Informatique tronc commun TP 02

16 septembre 2015

1. **Lisez attentivement tout l'énoncé avant de commencer.**
2. Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
3. Après la séance, vous devez rédiger un compte-rendu de TP et l'envoyer au format électronique à votre enseignant.
4. Le seul format accepté pour l'envoi d'un texte de compte-rendu est le format PDF.
5. Ce TP est à faire en binôme, vous ne rendrez donc qu'un compte-rendu pour deux.
6. Ayez toujours un crayon et un papier sous la main. Quand vous réfléchissez à une question, utilisez-les !
7. Vous devez être autonome. Ainsi, avant de poser une question à l'enseignant, merci de commencer par :
   — relire l'énoncé du TP (beaucoup de réponses se trouvent dedans) ;
   — relire les passages du cours [1] relatifs à votre problème ;
   — effectuer une recherche dans l'aide disponible sur votre ordinateur (ou sur internet) concernant votre question.

   Il est alors raisonnable d'appeler votre enseignant pour lui demander des explications ou une confirmation !

Le but de ce TP est de vous faire découvrir quelques commandes importantes sous Linux, et de vous démontrer la puissance de la ligne de commande.
Pour les manipulations de fichier, vous utiliserez le code source du logiciel python, que vous pourrez trouver à cette adresse :

https ://hg.python.org/cpython/archive/tip.tar.gz

Le nom du fichier que ce lien vous permet de télécharger est de la forme `cpython-????.tar.gz` où `????` est à remplacer par le numéro de version du jour. Le 26 août, ce numéro était `2e9cf58c891d`, mais il sera sûrement différent le jour du TP. Dans la suite du TP, on continuera de noter `????` ce numéro, mais il faudra le remplacer par la bonne valeur du moment.

1. Dans le cas fort improbable où vous ne vous en souviendriez pas.

1 Terminal et shell

Unix a été inventé à un moment où l'utilisateur avait la possibilité d'interagir avec l'ordinateur via un *terminal*, c'est-à-dire la combinaison d'un clavier et d'un écran pouvant écrire (en général) 25 lignes de 80 caractères (en une seule couleur, généralement vert ou orange sur fond noir). Cette façon d'interagir avec la machine peut paraître archaïque de nos jours mais elle est pourtant d'une puissance diabolique.

Vous trouverez un émulateur de terminal dans le menu « Applications », sous-menu « Accessoires », et sélectionnez « LXTerminal ». Ceci démarre un programme, appelé *shell* ou *interprète de commandes*. Ce *shell* vous donne quelques informations, et affiche un symbole `$`, appelé invite (ou *prompt* en anglais) signe qu'il attend vos ordres.

Pour lui donner un ordre, il suffit de taper le nom de la commande désirée, éventuellement suivie d'un espace puis d'options ou d'arguments séparés par des espaces, puis de valider par la touche Entrée.

Par exemple, vous pouvez essayer la commande `ncal`.

Une commande très pratique est `man` : elle permet d'obtenir le manuel de quasiment toutes les commandes. On l'utilise sous la forme `man` *page* où *page* est la page de manuel désirée.

Q1 Tapez `man ncal`. Que se passe t-il ?

Vous pouvez faire défiler le texte ligne par ligne avec Entrée ou page par page avec la barre d'espace et quitter `man` avec la touche q.

Q2 Quelle est la date de Pâques en 2019 ?

2 Fichiers et répertoires

Sous Unix (dont la distribution GNU/Linux est un représentant) les fichiers sont organisés hiérarchiquement en une arborescence unique de répertoires. La racine de cette arborescence, c'est-à-dire le répertoire supérieur de la hiérarchie contenant tous les fichiers auxquels à accès le système, est noté `/`. Ses sous-répertoires directs (de l'ordre de la dizaine ou quelques dizaines de répertoires), comme `home`, `media`, ... sont notés `/home`, `/media`, ...

Le *chemin absolu* d'un fichier est l'adresse complète de son emplacement, débutant de la racine et passant par tous les sous-répertoires requis pour atteindre le fichier visé.

Q3 Que fait la commande `ls` ?

Q4 Exécuter la commande `tar xvzf cpython-????.tgz` : que se passe-t-il ?

Q5 Exécuter les commandes `mkdir TP02` puis `mv cpython-????/ TP02/`, puis `rm cpython-????.tar.gz` : expliquer ce qui se passe à chaque étape.

La commande `cd` permet de changer de répertoire courant. `pwd` permet d'afficher le répertoire courant, en particulier, `cd` *d* où *d* est le nom absolu d'un répertoire change le

répertoire du *shell* en *d*. Essayez avec `cd /usr/bin` par exemple.

Q6 Que fait `cd` sans argument ? (i.e. `cd` non suivi du nom d'un répertoire)

Q7 Que fait `cd ~` ? `cd ~/TP02` ? De quoi `~` est-il l'abréviation ?

Q8 Après éxécuter la commande `cd ~/TP02`, qu'affiche la commande `pwd` ?

Dans un répertoire, les fichiers et répertoires dont le nom commence par un point sont dits *cachés*.

Q9 En consultant le manuel de `ls`, trouver la commande qui permet d'afficher les fichiers et répertoires cachés.

Q10 Dans `~/TP02`, vous pouvez alors voir deux répertoires cachés. Quels sont leurs noms ?

En fait, dans chaque répertoire du système, il existe deux répertoires cachés avec ces deux mêmes noms.

Q11 Que donne un `cd` sur chacun de ces répertoires ?

Q12 Placez-vous dans le répertoire `~/TP02/cpython-????/Lib/test/capath`. Que fait alors `cd ../../multiprocessing` ?

