

Système

L'objectif de la séance de TME est de manipuler la distribution DEBIAN du système d'exploitation Linux en se familiarisant avec la ligne de commande.

Pour cela, ouvrez un terminal depuis votre environnement préféré. Si vous travaillez sous l'environnement GNOME, utilisez l'application **Terminal** accessible depuis le menu APPLICATIONS → ACCESSOIRES. Similairement, si vous êtes sous l'environnement KDE utilisez l'application **Émulateur de terminal** (ou **Konsole**) accessible depuis le menu APPLICATIONS → UTILITAIRES.

1 Travail en ligne de commandes

1.1 Commandes de base dans le terminal

Exercice 1

Question 1 Tapez la commande `ls` suivi de la touche <Entrée>.

Question 2 Entraînez-vous à copier-coller du texte dans la fenêtre bash :

1. Boutons copier / coller dans le menu édition
2. Sélection du texte à la souris (mise en surbrillance) pour copier et clic avec le bouton central de la souris (molette) pour coller
3. Tapez la commande `ls -l` suivi d'un nom de fichier copié à l'aide de la commande `ls` précédente, suivi de la touche <Entrée>
4. Tapez la commande `pwd` pour déterminer le répertoire dans lequel vous vous trouvez
5. Tapez la commande `cd /` puis `ls` et enfin `cd ~` puis `pwd`. Commentez ce que vous observez.

1.2 Parcours de l'arborescence des fichiers en ligne de commandes

Exercice 2

Question 1 Tapez la commande suivante `nautilus &` (sous GNOME) ou `dolphin &` (sous KDE) dans le terminal et validez. Qu'observez-vous ?

Question 2 Tapez la commande suivante `gnome-search-tool &` (sous GNOME) ou `kfind &` (sous KDE) dans le terminal et validez. Qu'observez-vous ?

Question 3 Saisir les commandes suivantes en alternance avec la commande `pwd` (pour déterminer dans quel répertoire vous êtes) :

1. `ls *`, `ls .`, `ls ..`, `ls ../../..`, `ls ~` et `ls ~login` (où `login` représente un numéro d'étudiant valide), `ls /usr`, `ls /usr/*`, `ls /u*r/*`, `ls /u*/*`, `ls /usr/X*/b*/x[c-f]*`,
2. `cd`, `cd .`, `cd ..`, `cd ../../..`, `cd /`, `cd ~` et enfin `cd ~login` (où `login` représente un numéro d'étudiant valide)

1.3 Gestion des droits en ligne de commande

Exercice 3

Question 1 Aller dans votre répertoire principal et créer un nouveau répertoire nommé `Essai2`.

Question 2 Comme précédemment interdire l'accès aux membres de votre groupe ainsi qu'aux autres utilisateurs pour le répertoire `Essai2`.

Question 3 Modifier les droits d'accès du répertoire `Essai2` pour que personne ne puisse y accéder (en lecture, en écriture et en exécution).

1.4 Complétion automatique des commandes

Introduction BASH (l'interprète de commandes intégré aux applications `Terminal` et `Konsole`) aide à compléter les commandes, le chemin ou le nom d'un répertoire ou fichier que vous êtes en train de taper. Pour cela, il suffit de commencer à taper le début de votre commande, chemin ou nom et d'appuyer ensuite sur la touche `<TAB>` :

- s'il n'y a aucune ambiguïté, `bash` trouvera la solution et complètera votre commande, chemin ou nom.
- Dans le cas contraire, il y a zéro ou plusieurs solutions possibles. Appuyer une deuxième fois sur la touche `<TAB>`. S'il y a plusieurs solutions, `bash` vous en proposera ou bien affichera le nombre de possibilités si celui-ci est grand. Sinon, `bash` ne fait rien.

Exercice 4

Dans les questions suivantes, vous observerez le comportement de complétion de commande de `bash` en appuyant plusieurs fois sur `<TAB>` si nécessaire et indiquerez s'il y a 0, 1 ou plusieurs commandes possibles. Comment décider du nombre de commandes / chemins existants ?

Question 1 Pour les commandes suivantes :

1. `"l"<TAB>` ("`l`" comme "`L`" ucie en minuscule) puis `"s" <TAB>`
2. `"x" <TAB>` puis `"c" <TAB>` puis `"l" <TAB>` puis `"o" <TAB>`

Question 2 Pour les noms ou les chemins :

1. `"ls /us" <TAB>`, puis `"b" <TAB>`, ..., `"konquer" <TAB>`
2. `"ls /us" <TAB>`, puis `"loc" <TAB>`, ..., `"f" <TAB>`, puis `"f" <TAB>`
3. `"ls /usr/X*" <TAB>`
4. `"ls /usr/X*R6" <TAB>`
5. `"ls /usr/X*R*" <TAB>`
6. `"ls /usr/X??R*" <TAB>`
7. comparer `"ls /usr/X" <TAB>` puis `"/b" <TAB> <TAB>` `"[d-s]*"` et `"ls /usr/X*/b*/mkfont[d-s]*"`
8. `"ls /usr/X" <TAB>` puis `"/b" <TAB> <TAB>` `"[a-cs-z]*"`
9. `"ls /usr/X*R" <TAB>` ... `"6" <TAB>`

2 Utilisation de commandes avancées

2.1 Commande `cut`

Exercice 5

Soit le fichier `proverbe.txt` contenant le texte suivant :

*Rien ne sert de courir
Il faut partir à point.
La Fontaine.*

Question 1 créer le fichier `proverbe.txt` dans votre répertoire personnel.

