

TP 1 - Les bases d'un système GNU / Linux

Tous les exercices de la séance doivent être réalisés uniquement à l'aide de la ligne de commande. Utilisez les commandes d'aides `help` et `man` pour connaître la syntaxe d'une commande et trouver les options permettant de répondre aux questions posées.

Remarque: certaines questions vous demandent d'expliquer les problèmes rencontrés. **Ne vous contentez pas de copier / coller les messages d'erreurs: comprenez-les et expliquez-les.**

Cheatsheet 1: `man`

`man` commande ouvre la page du manuel de commande. Ensuite:

- utilisez les touches `↑`, `↓`, `PgUp` et `PgDn` pour naviguer dans le texte;
- pour chercher du texte (équivalent de `Ctrl+F`): tapez `/`, le texte désiré, puis `↵`. On peut également naviguer entre les occurrences:
 - `n` vous emmène à l'occurrence suivante (**n**ext);
 - `b` vous emmène à l'occurrence précédente (**b**ack).
- `q` vous permet de sortir du manuel (**q**uit);
- `h` affiche l'aide du manuel (**h**elp).

Échauffement

Dans les exercices suivants, nous créerons:

- des répertoires avec la commande `mkdir` (exemple: `mkdir repertoire`);
- des fichiers vides avec la commande `touch` (exemple: `touch nouveau_fichier`).

Exercice 1 (Répertoires et fichiers).

1. utilisez la commande `mkdir ~/Linux/TP01`; que se passe-t-il? Pourquoi?
2. trouvez dans le manuel de `mkdir` l'option permettant de régler le problème et réessayez;
3. utilisez la commande `cd` pour vous placer dans le répertoire `~/Linux/`;
4. utilisez la commande `touch` pour créer un fichier vide nommé `premier_fichier` dans `TP01`;
5. essayez de copier `TP01` vers un nouveau répertoire `TP01b` à l'aide de la commande `cp`; que se passe-t-il? Pourquoi?
6. trouvez dans le manuel de `cp` l'option permettant de régler le problème et réessayez;
7. essayez de supprimer le répertoire `TP01b` avec la commande `rm`; que se passe-t-il? Pourquoi?
8. trouvez dans le manuel de `rm` l'option permettant de régler le problème et réessayez.

Fichiers et permissions

Les questions suivantes nécessitent de manipuler des fichiers texte. C'est l'occasion de se familiariser avec `nano`, un éditeur de texte en ligne de commande très simple.

Cheatsheet 2: nano

Pour ouvrir un fichier, utilisez `nano nom_fichier` (ou juste `nano` pour lancer le programme). Les raccourcis suivants seront utiles:

- `Ctrl` + `O`: sauvegarder le fichier;
- `Ctrl` + `X`: quitter le programme;
- `Ctrl` + `K`: couper la ligne actuelle;
- `Ctrl` + `U`: coller ce qu'on a coupé;
- `Ctrl` + `C`: annuler l'action en cours;
- `Ctrl` + `G`: affiche l'aide de `nano` (q pour en sortir).

Exercice 2 (Exécution de programmes). `mkdir` permet de créer un répertoire.

1. placez-vous dans le répertoire `TP01`, créez-y le fichier `fichier01` avec `nano`, et écrivez-y la commande suivante:

```
for i in $(seq 5); do echo "Bonjour du shell -- numéro $i"; done
```

Essayez d'exécuter le programme résultant avec la commande `./fichier01`; que se passe-t-il? Pourquoi?

2. rendez `fichier01` exécutable à l'aide de la commande `chmod` et réessayez; que se passe-t-il?

Pour les exercices suivants, nous allons utiliser des données générées aléatoirement. Toujours dans `~/Linux/TP01`, récupérez le fichier `dataset01.zip` sur la page du cours à l'aide de la commande suivante:

```
1 $ wget http://igm.univ-mlv.fr/~alabarre/teaching/shnu/linux/dataset01.zip
```

Extrayez son contenu dans un nouveau répertoire `data` avec la commande:

```
1 $ unzip dataset01.zip -d data
```

Exercice 3 (1s). Utilisez les commandes données dans le répertoire `data` pour écrire des instructions répondant aux questions suivantes.

1. avec `ls`: quel est le fichier le plus gros du répertoire actuel? (vous devriez trouver le fichier `5DSB6A6MGF`)
2. avec `ls`: quel fichier du répertoire actuel a été modifié le plus récemment? (vous devriez trouver le fichier `5RCFOJZ3LH`)
3. que fait l'option `-a`?

Exercice 4 (Caractères génériques). La ligne de commande permet d'utiliser des *caractères génériques* (ou *wildcards*): le caractère “?” représente n'importe quel caractère, et le caractère “*” représente n'importe quelle *suite* de caractères. Utilisez `ls` et ces caractères génériques pour afficher:

1. tous les fichiers dont le nom commence par 5;
(résultat attendu: 57VHGBS66F 5DSB6A6MGF 5L7BYH60YK 5RCF0JZ3LH)
2. tous les fichiers dont le nom se termine par 5;
(résultat attendu: CPKIQZTYQ5 E7056VJ4S5 IE7L6PIDK5)
3. tous les fichiers dont le nom contient A;
(résultat attendu: 5DSB6A6MGF AYR5I03Q3C CJ6JWHPWTA XBG5AE2AWK)
4. tous les fichiers dont le nom contient un 3 en troisième position;
(résultat attendu: IZ3M7TRPOY NT3FTC27L2)

Installation

Exercice 5 (`most`). Pour rendre les pages du manuel plus lisibles, installez le package `most`. Ensuite, tapez la commande `export PAGER=most`. Les pages de manuel sont maintenant en couleurs.

