Défi Big Data

Prise en main du Shell

Xavier Serpaggi

Mars 2018

Travail à rendre

Par binôme, vous écrirez les réponses aux questions ci-dessous dans un fichier texte. Aucun autre format n'est attendu.

Ce fichier texte sera déposé sur Campus, même partiel, à la fin de la séance.

Hormis pour l'étape 1, les réponses seront toujours les commandes ou les scripts vous ayant permis de résoudre le problème. Il ne faut mettre que les commandes ou les scripts, pas les résultats.

Arrivée

Démarrez votre machine virtuelle Linux (Ubuntu) en prenant soin de vérifier que le mode réseau dans VirtualBox est bien positionné sur NAT.

Étape 1 : Se familiariser avec l'environnement

> ■ *Pour commencer, vous vous familiariserez avec l'environnement et les commandes de base qu'il vous faudra maîtriser.*
> *Durant ce cours et particulièrement dans cette partie, il faut lire beaucoup de documentation. Ne faites pas l'impasse, c'est du temps que vous gagnerez ensuite.*

Les systèmes d'exploitation de la famille Unix, comme ici Linux, ont toujours fait la part belle aux utilitaires de traitement de texte. En effet, la majorité des fichiers de configuration, des fichiers de données et des textes, sont stockés sous la forme de fichiers texte.

Un livre y a même été consacré il y a quelques années de cela : *"Unix Text Processing"*. Il est à présent disponible gratuitement sur le site de l'éditeur [1].

Un des chapitres de ce livre commence comme celà :

> Let the Computer Do the Dirty Work
>
> Computers are very good at doing the same thing repeatedly, or doing a series of very similar things one after another. These are just the kinds of things that people hate to do, so it makes sense to learn how to let the computer do the dirty work.

Et même dans le domaine du traitement de données *a priori* numériques, connaître ces outils s'avère être un atout important. Un autre livre, plus récent, a été consacré à cela : *"Data Science at the Command Line"* [2]. Le sous-titre est là encore révélateur : *"Facing the Future with Time-Tested Tools"*.

Mais pour courir il faut d'abord apprendre à marcher. C'est ce que les quelques exercices suivants vont vous permettre de faire.

1. http://www.oreilly.com/openbook/utp/
2. http://datascienceatthecommandline.com/

Trucs & astuces
Lorsque l'on n'est pas habitué à l'utilisation de la ligne de commande on peut être parfois dérouté par ce qu'il s'y passe. Vous trouverez ici quelques trucs pour vous en sortir.

- Pour sortir d'un outil de visualisation de fichier, la touche `Q` est souvent utile.
- Quand une commande bloque, la combinaison de touches `Control`+`C`, dans le terminal, permet d'y mettre fin (*kill* en anglais).
- La combinaison de touches `Control`+`D` permet d'envoyer le caractère EOF (*fin de fichier*).
- Si on a lancé une commande et qu'elle ne rend pas la main, on peut la passer en arrière plan en 1) la STOPpant avec `Control`+`Z` et 2) la relançant avec la commande `'bg'` (*background*).
- Parfois, on appuie par erreur sur `Control`+`S` et tout se bloque dans le terminal. Tout se débloquera avec `Control`+`Q`.

Exercices

1. Connectez-vous avec votre compte [3].
2. Ouvrez au moins un terminal (il y a un icône dans la barre du bas et une entrée dans un des menus).
   Nous ne travaillerons qu'avec ça, forcez-vous à bannir l'utilisation du navigateur de fichiers, particulièrement pour *ouvrir* un fichier en double-cliquant dessus...
   Ce terminal est le support de l'interprète de commandes `'bash'`.
3. Les environnements de la famille Unix sont toujours très fortement documentés. Avec la commande `'man'`, lisez le manuel de la commande ... `'man'` (indice : tapez `man man` dans votre terminal).
4. Avec (au moins) les commandes `'cd'`, `'ls'`, `'mkdir'`, `'pwd'` et `'touch'` dont chacune a une page de manuel qu'il est conseillé de lire, accomplissez les actions suivantes :
   - trouvez le nom du répertoire dans lequel vous vous trouvez ;
   - listez le contenu de ce répertoire et donnez les fichiers et les répertoires qui s'y trouvent ;
   - créez un répertoire du nom de `BigData_Shell` ;
   - déplacez-vous dans ce répertoire [4] ;
   - créez un fichier vide dans ce répertoire, ce sera le fichier dans lequel vous noterez vos réponses.
5. Vous remarquerez qu'il y a plusieurs façons d'exprimer un chemin : de manière absolue ou de manière relative. Donnez les chemins absolus et relatifs à votre répertoire personnel, du répertoire `BigData_shell`.

À propos de l'interprète de commandes, il manipule des variables. Vous pouvez afficher les principales avec la commande interne `'declare -x'`.

