

BUT INFO1 - Accueil septembre 2023


Ceci une première approche d'un interpréteur de commande. La version originelle de ce document a été rédigée par L. Jezequel.

1 Interpréteur de commandes

Pour contrôler un ordinateur, le moyen de plus direct est d'utiliser la *ligne de commande*. Cela consiste à écrire dans un *langage de commande* des séquences de mots (ou *commandes*) qui donnent des instructions à la machine. Ces commandes sont rédigées dans un *interpréteur de commandes* : un programme qui transcrit ces commandes à la machine. Il existe sous Linux plusieurs interpréteurs de commandes, nous utiliserons l'un d'entre eux : *bash*.

Question. Ouvrez l'interpréteur de commandes.

Une commande se décompose généralement en trois parties : en premier le nom de la commande, puis des paramètres et/ou des options. Par exemple la commande `man` (*manuel*) donne des informations sur l'utilisation des commandes.

Question. Tapez `man man` dans l'interpréteur de commandes, puis appuyez sur la touche entrée () afin d'utiliser la commande `man` pour connaître son propre fonctionnement. Lisez la section DESCRIPTION du texte qui s'affiche puis essayez de comprendre comment s'utilise la commande `man` à l'aide de la section SYNOPSIS.

L'objectif des séances de cet accueil est d'appréhender le fonctionnement de l'interpréteur de commandes ainsi que de découvrir quelques commandes de base permettant notamment de naviguer parmi les fichiers et les dossiers et de manipuler ceux-ci. Ceci sera utile pour les ressources de développement.

2 Système de gestion de fichiers

Les fichiers et les dossiers (ou *répertoires*) sont organisés sous Linux selon une arborescence : la racine (`/`) est un dossier qui contient tout le reste, les nœuds sont les autres dossiers (qui contiennent d'autres dossiers ou des fichiers), les feuilles sont des fichiers ou des dossiers (vides). La figure 1 montre un exemple d'organisation de dossiers et de fichiers.

Cette organisation des fichiers et des dossiers ressemble à ce qui existe sous Windows. La principale différence est qu'ici tous les éléments (même ceux situés sur une clé usb par exemple) sont organisés au sein d'une unique arborescence (on ne retrouve pas les C:, D:, etc de Windows).

Vous avez pu remarquer, dans l'interpréteur de commandes, que chaque ligne sur laquelle vous pouvez taper une commande commence de cette façon : `utilisateur@machine:chemin$` où `utilisateur` indique votre nom d'utilisateur (celui que vous devez rentrer pour vous connecter à la machine) et `machine` indique le nom de l'ordinateur. Ce début de ligne s'appelle *l'invite de commande*. La partie `chemin` vous indique votre position actuelle dans l'arborescence des fichiers et dossiers, avec pour convention que `~` remplace `/home/utilisateur` (qui est votre *répertoire personnel*). Cette position est le *répertoire courant*.

Question. Vérifiez que vous vous situez bien dans votre répertoire personnel.

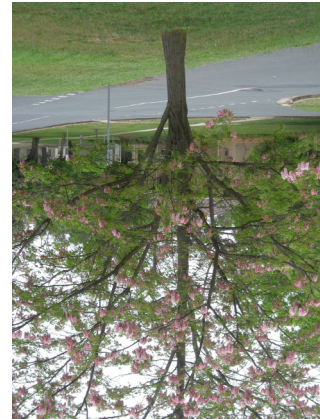
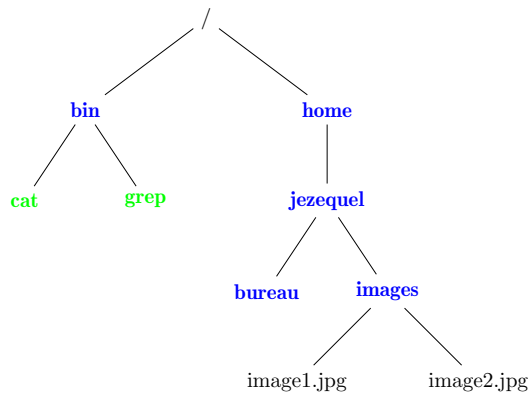



Fig. 1: À gauche, un morceau d'arborescence de fichiers, **bin** est un nœud (un dossier), **bureau** est une feuille et un dossier vide, **image1.jpg** est une feuille et un fichier. À droite un arbre.

