

1 Introduction

L'objectif du présent TD est de jouer avec quelques commandes classiques disponibles en ligne de commandes sous Unix, et de se familiariser avec l'enchaînement de commandes et la redirection des entrées sorties. On verra aussi quelques exemples de règles de ré-écritures du BASH : l'expansion de variables, l'expansion de chemins et l'expansion de commandes. Si vous avez des difficultés, consultez les indices de la dernière page, voire les solutions si ce n'est pas suffisant et pour vérifier sachant qu'il existe plusieurs bonnes réponses à une même question !

2 Téléchargement de fichiers

Nous allons commencer par récupérer quelques fichiers en ligne, grâce à la commande `wget` qui permet de télécharger un fichier à partir d'une URL via les protocoles `http` et `https`. À faire :

1. installer `wget` (indice 1) et parcourir la page de man correspondante ;
2. créer un répertoire « TD3 » dans votre home et s'y rendre (indice 2) ;
3. utiliser `wget` pour récupérer la version « plain Text UTF-8 » du livre « De l'origine des espèces » de Charles Darwin à l'URL <https://www.gutenberg.org/ebooks/14158> (indice 3).

Si vous avez un message d'erreur, c'est sans doute que l'installation de la Debian sous WSL n'inclut pas les « certificats » nécessaires au protocole `https`. Vous en apprendrez davantage sur ces certificats dans un prochain cours de réseau et/ou de sécurité. En attendant, il suffit d'installer les certificats, ce qui hélas peut prendre un peu de temps. À faire :

1. taper « `sudo apt-get install ca-certificates` », profiter du temps de traitement pour commencer à lire le reste de la fiche de TD ;
2. essayer de nouveau de récupérer le fichier (indice 3) ;
3. compter le nombre de mots présents dans le fichier récupéré (indice 4) ;
4. lire le début du fichier sans utiliser d'éditeur (indice 5) ;
5. créer un fichier texte nommé « `liste.txt` » et y copier la liste des fichiers donnée sur la page Moodle du CM du cours (ressource « Liste de fichiers pour le TD 3 ») (indice 6) ;
6. télécharger tous les fichiers listés dans `liste.txt` (indice 7) ;
7. afficher la ligne contenant le titre du livre pour chaque fichier d'extension `.txt` (indice 8).

3 Enchaînement de commandes et redirection

Nous continuons à travailler dans le répertoire TD3. À faire :

1. reprendre la dernière question de la section précédente, mais n'afficher que les titres et non les lignes complètes, lire le man de la commande `cut` pourrait être utile (indice 9) ;
2. créer à la racine de votre compte un fichier `titres.txt` contenant les titres des livres contenus dans le répertoire TD3 (indice 10) ;
3. trouver et tester l'enchaînement de commandes qui affiche « Un livre se passe à Baltimore » si le mot « Baltimore » apparaît dans un des livres (indice 11), faire en sorte qu'aucun autre texte ne s'affiche ;

4. trouver et tester l'enchaînement de commandes qui affiche « Le répertoire /root n'est pas accessible » si c'est bien le cas (indice 12), faire en sorte qu'aucun autre texte ne s'affiche ;
5. tester la commande `date` puis créer un fichier `log.txt` à la racine de votre compte contenant la date et l'heure (indice 13) ;
6. utiliser la commande `date` pour ajouter la date et l'heure à la fin de `log.txt` (indice 14) ;
7. trouver et tester la ligne de commandes qui affiche le nombre de fichiers et répertoires avec le droit de lecture (r) pour leur propriétaire dans le répertoire courant (indice 15) ;
8. chercher la notion d'ancrage dans la page de man de `grep` (métacaractère « `^` ») et tester les lignes de commandes « `ls -l | grep ^d` » et « `ls -l | grep ^-` » ;
9. trouver et tester la ligne de commandes qui affiche le nombre de fichiers exécutable (x) par leur propriétaire dans le répertoire courant (indice 16), prendre garde à ne pas inclure dans le décompte les répertoires traversables (qui ont aussi le droit x).

4 Expansions de commandes

Nous avons vu en cours qu'une force du BASH est sa capacité à procéder à des « expansions » (ou « substitution » ou « ré-écriture ») avant de lancer les commandes. En effet, dans la ligne de commandes « `ls *.txt` », « `*.txt` » est remplacé par l'interpréteur par la liste des fichiers correspondant avant que l'interpréteur ne lance la commande `ls`, ainsi la commande `ls` n'a pas besoin de faire elle-même ce traitement, il en va évidemment de même pour les programmes écrits par l'utilisateur. De même, le BASH va par exemple effectuer l'expansion de variables en remplaçant `$toto` par le contenu de la variable `toto`, ou encore l'expansion de commandes en remplaçant `$(cmd)` par le résultat de la commande `cmd`. À faire dans le répertoire TD3 :

1. tester « `echo "Il y a $(ls *.txt | wc -l) fichiers texte dans $PWD".` » ;
2. tester « `nbt=$(grep -c évolution 799-0.txt) ; echo $nbt` ».