Exercice 6 (Commandes en séquence). On peut exécuter plusieurs commandes en séquence de deux façons différentes: en les séparant par des `;` ou par des `&&` (l'opérateur logique `and`, comme en C et en Python).

1. exécutez les commandes suivantes dans le terminal, et expliquez les différences entre les résultats obtenus:

```
1 $ ls fichiermanquant; echo "Bonjour"
2 $ ls; echo "Bonjour"
3 $ ls fichiermanquant && echo "Bonjour"
4 $ ls && echo "Bonjour"
```

2. écrivez une combinaison de commandes permettant en une ligne de mettre à jour les sources de `apt` et d'installer les mises à jour s'il y en a. Vaut-il mieux utiliser `;` ou `&&`? Pourquoi?

Cheatsheet 3: bash

`bash` propose les raccourcis clavier suivants, qui vous permettront de gagner un peu de temps:

- `↩`: déclenche la complétion automatique;
- `↑`: affiche la dernière commande exécutée, utilisable plusieurs fois;
- `↓`: le contraire de `↑`;
- `Ctrl` + `W`: efface le mot précédant le curseur;
- `Ctrl` + `Del`: efface le mot suivant le curseur;
- `Ctrl` + `←`: va au “mot” précédent dans la ligne;

- `Ctrl` + `→`: va au “mot” suivant dans la ligne;
- `Ctrl` + `C`: annule la saisie en cours;
- `Ctrl` + `D`: déconnecte l'utilisateur du terminal et le ferme;

La commande `history` affiche toutes les commandes qui ont été tapées dans la session actuelle. `~` désigne votre répertoire personnel (donc `/home/pingouin` si votre nom d'utilisateur est `pingouin`).

Faire connaissance avec son système

Utilisez la commande suivante dans le répertoire `~/Linux/TP01` (nous expliquerons plus tard son fonctionnement):

```
1 $ touch $(whoami)-$(date -I)-config.txt
```

Cette commande crée un fichier dont le nom est votre nom d'utilisateur, suivi de la date d'aujourd'hui au format international, suivie de `-config.txt`. Nous allons y enregistrer les informations qui nous intéressent, à l'aide de deux moyens, que nous expliquerons plus tard:

1. pour ajouter le résultat de la commande `cmd` à la fin de `mon_fichier`, utilisez:

```
1 $ cmd >> mon_fichier
```

2. pour ajouter le contenu d'une ou plusieurs variables à la fin de `mon_fichier`, utilisez:

```
1 $ echo "$var" >> mon_fichier
2 $ echo "$var1", "$var2" >> mon_fichier
3 # (ajoute la ligne "valeur de var1, valeur de var2" à mon_fichier)
```

Pour les exercices suivants, nous enregistrerons dans le fichier créé plus haut les résultats des commandes. **Testez vos réponses avant de les rajouter au fichier!**

Exercice 7 (Informations sur le système). Pour chacune des questions suivantes, trouvez la **commande** qui permet de répondre à la question posée (par exemple: si l'on vous demande le chemin vers votre répertoire personnel, n'écrivez pas `/home/mon_nom/`; trouvez la commande qui affichera `/home/mon_nom/`), et rajoutez le résultat au fichier créé plus haut.

1. quels sont votre nom d'utilisateur, votre numéro d'utilisateur, et votre répertoire personnel?
2. quelle est la version de `bash` que vous utilisez?
3. quels sont le nom de votre machine et son type (`amd64`, `x86_64`, ...)?
4. utilisez la commande `df` pour afficher l'espace disponible, utilisé et restant dans votre répertoire personnel sous une forme lisible (par exemple: `219M` au lieu de `224256`), et sous la forme d'un pourcentage.

Exercice 8 (Découvrir le matériel).

1. Utilisez `df` pour afficher l'espace disponible, utilisé et restant sur votre disque dur, sous une forme lisible (par exemple: 219M au lieu de 224256), et sous la forme d'un pourcentage. Le résultat attendu ressemble à ceci:

```
1 total          1,0T  356G  620G  37% -
```

2. Utilisez `free` pour afficher la mémoire RAM disponible et utilisée sur votre système, sous une forme lisible.
3. Utilisez la commande `du` pour afficher, sous forme lisible, uniquement l'espace consommé par le répertoire actuel.

Exercice 9 (Archivage). Le programme `tar` permet de créer une archive au format `nom.tar.EXT`, où `EXT` est une extension qui dépend du format de compression utilisé.

1. quels sont les formats de compression proposés par `tar`?
2. remontez dans votre répertoire `~/Linux`, et pour chaque format disponible, créez une archive de votre répertoire `TP01` (attention, vous devez préciser vous-même le nom de l'archive, y compris son extension). Quel format occupe le moins de place?

programme qui lance plusieurs scripts aléatoires, seul l'un d'entre eux est "mauvais"; il faut détecter lequel et tuer seulement celui-là