

Licence 1

Parcours CMI OPT/IM — Parcours CMI IS — Parcours Mathématiques-informatique — Parcours informatique

Année académique 2018-2019 — Quadrimestre 2

Fonctionnement des ordinateurs

Laboratoire 1 – Série 1





Commandes Unix

Prof. Dr. Xavier Gandibleux*

Consigne générale : à l'aide de l'éditeur de texte "TextEditor (gedit)", collecter et structurer dans un fichier texte nommé `laboratoire1.txt` l'ensemble des réponses produites.

A la fin du TD, n'oubliez pas de vous déconnecter !

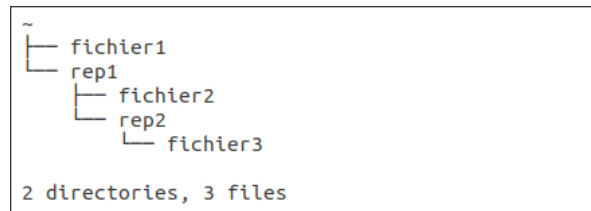
1 Enoncé (Manipulations de base)

1. Démarrer l'ordinateur, ouvrir une session linux, ouvrir un terminal.
Observez votre prompt. Comment se présente-t-il ?
2. L'instruction `whoami` indique votre *username*.
Invoquer cette instruction et relevez votre *username*.
3. A ce moment précis, où vous situez-vous dans l'arborescence des fichiers ?
Invoquer l'instruction ad-hoc pour vérifier votre réponse.
Cette localisation porte un nom particulier, quel est-il ?
4. Créer un fichier texte nommé `hello.txt`
Invoquer l'instruction ad-hoc pour vérifier qu'il est bien présent dans votre répertoire courant.
5. Essayez la séquence `ls he` suivi immédiatement de la touche  (tabulation).
Qu'observez-vous ?
6. Créer un fichier texte nommé `hello2.txt` en récupérant et modifiant votre première commande ;
pour cela, appuyez sur la touche 
7. Reprendre `ls he` suivi de 
Qu'observez-vous ?
Pressez encore la touche  (tabulation).
Qu'observez-vous ?
8. Ouvrez un second terminal.
Sans la resaisir, copiez votre dernière commande invoquée sur le premier terminal, dans le second terminal, suivant la méthode « sélectionnez un texte (bouton gauche de la souris) dans une fenêtre terminal pour ensuite le coller (bouton du milieu de la souris) dans une autre fenêtre terminal. »

*Avec la participation de Dr. Audrey Queudet durant l'année académique 2017-2018.

2 Enoncé (Création d'arborescence)

L'utilisateur `etu-0579` envisage de construire l'arborescence suivante :



1. A quel répertoire correspond `~` ?
2. Créez l'arborescence précédente :
 - (a) en adressage relatif au *home directory*,
 - (b) en adressage absolu,
 - (c) en adressage relatif au *current working directory* :
 - i. en supposant qu'on se situe dans le répertoire `/bin`
 - ii. en supposant qu'on se situe dans le répertoire `~`

3 Enoncé (Navigation dans l'arborescence (1/2))

1. Dans votre *home directory*, créer un répertoire `test` et deux sous-répertoires `pub` et `bin`.
2. Modifier les droits d'accès de façon que :
 - le répertoire `test` et son sous-répertoire `pub` soient accessibles en lecture et exécution pour tous, et en écriture pour le propriétaire seul.
 - le répertoire `bin` soit accessible en exécution pour tous, et en lecture et écriture pour le propriétaire et son groupe.
3. Aller dans le répertoire `test`. Créer un fichier `doc.txt`. Recopier le fichier dans `pub`.
4. Aller dans `bin`. Afficher à l'écran le contenu de `test/pub/doc.txt`.
5. Afficher la liste des fichiers contenus dans `test` et ses sous-répertoires.
6. Dessinez votre arborescence depuis votre *home directory* en faisant apparaître les répertoires et les fichiers
7. Supprimer du répertoire `test` et de ses sous-répertoires tous les fichiers se terminant par le caractère `~`.
8. Donner la commande qui permet d'afficher à l'écran le contenu des fichiers `bro1` et `truc` qui se trouvent dans le répertoire d'accueil de l'utilisateur `geluck-p`.

4 Enoncé (Navigation dans l'arborescence (2/2))

On désire aller dans le répertoire `/usr/local/games/mariokart`, et le répertoire courant est `/usr/local`.

Quelle(s) commande(s) peut-on taper (expliquer l'effet des 6 commandes proposées) ?

- A : `cd /games/mariokart`
- B : `cd games/mariokart`
- C : `cd local/mariokart`
- D : `cd /usr/local/games/mariokart`
- E : `cd /usr/local/../../local/games/mariokart`
- F : `cd ../games/mariokart`

5 Enoncé (Manipulation de fichiers)

Quelle différence y a-t-il entre les commandes `mv brol truc` et `cp brol truc` ?

