieurs chemins relatifs, selon le répertoire où on se trouve. Le chemin absolu est, par contre, unique, pour un fichier donné.

III./ Quelques commandes utiles :

*1./ Connaître le chemin du répertoire de travail : **pwd** (present working directory)*

Lorsque le simulateur est lancé et que le curseur apparait, taper la commande « pwd » : cela affiche le *répertoire courant*. On obtient « /root ». Cela veut donc dire que nous nous trouvons dans un répertoire nommé « root », se trouvant lui-même dans le répertoire racine « / » du sytème.

*2./ Connaître le contenu du répertoire de travail : **ls** (liste)*

```
[root@localhost ~]# pwd
/root
[root@localhost ~]# ls
dos       hello.c   hello.js
[root@localhost ~]# ls -l
total 12
drwxr-xr-x    3 root     root          163 Aug 20  2011 dos
-rw-r--r--    1 root     root          242 Jul 15  2017 hello.c
-rw-r--r--    1 root     root           63 Jan 29 20:02 hello.js
[root@localhost ~]#
```

La commande « ls » nous permet d'afficher le contenu du répertoire dans lequel nous nous trouvons. Les noms des fichiers peuvent avoir différentes couleurs selon leur nature. Par exemple, le nom « dos » est en bleu : cela indique qu'il s'agit d'un répertoire nommé « dos », qui est susceptible de contenir d'autres fichiers et d'autres répertoires.

Pour avoir plus d'informations, on peut retaper la commande en rajoutant *l'option* « –l » : « ls –l ». Cela affiche la liste des fichiers avec des informations supplémentaires.

Note : le comportement de certaines commandes peut être modifié en utilisant des options.

Par exemple, regardons la première ligne :

- Le premier caractère est un « d », comme directory : il s'agit donc bien d'un répertoire.
- Les trois caractères suivants donnent les droits du propriétaire du fichier (i.e. l'utilisateur qui a créé le fichier). « rwx » veut dire que le propriétaire peut lire « r », écrire (et donc modifier ou effacer) « w » et exécuter (si celui-ci est un exécutable) « x » le fichier. Pour un répertoire, le « x » veut dire que l'on peut y pénétrer.
- Les trois caractères suivants donnent les droits des membres du groupe lié au fichier (c'est un groupe d'utilisateurs que le propriétaire du fichier peut créer). « r–x » veut dire que les membres de ce groupe peuvent lire « r », ne peuvent pas écrire « – », et peuvent exécuter « x » le fichier.
- Les trois caractères suivants donnent les droits des autres utilisateurs. « r–x » a la même signification que précédemment.
- Le nombre qui suit est le nombre de liens du fichier (notion hors programme).
- Le nom qui suit est le nom du propriétaire du fichier, ici « root ».
- Le nom qui suit est le nom du groupe lié au fichier, ici il a le même nom « root ».
- Le nombre suivant est la taille du fichier en octets. (Attention, pour un répertoire, la taille indiquée ne correspond pas à la taille de tous les fichiers compris à l'intérieur de celui-ci).
- Ensuite viennent la date et l'heure de création du fichier.
- Puis finalement son nom.

*3./ Se déplacer dans l'arborescence : **cd** (change directory)*

Taper « cd / » puis faire « ls ». Que remarque-t-on ? Taper « pwd ». Le résultat était-il prévisible ?

Avec cette commande « cd », on peut se déplacer de manière absolue : taper « cd /root/dos/asm-1.9/ », puis faire « ls », puis « pwd ».

Cette manière de procéder implique que l'on connaisse parfaitement la structure de l'arborescence, ce qui n'est pas toujours le cas. On utilise alors un mode de déplacement relatif.

Taper « cd / » pour retourner dans la racine. Faire « pwd » pour vérifier que nous y sommes bien. Puis faire cd « root », puis « pwd » et « ls ». Faire cd « dos », puis « pwd » et « ls ». Faire « cd asm-1.9 », puis « pwd » et « ls ». On peut donc avancer pas à pas dans les répertoires.

Faire maintenant « cd .. », puis « pwd ». La commande « cd .. » permet de remonter au *répertoire parent*, c'est-à-dire au répertoire qui contient le répertoire courant. Remonter ainsi jusqu'au répertoire racine.

*4./ Créer un nouveau répertoire : **mkdir** (make directory)*

Faire « cd /root », puis taper « mkdir bidon ». Faire « ls ». Que remarque-t-on ? Placer vous dans le répertoire « bidon » nouvellement créé puis vérifier que celui-ci est vide.