Pour la dernière commande, notez que l'introduction d'un espace avant ou après le symbole égal provoque des erreurs. N'hésitez pas à tester et à essayer de comprendre le pourquoi des messages d'erreur, en gardant à l'esprit que le BASH procède par ré-écritures successives de la ligne de commandes et non par son analyse comme le fait par exemple un interpréteur Python. À faire :

1. relire la page de man de `date` (notamment l'utilisation de « Format ») et créer une variable « jour » contenant le jour de la semaine (par exemple « lundi ») (indice 17) ;
2. créer un répertoire à la racine de votre compte – en utilisant un chemin absolu – nommé à partir du jour de la semaine (indice 18).

5 Scripts

Un script BASH est simplement un fichier texte avec une liste de commandes. À faire :

1. créer un script nommé « coucou » qui affiche la phrase « *hello world* » (indice 19) ;
2. tester coucou (indice 20) ;
3. créer un répertoire `bin` à la racine de votre compte et modifier la variable d'environnement `PATH` pour inclure ce nouveau répertoire (indice 21) ;
4. déplacer coucou dans `bin` et tester coucou sans préciser le chemin (indice 22) ;
5. modifier coucou pour qu'il souhaite un bon lundi si le jour de la semaine est lundi, en plus d'afficher « *hello world* » (solution 23).

6 Indices

- 16 s'inspirer des deux questions précédentes
- 20 il faut préciser le chemin et se donner le droit d'exécution
- 5 utiliser la commande `more`, re-parcourir la page de man si nécessaire
- 13 utiliser la redirection vers un fichier (`cmd > fichier`), utiliser un chemin absolu pour le fichier
- 12 utiliser par exemple les commandes `cd` et `echo`, rediriger les erreurs vers `/dev/null`
- 19 utiliser un éditeur comme `nano` et la commande `echo`, ou simplement `echo;-)`
- 17 utiliser la commande `date` et l'expansion de commandes
- 4 utiliser la commande `wc`, sans oublier de parcourir la page de man
- 11 utiliser les commandes `grep` et `echo` et l'enchaînement en cas de succès
- 10 utiliser la redirection vers un fichier (`cmd > fichier`), utiliser un chemin absolu pour le fichier
- 18 utiliser la commande `mkdir` et la variable créée à la question précédente
- 8 utiliser la commande `grep` et l'expansion de chemin (« `*.txt` »)
- 14 utiliser la redirection vers un fichier pour ajout (`cmd >> fichier`)
- 21 utiliser l'expansion de variables pour ajouter `~/bin` au `PATH`
- 9 utiliser un tube (« `|` ») et la commande `cut` en spécifiant le champ (3e) et le délimiteur (« `:` »)
- 6 utiliser l'éditeur `nano` et faire un copier-coller depuis votre navigateur vers l'éditeur
- 3 aller sur la page web, récupérer l'URL du fichier recherché, utiliser `wget` pour le récupérer
- 7 relire le man de `wget` et notamment ce qui est relatif à l'option « `-i` »
- 1 utiliser la commande `apt-get` pour mettre à jour la liste des paquets puis pour installer `wget`
- 22 déplacer `coucou` avec la commande `mv`, aller n'importe où dans l'arborescence et taper `coucou`
- 15 enchaîner via des tubes les commandes `ls`, `cut` (avec l'option `-c`) et `grep`
- 2 utiliser la commande `mkdir`

7 Solutions

- 21 « `PATH=$PATH:~/bin` »
- 12 « `cd /root 2> /dev/null || echo "Le répertoire /root n'est pas accessible" »`
- 10 « `grep Title *.txt | cut -f 3 -d : > ~/titres.txt` »
- 9 « `grep Title *.txt | cut -f 3 -d :` »
- 18 « `mkdir ~/$jour` » ou « `mkdir ~/(date +%A)` »
- 1 « `sudo apt-get update ; sudo apt-get install wget` »
- 8 « `grep Title *.txt` »
- 19 « `echo "echo hello world" > coucou` »
- 14 « `date >> ~/log.txt` »
- 16 « `ls -l | grep ^- | cut -c 4 | grep -c x` »
- 22 aller où est `coucou` et faire « `mv coucou ~/bin ; cd ; coucou` »
- 17 « `jour=$(date +%A)` »
- 15 « `ls -l | cut -c 2 | grep -c r\fg` »
- 20 « `chmod u+x coucou ; ./coucou` »
- 6 « `nano liste.txt` », copier-coller (bouton droit pour coller dans WSL), sauvegarder, quitter
- 23 ajouter « `date +%A | grep -q lundi && echo "bon lundi !"` » à `coucou`
- 3 « `wget https://www.gutenberg.org/cache/epub/14158/pg14158.txt` »
- 13 « `date > ~/log.txt` »
- 5 « `more pg14158.txt` », touche « `h` » pour l'aide et « `q` » pour quitter
- 4 « `wc pg14158.txt` » ou mieux « `wc -w pg14158.txt` »
- 11 « `grep -q Baltimore *.txt && echo Un livre se passe à Baltimore` »
- 2 « `cd ; mkdir TD3 ; cd TD3` »
- 7 « `wget -i liste.txt` »