Étape 2 : Entrées et sorties

■ *Nombreuses sont les commandes qui produisent un affichage sur la sortie standard.*
Nombreuses sont les commandes qui peuvent lire leurs données depuis l'entrée standard.
Vous allez ici apprendre à utiliser les redirections pour écrire et lire les données où bon vous semble.
Pour répondre à chacune des questions, une seule ligne de commande est nécessaire et bien entendu, tout doit être fait en ligne de commande, sans utiliser le navigateur de fichiers.

La sortie standard, c'est en général le terminal dans lequel est lancée votre commande. L'entrée standard, c'est en général le clavier que vous utilisez pour taper vos commandes. Pour modifier ces

3. Si ça ne fonctionne pas, il faut redémarrer la machine virtuelle jusqu'à ce que ça fonctionne...
4. Si vous avez l'habitude d'organiser votre travail par répertoires, vous pouvez créer le répertoire `BigData_Shell` dans un de vos répertoires, au lieu de le créer directement dans votre répertoire personnel

valeurs par défaut, il existe des mécanismes de redirection. Un chapitre y est consacré dans la page de manuel de `'bash'`, votre interprète de commandes, c'est le chapitre *REDIRECTIONS*[5]. Lisez-le au moins jusqu'à *Ajout d'une sortie redirigée* inclus.

D'autres commandes seront abordées, comme `'echo'`, `'cat'`, `'cp'`, `'mv'` et `'rm'`. Elles vous permettront d'afficher, copier, déplacer/renommer et supprimer des fichiers.

Exercices

1. Avec la commande `'cat'`, affichez le contenu du fichier dans lequel vous avez noté les réponses aux questions précédentes.
2. Faites une copie de votre fichier réponse avec la commande `'cp'`.
3. À l'aide de la commande `'echo'` et de la redirection adéquate, écrivez la réponse à cette question à la fin **de la copie** de votre fichier.
4. Si tout s'est bien passé (si vous avez utilisé la bonne redirection), votre fichier copie doit contenir une nouvelle ligne à la fin. Affichez-le pour vérifier et recommencez les étapes précédentes si ce n'est pas le cas.
5. Supprimez votre fichier réponse avec la commande `'rm'`.
6. Renommez la copie pour qu'elle ait le même nom que le fichier original avec la commande `'mv'`.

Bien entendu, de nombreuses étapes de la manipulation de votre fichier réponse demandées ci-dessus sont inutiles. Elles ont cependant un but pédagogique !

Étape 3 : Chercher et compter

> *Ici vous prendrez en main les outils basiques de traitement de fichiers qui ne relèvent pas du système de fichiers.*
> *Encore une fois, la lecture des pages de manuel des commandes fait partie de l'exercice, même si ces pages peuvent parraître rebutantes de premier abord.*

Les recherches peuvent se faire de deux façons différentes : avec la commande `'find'` pour trouver des fichiers et des répertoires dont le nom correspond à un motif donné ; avec la commande `'grep'` pour trouver les lignes d'un fichier qui correspondent (ou ne correspondent pas) à un motif donné.

Les commandes peuvent s'enchainer (la sortie de l'une devient l'entrée de la suivante) à l'aide du *pipe* représenté par le caractère `|` (AltGr+8).

Compter se fait facilement avec la commande `'wc'`.

Exercices

1. Utilisez la commande `'find'` pour donner la liste de tous les fichiers présents dans votre répertoire personnel (uniquement les fichiers)
2. En utilisant la commande `'grep'`, obtenez la liste des fichiers de votre répertoire personnel (et de tous les sous-répertoires) qui contiennent la sous-chaîne `le`. On ne veut que le nom des fichiers.
3. À l'aide de la commande `'find'` et d'options judicieusement choisies donnez la liste 1) des fichiers et 2) des répertoires présents dans le répertoire `/etc`. On ne s'intéresse qu'aux fichiers et répertoires directement présents dans le répertoire `/etc`, pas à ceux présents dans les sous-répertoires.
4. À l'aide de la commande `'wc'`, comptez les caractères, mots et lignes du fichier `/etc/passwd`. De quelle grandeur pouvez-vous rapprocher le nombre de caractères que vous obtenez ?

5. pour trouver le paragraphe rapidement, vous pouvez utiliser le mécanisme de recherche couplé à un mécanisme d'expression régulière : dans la page de manuel, tapez `/^REDIR` et validez. Le `/` permet de faire une recherche et `^REDIR` signifie que l'on cherche les caractères `REDIR` qui sont au début d'une ligne (`^`).

5. À l'aide des commandes `'find'` et `'wc'`, comptez le nombre de fichiers présents dans le répertoire `/etc` 1) sans prendre en compte les sous-répertoires et 2) en tenant également compte des fichiers présents dans les sous-répertoires.
6. La plupart des fichiers présents dans le répertoire `/etc` sont des scripts shell. En utilisant les commandes `'grep'` et `'wc'`, comptez le nombre de lignes de commentaires (lignes qui débutent par `#` mais pas par `#!`) dans tous les fichiers du répertoire `/etc` ainsi que ses sous-répertoires.
7. Reprenez la question 2 en ne recherchant que les fichiers dans lesquels apparaît le mot `le` qui ne soit pas une partie d'un autre mot comme par exemple `lequel`, `poule`, `facilement`.