*5./ Supprimer un répertoire : **rmdir** (remove directory)*

Taper « cd .. » pour retourner dans le répertoire parent, puis faire « ls ». Taper ensuite « rmdir bidon », puis refaire « ls ». Que remarque-t-on ? La commande « rmdir » permet de supprimer un répertoire, à condition que celui-ci soit totalement vide (c'est une sécurité qui permet d'éviter d'effacer une grande quantité de fichiers accidentellement).

*6./ Créer un nouveau fichier vide : **touch***

Faire « touch toto.txt », puis faire « ls –l ». Vérifier qu'un nouveau fichier de taille 0 octet et nommé « toto.txt » a été créé. Vérifier sa date et son heure de création.

*7./ Effacer un fichier : **rm** (remove)*

Faire « rm toto.txt », puis faire « ls ». Que remarque-t-on ?

*8./ Afficher le contenu d'un fichier : **cat***

Se placer dans le répertoire « /root » si ce n'est pas déjà fait. Puis faire « cat hello.c ». Se placer ensuite dans le répertoire « /root/dos », puis faire « cat asm.com ». La commande « cat » affiche le contenu d'un fichier sous forme d'une suite de caractères ASCII. Si le fichier n'est pas un fichier texte mais un fichier exécutable, le résultat ne sera pas lisible par un humain.

*9./ Copier un fichier : **cp** (copy)*

Faire « cd /usr/local », puis taper « pwd », puis« ls ». Noter le nom des fichiers présents dans ce répertoire. Ensuite, faire « cp /root/hello.c /usr/local », puis « ls ». Que remarque-t-on ? La commande « cp » permet de copier un fichier dont on donne la source, puis la destination. Il s'agit d'une copie, qui n'efface pas l'original. Vérifier que le fichier « hello.c » est toujours présent dans « /root ».

*10./ Déplacer ou renommer un fichier : **mv** (move)*

Faire « cd /root », puis « mv hello.c /var ». Faire ensuite « ls ». Où est passé le fichier « hello.c » ? Vérifier votre hypothèse. La commande « mv » permet ainsi de déplacer un fichier, qui disparait ainsi du répertoire source pour se retrouver dans le répertoire de destination. On peut aussi se servir de cette commande pour renommer un fichier. Faire « touch test », puis « ls ». Puis faire « mv test essai », puis « ls ».

IV./ Pour aller plus loin :

Dans le III./2./, nous avons parlé des droits d'accès des fichiers. Lorsqu'on est propriétaire d'un fichier, on peut changer ces droits de façon à limiter ou augmenter les permissions accordées à un utilisateur. Pour cela, il existe une commande particulière « chmod ». Vous trouverez des informations sur cette page : https://doc.ubuntu-fr.org/permissions. Une fois cette page étudiée, faire le travail suivant :

- Dans le répertoire « /root », créer un répertoire appelé « test_NSI »
- Se placer dans le répertoire « test_NSI », puis créer un fichier vide appelé « essai.txt »
- A l'aide de la commande « chmod », modifier les permissions de ce fichier afin que les utilisateurs puissent écrire « w » dans ce fichier
- Vérifier que les permissions ont été modifiées avec la commande « ls –l ».

V./ Ligne de commande Windows :

Windows possède aussi sa ligne de commande. Pour éviter de faire des mauvaises manipulations sur votre PC personnel, vous pouvez utiliser un autre simulateur se trouvant à l'adresse suivante :
https://bellard.org/jslinux/vm.html?url=freedos.cfg&mem=64&graphic=1&w=720&h=400

Attention : le simulateur démarre en clavier QWERTY. Taper la commande « keyb fr » afin de passer en AZERTY.

Commande LINUX	Commande DOS	Commande Linux	Commande DOS
pwd	cd	rmdir	rd
cd chemin	cd chemin	rm	del
cd ..	cd ..	cat	type
ls	dir	cp	copy
mkdir	md	mv	move

Remarques :

- sous Windows, le répertoire racine s'appelle « \ » et non pas « / ».
- de la même façon, le caractère séparant les répertoires est « \ » et non pas « / ».
- sous Windows, chaque lecteur (disque dur, CD, clef USB, lecteur réseau, …) a une lettre qui se place avant le chemin. Par exemple, le disque dur principal est « C : ». (Les lettres « A : » et « B : » ne sont plus utilisées, et correspondaient aux lecteurs de disquettes, technologie totalement obsolète !).

2025
Bruno PEYROT