TP 1 - Premiers pas sous GNU/Linux

Organisation.

- Travaillez dans le terminal de la machine virtuelle Debian. Vous pouvez consulter `man` et l'aide des commandes.
- Il n'est pas obligatoire de terminer tous les exercices aujourd'hui : nous poursuivrons lors de la prochaine séance.
- Écrivez vos commandes et vos explications dans `~/Linux/TP01/reponses.txt` avec `nano ~/Linux/TP01/reponses.txt`.
- Ne recopiez pas seulement les messages d'erreur : expliquez ce que vous avez compris. Vous pouvez également ajouter les fichiers créés pendant les exercices.

Rappel : le mot de passe du compte `etudiant` a été communiqué dans le groupe WeChat.

Exercice 1 (Premiers pas)

(a) Ouvrez le terminal. Exécutez `pwd`, puis `ls`. Que vous apprend chaque commande ?

(b) Créez un répertoire `Linux` dans votre répertoire personnel, puis placez-vous dedans avec `cd`. Vérifiez votre position avec `pwd`.

(c) Créez le sous-répertoire `TP01`, puis le fichier vide `TP01/reponses.txt`. Quelle commande permet de vérifier qu'il existe ?

(d) Essayez maintenant `mkdir TP01/essais/premier`. Pourquoi la commande échoue-t-elle ? Cherchez dans `man mkdir` une option qui permet de créer les répertoires parents manquants, puis réessayez.

(e) Copiez le répertoire `TP01` sous le nom `TP01b`. Essayez d'abord `cp TP01 TP01b`, puis cherchez dans `man cp` comment corriger l'erreur.

(f) Essayez de supprimer `TP01b` avec `rm TP01b`. Pourquoi cela échoue-t-il ? Cherchez comment supprimer ce répertoire et son contenu. **Attention :** vérifiez le nom avant toute suppression.

Exercice 2 (Comprendre les chemins)

(a) Placez-vous dans `~/Linux/TP01/essais/premier`. Sans lancer de commande, prédisez le résultat de `pwd` après `cd ..`, puis vérifiez.

(b) Depuis `~/Linux/TP01`, donnez deux commandes différentes pour aller dans `~/Linux` : l'une avec un chemin relatif, l'autre avec un chemin absolu (ou commençant par `~`).

(c) Dans `~/Linux`, essayez de compléter `cd TP0` avec la touche Tab. Que se passe-t-il ?

(d) Créez un fichier `notes.txt` dans `TP01`, copiez-le sous le nom `copie.txt`, puis renommez cette copie `brouillon.txt`. Quelle différence observez-vous entre `cp` et `mv` ?

Attention

Dans `~/Linux/TP01`, téléchargez le jeu de données, puis extrayez-le dans le répertoire `data` :

```
cd ~/Linux/TP01
wget http://igm.univ-mlv.fr/~alabarre/teaching/shnu/linux/dataset01.zip
unzip dataset01.zip -d data
cd data
```

Exercice 3 (Explorer avec `ls`)

(a) À l'aide du manuel, trouvez une option de `ls` permettant de classer les fichiers par taille décroissante. Quel est le plus gros fichier ? (Résultat attendu : `4U6LA6BKJC`.)

(b) Trouvez une option de `ls` permettant de classer les fichiers du plus récemment modifié au plus ancien. Quel est le premier ? (Résultat attendu : `ZHJQ4KZJOP`.)

(c) Que montre `ls -a` que `ls` ne montre pas toujours ?

Exercice 4 (Caractères génériques : `*` et `?`)

Quand on écrit une commande, le terminal peut remplacer des caractères génériques par des noms de fichiers : `*` représente une suite de zéro ou plusieurs caractères, et `?` représente exactement un caractère. Par exemple, `ls A*` affiche les noms commençant par `A`. Ce ne sont pas des expressions régulières.

(a) Affichez les fichiers dont le nom commence par `A`. (Attendu : `A53U4JFMKB` et `ATVXUWWCAN`.)

(b) Affichez les fichiers dont le nom se termine par `B`. (Attendu : `7GSNDNLREB` et `A53U4JFMKB`.)

(c) Affichez les fichiers dont le nom contient `D`.

(d) Affichez les fichiers dont le quatrième caractère est `4`.

(e) Affichez les fichiers dont l'avant-dernier caractère est `B`.

Exercice 5 (Pour aller plus loin (facultatif))

(a) Que fait `rmdir` sur un répertoire non vide ? Expliquez comment supprimer un répertoire vide sans utiliser `rm -r`.

(b) Consultez `man du` et trouvez comment afficher la taille totale de `~/Linux/TP01` dans un format lisible (Ko, Mo, etc.).

(c) Essayez `ls [AB]*` dans `data`. Quels noms sont sélectionnés ?

Exercice 6 (Prêt à l'envoi)

Vérifiez que `~/Linux/TP01/reponses.txt` contient vos réponses. Depuis `~/Linux`, créez une archive :

```
cd ~/Linux
tar -czf TP01.tar.gz TP01
ls -lh TP01.tar.gz
```

Renommez l'archive en respectant la convention donnée au cours (numéro étudiant, prénom français, TP01), puis envoyez-la à l'adresse indiquée en classe. **Ne supprimez pas votre répertoire TP01 après l'envoi.**

Défis supplémentaires (facultatifs)

Ces exercices sont destinés aux étudiants qui ont terminé le travail principal. Vous pouvez utiliser `man` pour découvrir de nouvelles options. Aucun rendu supplémentaire n'est demandé.

Exercice 7 (Défi : construire une arborescence)

Dans votre répertoire personnel, créez l'arborescence suivante **uniquement avec des commandes du terminal** :

```
defi/
|-- cours/
|   |-- linux/
|   `-- python/
|-- tp/
|   |-- tp01/
|   `-- tp02/
`-- archives/
```

(a) Créez tous ces répertoires. Essayez d'utiliser le moins de commandes possible.

(b) Créez un fichier vide `notes.txt` dans `defi/cours/linux`.

(c) Copiez ce fichier dans `defi/tp/tp01`, sous le nom `compte-rendu.txt`.

(d) Supprimez le répertoire `defi/cours/python`, sans supprimer les autres.

(e) Placez-vous dans `defi/tp/tp01`, puis revenez à votre répertoire personnel en une seule commande.

Exercice 8 (Défi : explorer sans se déplacer)

(a) Sans changer de répertoire courant, affichez le contenu de `/etc`.

(b) Affichez les fichiers cachés de votre répertoire personnel.

(c) Avec une seule commande `ls`, affichez le contenu de `/etc` et de `/tmp`.

(d) Cherchez dans `man ls` comment afficher les fichiers de `/etc` par taille décroissante.

Exercice 9 (Défi : sélectionner des noms de fichiers)

Créez un répertoire `~/defi/motifs` et placez-vous dedans. Créez ensuite les fichiers suivants :

```
rapport1.txt  rapport2.txt  rapport10.txt  rapportA.txt
notes1.txt   notes2.md     rapport.txt
```

Trouvez une commande `ls` qui affiche :

(a) Tous les noms commençant par `rapport`.

(b) Uniquement `rapport1.txt` et `rapport2.txt`.

(c) Tous les fichiers dont le nom se termine par `.txt`.

(d) Tous les fichiers dont le nom se termine par un chiffre suivi de `.txt`.

Remarque : dans le terminal, le motif `[0-9]` représente un caractère compris entre 0 et 9. Utilisez cette information pour afficher uniquement les fichiers dont le nom se termine par un chiffre suivi de `.txt`. Pour aller plus loin : retrouvez cette règle dans `man bash`, à la rubrique Pattern Matching.