Étape 4 : Manipulations de textes

> *Le but de cette étape est de réaliser des opérations un peu plus complexes sur des fichiers textes. Nous allons travailler sur le texte de H.G Wells « The Time Machine ».*
> *Le choix d'un texte en langue anglaise est volontaire. En effet, c'est une langue qui ne comporte pas de caractères au delà de ce que le code ASCII est capable de représenter. Même encodé en UTF-8, des fichiers avec d'autres caractères (les accents en Français par exemple) deviennent une plaie à traiter de manière systématique.*

Tous les textes proposés ici sont dans le domaine public.

Exercices

1. Avec le programme `'curl'`, récupérez le fichier texte correspondant au livre
   https://www.gutenberg.org/files/35/35-0.txt
   et renommez-le `time-machine.txt`.
   Attention à la configuration du proxy si le téléchargement ne fonctionne pas.
2. Une fois le fichier téléchargé, vous pouvez obtenir quelques informations sur son type à l'aide de la commande `'file'`
3. Ce fichier comporte un petit texte de description/copyright au début et un long texte de licence à la fin.
   La description/copyright se termine par une ligne du type `*** START OF THIS ...` et le début de la licence est identifié par une ligne du type `*** END OF THIS ...`
   Avec les outils `'grep'`, `'head'` et `'tail'`, extraire le texte seul. Plusieurs étapes sont nécessaires :
   — trouver les numéros des lignes de début et de fin du texte avec les commandes `'grep'` et `'cut'` ;
   — retirer la licence avec `'head'` ;
   — retirer l'en-tête avec `'tail'`.
4. Avec tout ou partie des outils `'cat'`, `'grep' 'sort'`, `'tr'`, `'uniq'`, créez la liste des mots[6] présents dans ce fichier et sauvegardez la dans le fichier `time-machine.idx`.
   Vous pouvez être gênés par le caractère `^M` qui représente un saut à la ligne. Vous pouvez le filtrer avec `'grep'` ou le transformer avec `'tr'`.
5. Améliorez la liste précédente en ne conservant que les mots qui pourraient apparaître dans un dictionnaire (sans caractères spéciaux, valeurs numériques, ponctuation, ...)

Étape 5 : Automatiser la tâche

> *Pouvoir réaliser une tâche complexe à l'aide d'outils simples et éprouvés est une bonne chose. Il est encore plus intéressant de pouvoir automatiser cette tâche si elle doit se répéter souvent. C'est à dire qu'elle doit pouvoir se réaliser de bout en bout, quels que soient les paramètres, avec une intervention humaine minimale.*

6. Un mot est défini par toute séquence maximale de caractères non séparateurs de mots. Les séparateurs de mots sont les espaces, les tabulations et les retours à la ligne.

Exercices

1. Écrire un script qui automatise la construction d'un dictionnaire comme vu à l'étape précédente. Votre script doit prendre en paramètre l'URL du fichier à traiter. On ne saura donc pas exactement où se termine la licence ni où commence la table des matières. On ne connait (à peu près) que les textes qui marquent ces frontières.
2. Vous testerez votre script sur les livres suivants :
   — http://www.gutenberg.org/cache/epub/910/pg910.txt
   — https://www.gutenberg.org/files/84/84-0.txt
   — http://www.gutenberg.org/cache/epub/1112/pg1112.txt
   — http://www.gutenberg.org/cache/epub/345/pg345.txt
3. Vous pouvez améliorer votre script en tirant partie des informations de l'en-tête où figurent le titre et l'auteur du livre pour renommer le fichier après téléchargement sous la forme `<auteur>-<titre>.txt`.

Étape 6 : Sans programmer

> ■ *Parfois il vaut le coup d'écrire un programme pour traiter ses données, parfois il est suffisant d'utiliser des outils déjà existants.*

La commande `'time'` permet de mesurer le temps nécessaire à l'exécution d'une tâche.

1. Reprenez le fichier proposé dans l'UP Organisation des données (le fichier `final` de 1 Go) et refaites l'exercice en utilisant les outils que vous avez découverts ici.
2. Recherchez le nombre d'occurrences de `100000`, 100001, `375888`, puis de `1` et de `0`. Adaptez votre commande en fonction des résultats obtenus.
3. Comparez les temps obtenus.
4. Utilisez[7] le programme `random.c` disponible sur Campus pour générer un fichier équivalent à `final`. Essayez de trouver le moyen le plus rapide pour y parvenir et mesurez les temps pris par vos expériences.

7. Téléchargez-le et compilez-le, mais ne le modifiez pas, sauf si vous y trouvez un bug !