Question 2 donner la commande pour n'afficher que les 5 premiers caractères de chaque ligne

Question 3 donner la commande pour n'afficher que les 2 premiers mots de chaque ligne

Question 4 quel sera le résultat de la commande : `cut -f 2 -de proverbe.txt` ?

2.2 Commandes tail et sort

Soit le fichier `pantheon.csv` contenant le texte suivant :

```
Nom; Prénom; Naissance
Page; Jimmy; 1944
Van Halen; Eddie; 1955
Vaughan; Stevie Ray; 1954
Clapton; Eric; 1945
Lagrène; Birelli; 1966
Beck; Jeff; 1944
McLaughlin; John; 1942
Vai; Steve; 1960
Hendrix; Jimi; 1942
```

Exercice 6

Question 1 Donner la commande qui permet d'obtenir le fichier `liste.csv` qui est une copie de `pantheon.csv` sans la première ligne.

Question 2 Donner la commande pour obtenir un affichage de la liste des guitaristes par noms croissants.

Question 3 Donner la commande pour obtenir un affichage de la liste par dates décroissantes.

3 Variables d'environnement

On poursuit dans cet exercice l'exercice portant sur les variables d'environnement du TD3. Si celui-ci n'a pas été abordé, commencez par le traiter avant de poursuivre sur les questions suivantes.

Exercice 7

Question 1 En utilisant la variable `HOME`, écrire une commande pour afficher la liste de tous les répertoires dont le père est commun avec votre répertoire personnel.

Question 2 Modifier la commande précédente pour afficher la liste de tous les utilisateurs dont le nom comporte un ou plusieurs 3.

Question 3 Modifier la commande précédente pour afficher la liste de tous les id des utilisateurs dont le nom comporte un ou plusieurs 3 triée en ordre décroissant.

4 Écriture de scripts simples

Exercice 8

Question 1 Rappeler ce qu'est un script.

Question 2 Ecrire un script qui ne prend pas d'arguments et qui renvoie le nombre de dossiers ou fichiers de votre répertoire personnel contenant le motif "jpeg".

Question 3 Transformer le script précédent en un script qui accepte pour argument le chemin absolu d'un répertoire à analyser.

Question 4 Ecrire maintenant un script acceptant 2 arguments : un nom de fichier et un motif et qui renvoie le nombre de lignes de ce dernier dans lesquelles apparaît le motif.

5 Expressions rationnelles

Nous étudions ici quelques expressions rationnelles (aussi appelées "régulières" du fait de leur nom en anglais **regular expression**).

Exercice 9

On supposera que le répertoire de travail est `/usr/include/`.

Question 1 Lire la page `man` de la commande `egrep` et indiquer son rôle.

Question 2 Utiliser la commande `egrep` (syntaxe : `egrep -n <motif> <fichier>`) avec les motifs suivants, en terminant à chaque fois par la touche `<Entrée>` :

1. `ff` : avec comme fichiers `*.h`, puis `X.h`. Les noms de fichiers sont aussi affichés.
2. `[0-9]` : avec comme fichier `X.h`
3. `0x[a-f0-9A-F]+` : idem
4. `0x[a-f0-9A-F]` : ce motif est suffisant pour donner le même résultat que le motif précédent.
5. `0x[a-f0-9A-F]+` : avec comme fichier `*.h`
6. `0x[a-f0-9A-F]` : idem.

6 Processus, commande `ps` et `kill`

Exercice 10

Question 1 Lancer le terminal (ci-après la fenêtre 1) (taper la commande `terminal &` ou `konsole &`) et taper la commande `ps -Al` pour afficher en format long tous les processus. Repérer le processus `bash` correspondant à cette instance du terminal en cours d'exécution et dessiner le sur un brouillon comme le nœud d'un arbre avec son `PID` et son `PPID`.

Question 2 À partir de la fenêtre 1, ouvrir un nouveau terminal comme précédemment pour créer une nouvelle fenêtre (ci-après la fenêtre 2). Taper la commande `ps -Al`. Repérer les nouveaux processus de terminal et `bash` engendrés et compléter l'arbre ci-dessus avec ces nouveaux nœuds.

Question 3 Dans la fenêtre 2, tuer le processus `bash` ou terminal de la même fenêtre avec la commande `kill -9 PID-du-processus`.

Question 4 Recommencer tout le processus de création des deux fenêtres et depuis la seconde arrêter le `bash` de la première. Que remarquez-vous ?