6 Enoncé (Commande `ls`)

- Précisez les options à utiliser pour la commande `ls` dans les cas listés ci-dessous :
 - Listage incluant les fichiers cachés ou ceux qui commencent par `.`
 - Listage avec descriptif complet de chaque référence : droits, nombres de liens, dates, taille user group, etc.
 - Listage récursif du contenu d'un répertoire et de tous ses sous-répertoires.
 - Listage par ordre chronologique, et inverse.
 - Listage simple du contenu du répertoire, avec spécification du type de fichier : répertoire `/`, lien symbolique `@`, exécutable `*`

7 Enoncé (Jokers et expressions régulières)

- Lister tous les fichiers :
 - se terminant par `5`
 - commençant par `annee4`
 - commençant par `annee4` et de 7 lettres exactement
 - commençant par `annee` avec aucun chiffre numérique pour le caractère immédiatement suivant
 - contenant la chaîne `ana`
 - commençant par `a` ou `A`
- Donner la commande qui permet de copier depuis `.` les fichiers dont l'avant-dernier caractère est un `4` ou `1` dans le répertoire `~/tmp` en une seule commande.

8 Enoncé (Gestion des droits d'accès)

Dans votre répertoire courant, vous créez un répertoire courant `essai_droit`. Par défaut, ce répertoire est à `rwxr-xr-x`. Quelles sont les commandes pour lui donner les droits suivants (on suppose qu'après chaque commande on remet le répertoire à `rwxr-xr-x` :

	user			group			others		
	read	write	execute	read	write	execute	read	write	execute
Command1	oui	oui	oui	oui	non	oui	non	non	oui
Command2	oui	non	oui	non	oui	non	non	non	oui
Command3	non	oui	non	non	non	oui	oui	non	non
Command4	non	non	oui	oui	non	oui	non	non	non

9 Enoncé (Processus)

- Depuis un terminal, lancez une application graphique, par exemple "gedit" en invoquant `gedit`. Revenez sur le terminal, que constatez vous, pourquoi ?
- Quitter l'application via l'interface graphique.
Relancer l'application graphique selon la variante suivante : `gedit &`
Revenez sur le terminal, que constatez vous, pourquoi ?
- Trouvez le PID de l'application "gedit"
Depuis le terminal, tuez l'application "gedit" en invoquant l'instruction ad-hoc.

10 Enoncé (Commande find)

A l'aide de la commande `man`, reprendre la documentation concernant la commande `find`, ses options et paramètres.

1. Chercher tous les fichiers dont le nom est `passwd`.
2. Chercher tous les fichiers du *home directory* dont la dernière modification remonte à 10 jours.
3. Trouver tous les fichiers du groupe `root`.
4. Chercher tous les fichiers dont la taille est supérieure à 200 ko.
5. Chercher tous les répertoires se trouvant sous `/etc`.
6. Chercher tous vos fichiers en filtrant sur votre identifiant utilisateur.

11 Enoncé (Redirection et tubes)

1. Expliquez en quoi consistent les redirections suivantes :
 - (a) `ls > fic`
 - (b) `date >> fic`
 - (c) `find / -name toto 2 > /dev/null`
 - (d) `cp *.* /tmp 2>>erreur.log`
 - (e) `cat ~/.bashrc > config.txt 2>erreur`
2. Qu'affichent les commandes suivantes ?
 - (a) `ls -a /bin | wc -l`
 - (b) `cat /etc/lsb-release | tail -2`
 - (c) `cat /proc/meminfo | head -2`

12 Enoncé (Commandes grep et cut)

A l'aide de la commande `man`, reprendre la documentation concernant les commandes `grep` et `cut`.

On suppose qu'un fichier `liste.txt` contient des informations sur les étudiants (10 au moins). Chaque ligne représente un étudiant, et contient les informations suivantes : nom, âge et année d'études en cours. Les champs seront séparés par le symbole ;

Exemple : la ligne `Dumont;23;L1` correspond à l'étudiant Dumont, âgé de 23 ans et inscrit en L1.

1. Elaborer le fichier `liste.txt`
2. Renvoyer toutes les lignes du fichier `liste.txt` qui correspondent à l'étudiant s'appelant `Jules`.
3. Renvoyer toutes les lignes correspondant à des étudiants de L1.
4. Renvoyer toutes les lignes des étudiants âgés de 22 ans.
5. Renvoyer les lignes des étudiants n'étant pas en L1.
6. Renvoyer toutes les lignes contenant la chaîne `an` sans tenir compte de la casse.
7. Afficher le nom et l'âge de chaque étudiant, puis le nom et l'année.
8. Afficher les trois premiers caractères de chaque ligne.