3 Lister les fichiers

La commande `ls` (*list segments*) permet de lister les fichiers et dossiers d'un répertoire.

Question. Tapez `ls` puis entrée () dans l'interpréteur de commandes. Observez la sortie obtenue. Quels éléments ont été listés ?

Question. Utilisez maintenant la commande `man` pour obtenir de l'aide sur `ls`. Comment lister les fichiers avec leur taille ? Comment faire pour que cette taille soit représentée de façon lisible simplement par un humain (i.e avec des unités type K, M, etc) ?


Question. D'après l'aide, que liste la commande `ls -a` ? Comment cela se reflète-t-il dans l'interpréteur de commandes ?

4 Se déplacer

La commande `cd` (*change directory*) permet de se déplacer parmi les répertoires.

4.1 Chemins absolus


On appelle *chemin absolu* un chemin dans l'arborescence des fichiers qui part de la racine. Par exemple, `/home/jezequel/images` est le chemin absolu du dossier **images** de la figure 1.

Question. Dans l'interpréteur de commandes, tapez `cd /etc` puis entrée (). D'après l'invite de commandes, dans quel répertoire vous trouvez vous maintenant ?

Question. Placez-vous maintenant dans le répertoire `/bin`. Retournez ensuite à votre répertoire personnel.

4.2 Chemins relatifs

Il peut parfois être laborieux de rentrer un chemin absolu. On peut l'éviter en utilisant des *chemins relatifs*, c'est-à-dire des chemins exprimés depuis le répertoire courant. Ainsi, `../..`/`etc` est un chemin relatif du dossier `etc` de la figure 1 lorsque le répertoire courant est `jezequel` (c'est-à-dire le répertoire personnel de l'utilisateur `jezequel`). Le double point `..` permet de remonter d'un répertoire dans l'arborescence.

Question. Dans l'interpréteur de commandes, tapez `cd ..` puis entrée (). D'après l'invite de commandes, dans quel répertoire vous trouvez vous maintenant ?


Question. En utilisant des chemins relatifs, placez-vous maintenant dans le répertoire `/bin` puis retournez à votre répertoire personnel.


5 Ne pas se perdre

Question. Essayez la commande `cd ~`. Quel est maintenant le répertoire courant ?


La commande `pwd` (*print working directory*) permet d'afficher le chemin complet du répertoire courant.

Question. Essayez la commande `pwd`.

Si on n'est pas certain du nom d'un répertoire, d'un fichier, ou d'une commande (ou simplement pour gagner du temps) on peut utiliser à n'importe quel moment la touche `tab` () pour obtenir une liste des noms de commandes (si on est en train de taper le premier mot d'une commande) ou des dossiers/fichiers (sinon) qui commencent par les caractères déjà tapés.

Question. Tapez `ma` puis `tab` (). Quelles sont les commandes qui commencent par `ma` ?

Question. Tapez `cd`, puis espace (), puis (). Qu'observez-vous ?

Quand la liste obtenue par la touche `tab` () ne comprend qu'un seul élément, celui ci est automatiquement ajouté à la commande en cours.

Question. Tapez `cd ../uti` en remplaçant `uti` par les premières lettres de votre nom d'utilisateur pour essayer cette fonctionnalité.

Lorsque l'on utilise la touche haut () dans l'interpréteur de commandes on a accès aux commandes précédemment utilisées.

Question. Testez cette fonctionnalité en réutilisant quelques unes des commandes du début de ce sujet de TP.

6 Afficher du texte

La commande `echo` permet d'afficher du texte à l'écran.

Question. Consultez la page de manuel de la commande `echo`.

Question. Testez cette commande en affichant quelques lignes de texte.

7 Lecture de fichiers

La commande `cat` (*concatenate*) permet de lire des fichiers et de les afficher dans l'interpréteur de commandes l'un à la suite de l'autre (ce qu'on appelle concaténer).

Question. Sur madoc, récupérez `fichierTexte.txt`. Affichez le contenu de ce fichier.

Question. Affichez deux fois à la suite le contenu du fichier `fichierTexte.txt` en numérotant les lignes.

Attention, la commande `cat` ne décode pas les fichiers, elle n'est pas capable d'afficher en clair un fichier pdf par exemple.

Question. Pour vous en convaincre, récupérez (toujours sur madoc) `fichierBinaire.pdf` affichez le par la commande `cat`.