On dit que `../../multiprocessing` est un nom de répertoire *relatif* au répertoire courant. La plupart des commandes qui acceptent des noms de fichiers ou de répertoires acceptent aussi bien les noms relatifs que les noms absolus.

Q13 Comment obtenir grâce à la commande `ls` et l'option `-l` la taille en Ko, Mo et Go, de tous les fichiers de `TP02/cpython-????/Modules`, en les triant par ordre décroissant de taille ?

Q14 Avec la commande précédente, que remarquez-vous quant à la taille des sous-répertoires de `TP02/cpython-????/Modules` ?

Q15 En utilisant la commande `du`, donner la taille du répertoire `cjkcodecs`. En comparant ce résultat à celui de la question précédente, que pouvez-vous dire de la manière dont Linux considère les répertoires ?

3 Chercher et rediriger

La commande `find` est la commande de recherche pour retrouver des fichiers, mais aussi effectuer des opérations sur les fichiers trouvés. C'est une commande excessivement puissante.
Commençons par rechercher un fichier dont on connaît le nom, dans un répertoire précis. La syntaxe est `find nom_repertoire -name nom_fichier`. Cette commande va chercher le fichier `nom_fichier` dans le répertoire `nom_répertoire` et ses sous-répertoires.

Q16 Quel est le chemin relatif à partir du répertoire `cpython-????` du fichier nommé `multibytecodec.c.h` ?

Si vous ne connaissez qu'une portion du nom du fichier recherché, ou que vous voulez trouver tous les fichiers dont le nom contient une certaine suite de caractères, vous pouvez utiliser le caractère joker * : par exemple, le fichier `toto_titi_tata` peut être recherché en cherchant les fichiers dont le nom est de la forme `*_titi*` : le fichier que vous cherchez sera parmi les fichiers obtenus, mais il peut aussi y en avoir d'autres, par exemple le fichier `toto_titi_tutu`.

Q17 Combien y a-t'il de fichiers dont l'extension est `.jpg` dans `cpython-????` et ses sous-répertoires ?

Une autre force du langage shell est la possibilité de rediriger la sortie d'une commande vers une seconde commande, en utilisant le symbole |, appelé *pipe*, ou *tube* en français. La syntaxe est : `commande 1 | commande 2`.

Q18 En utilisant la commande `wc -l` qui compte les lignes d'un affichage et l'option `type` de la commande `find`, donner le nombre total de fichiers contenus dans `cpython-????` et ses sous-répertoires.

4 Processus

Unix est un système multitâche, c'est-à-dire qu'il peut faire tourner en parallèle plusieurs programmes. Chacun de ces programmes en cours d'exécution est appelé un *processus*.

Pour visualiser les processus du système : `ps aux`.

Comme vous pouvez le constater, le résultat est difficile à lire. Les shells sous Unix possèdent un mécanisme très puissant, appelé redirection, qui permet de rediriger le résultat d'une commande vers un fichier. Essayez `ps aux > resultat.txt` et ouvrez le fichier produit avec leafpad pour regarder la liste des processus. La première ligne explique la signification des différentes colonnes (PID signifie : *process identifier*).

La commande `kill` permet de tuer un processus à partir de son numéro, pour autant

qu'on en ait la permission. Ce peut être utile dans le cas d'un processus qui continue à tourner (donc à consommer des ressources du système) alors qu'il aurait dû s'arrêter.

Lancez l'explorateur de fichiers (pcmanfm), trouvez son pid puis tuez ce processus par la ligne de commande.

Pour les processus ayant une interface graphique et qui seraient bloqués, la commande `xkill` est utile.

Q19 Que fait xkill ?

La commande `top` est également intéressante :

Q20 Que fait-elle ?

Télécharger les fichiers `titi.py` et `toto.sh` sur le site de la classe. Le fichier `toto.sh` est un *script* : il contient des commandes. Si l'on *exécute* ce script, les commandes qu'il contient seront lancées. Dans le cas présent, `toto.sh` va lancer un programme python, contenu dans le fichier `titi.py`. Pour pouvoir excéuter un script, il faut d'abord rendre ce fichier excéutable, c'est-à-dire modifier les droits d'accès au fichier pour signifier au système qu'une certaine catégorie de personnes a le droit d'exécuter le script. Il existe essentiellement trois types de personnes : le *propriétaire* du fichier, les *groupes d'utilisateurs*, qui regroupent plusieurs utilisateurs pouvant se connecter sur la machine ou le réseau, et l'*administrateur*. Ici, nous allons seulement donner le droit d'exécution au propriétaire, c'est-à-dire vous. Ceci se fait avec la commande `chmod u+x toto.sh`. On lance ensuite le script avec la commande `sh toto.sh`.

Q21 Lancer la commande `top` : décrivez ce que vous observez. Constatez-vous quelque chose de gênant ?
Q22 Tuer le processus qui pose problème.

5 Questions facultatives pour les rapides

Q23 Créer un fichier texte nommé `ty.txt`, que vous joindrez à votre compte-rendu, contenant la liste de tous les fichiers de `cpython-????` et ses sous-répertoires dont le nom contient la suite de caractères `ty` ; dire combien il y a de tels fichiers, et les effacer (on pourra utiliser les commandes `find`, `grep`, `xargs` et `rm`).

Q24 Trouver le plus gros fichier de `cpython-????` et ses sous-répertoires, et le déplacer dans le répertoire `TP02`. Préciser les commandes utilisées.

Q25 Guido Von Rossum est le créateur du logiciel Python : combien de fois le nom "Rossum" apparaît-il dans les fichiers de `cpython-????` et ses sous-répertoires ? Préciser les commandes utilisées.