La commande `less` permet d'afficher des fichiers dans l'interpréteur de commandes, écran par écran.

Question. Affichez le fichier `/etc/mime.types` avec la commande `cat` puis avec la commande `less`.

8 Création de dossiers

On peut créer des dossiers grâce à la commande `mkdir` (*make directory*). Ceci se fait en lui donnant en argument le chemin (relatif ou absolu) du dossier à créer.

Question. Consultez la page de manuel traitant de la commande `mkdir`.

Question. Créez un répertoire `trav` dans votre répertoire de travail.

Question. Comment pouvez vous vérifier que le nouveau répertoire a bien été créé ? (Proposer au moins deux méthodes.)

Question. Placez-vous dans le répertoire `trav`.

Question. Créez un répertoire `monrep` dans le répertoire `trav`.

Question. En gardant le répertoire `trav` comme répertoire courant, créez un répertoire `sousrep` dans le répertoire `monrep`.

Question. Essayez la commande `mkdir monrep/unrep/unautrerep`. Que se passe-t-il ?

Question. Placez-vous dans le répertoire `sousrep`.

Question. Créez, en utilisant un chemin relatif, le répertoire `sousrep1` dans `monrep`.

Question. Créez, en utilisant un chemin absolu, le répertoire `sousrep2` dans `monrep`.

Question. Créez, en utilisant un chemin relatif, le répertoire `prive` dans `sousrep`.

Question. Créez, en utilisant un chemin absolu, le répertoire `jeux` dans `sousrep`.

Question. Représentez l'arborescence obtenue dans le répertoire `trav`. Vous pouvez vérifier que vous avez représenté l'arborescence correcte en utilisant la commande `tree`.

9 Copie de fichiers

La commande `cp` (*copy*) permet de copier des fichiers (et dossiers).

Question. Consultez la page de manuel de la commande `cp` pour comprendre comment l'utiliser.

Question. Placez-vous dans le répertoire `monrep` créé à la section précédente.

Question. Copiez le fichier `fichierTexte.txt` situé dans votre répertoire de travail dans le répertoire `prive` créé précédemment.

Question. Copiez le fichier de la question précédente dans le répertoire `sousrep1` en utilisant des chemins absolus.

10 Déplacement de fichiers

La commande `mv` (*move*) permet de déplacer des fichiers (et dossiers).

Question. Consultez la page de manuel de la commande `mv` pour comprendre comment l'utiliser.

Question. Placez-vous dans le répertoire `sousrep1`.

Question. Renommez le fichier `fichierTexte.txt` situé dans le répertoire courant en `fichierTexte.txt.bak`.

11 Copies et déplacements multiples

Question. En une seule commande, copiez le fichier `fichierTexte.txt.bak` dans le répertoire `prive`, en le renommant en `scriptBasic2.sh` en utilisant des chemins relatifs.

Question. En une seule commande, copiez le fichier `fichierTexte.txt.bak` dans le répertoire `prive`, en le renommant en `fichierTexte.txt.old` en utilisant des chemins absolus.

Question. Placez-vous dans le répertoire `sousrep2` créé précédemment.

À la place d'utiliser un nom de fichier précis dans une commande, on peut utiliser le symbole `*` pour traiter plusieurs fichiers à la fois. Par exemple, la commande `cp ../*.jpg .` copie tous les fichiers finissant par `.jpg` du répertoire parent dans le répertoire courant (un point `.` indique le répertoire courant).

Question. En une seule commande, copiez tous les fichiers du répertoire `prive` dont l'extension est `.sh` dans le répertoire `sousrep1`.

Question. En une seule commande, déplacez toute l'arborescence du répertoire `sousrep` dans le répertoire `sousrep2`.

12 Suppression de fichiers et dossiers

La commande `rm` (*remove*) permet de supprimer des fichiers et des dossiers. La commande `rmdir` (*remove directory*) permet de supprimer des dossiers.

Question. Consultez les pages de manuel de ces commandes pour comprendre comment les utiliser.

Question. Placez-vous dans le répertoire `monrep`.

Question. Supprimez le fichier `fichierTexte.txt.old` du répertoire `~/trav/monrep/sousrep2/sousrep/prive`.

Question. Supprimez en une seule commande toute l'arborescence de `sousrep2`.

Question. Supprimez en une seule commande tous les fichiers de `sousrep1`.