

# Informatique S1 – Programmation C

## Exercices - Corrigés

### TD 1 : Prise en main de l'environnement

#### Solution Exercice 1

Objectif : prise en main de l'environnement graphique et de l'éditeur *gedit*.

L'idée est d'inviter les étudiants à l'usage du système Linux, en le démystifiant. Ils doivent prendre en main l'éditeur *gedit*, se familiariser avec lui et avec le mode de coloration. On doit s'assurer qu'ils enregistrent leur fichier *helloworld.c* dans le répertoire home de l'utilisateur.

Dans la question sur l'usage du navigateur, profiter pour rappeler l'importance de l'usage de l'adresse de courrier électronique de l'université. S'assurer que les étudiants savent envoyer un fichier en pièce jointe par le webmail de l'université (ça peut nous servir pour la suite).

#### Exercice 2

Objectif : initier les étudiants à l'usage de la ligne de commande et à l'utilisation des commandes simples.

- a) Menu applications -> Accessoire -> terminal
- b) Pour trouver le répertoire courant, utiliser la commande « `pwd` ». Le chemin complet doit être quelque chose comme « `/home/user` ».
- c) Afficher le contenu avec la commande « `ls` »
- d) Commande : « `mkdir code` »
- e) Commandes : « `cd code` » suivi de « `ls -la` ». Comme le répertoire est vide, `ls -la` n'indique que « `.` » et « `..` ». Expliquer (à l'aide du guide de référence) le signifié de chaque.
- f) Utiliser la commande « `pwd` » pour voir le chemin complet.
- g) Faire observer l'usage du « `..` » pour référencer le répertoire précédent.
- h) L'objectif de cet exercice est d'indiquer aux étudiants que les lettres majuscules et minuscules ne sont pas pareilles sous Linux. S'ils ont bien enregistré le fichier *helloworld.c* avec ce nom et sur le home de l'utilisateur, la première commande ne doit pas marcher, tandis que la deuxième doit afficher le contenu du fichier.
- i) Commande : `cp helloworld.c HelloWorld.c`
- j) Maintenant qu'on a créé un fichier « `HelloWorld.c` », la commande `cat` de la question h doit marcher. Renforcer le fait que les lettres majuscules et les minuscules sont différentes.
- k) Commande : `mv HelloWorld.c /home/user/code` (ça peut varier selon le répertoire)

- l) Exercice simple pour faire comprendre aux étudiants qu'un répertoire doit être vide avant de le supprimer. (conseil : éviter quand-même de les indiquer l'usage de « `rm -rf` » dans un premier moment, ça cause de dégâts facilement)

### Exercice 3

- a) L'usage de l'USB sur Linux est toujours un peu délicat. L'objectif ici est permettre aux étudiants de le prendre en main (pour que dans les séances suivantes, ils puissent enregistrer sur leur clé USB leurs programmes). Si le groupe paraît avancé, il existe la possibilité de leur expliquer comment Linux voit les dispositifs (tout est un fichier sur Linux).
- b) Commande : `cp -r code /media/USB\ DISK` (ça peut varier selon le *moint point*)
- c) L'objectif ici est d'indiquer qu'il faut déconnecter la clé USB (tel qu'on le fait sous Windows). On peut le faire à l'aide de l'interface graphique (clique droit sur l'icône qui normalement doit apparaître au bureau, option « démonter le volume ») ou avec la commande « `umount /media/USB\ DISK` » (ça peut varier en fonction du *moint point*). Attention : l'étudiant ne peut pas avoir d'application ouverte (éditeur de texte, terminal...) utilisant la clé pour qu'elle soit déconnectée.

### Exercice 4

- a) La commande « `which` » indique le chemin utilisé lors de l'exécution d'une commande donnée, tandis que la commande « `whereis` » indique le chemin où on peut trouver cette commande (éventuellement les répertoires qui lui font référence).
- b) L'objectif de cet exercice est de le faire comprendre la redirection de la sortie avec « `>` » et « `>>` ».
- c) L'objectif ici est de présenter l'usage du « `\` » pour pouvoir écrire une commande sur plusieurs lignes. Deuxième objectif est de le faire consulter le guide de référence, dans lequel il y a un exemple de « `\` » qui utilise la commande « `cat` » (en somme, la solution de l'exercice est là pour ceux qui voudront bien lire le guide).

Par ailleurs, pour que le prochain exercice soit clair, il faut qu'ils exécutent la redirection (`>>`) sur le fichier « `test_redir.txt` » au moins trois fois, afin d'avoir suffisant de contenu pour que l'exécution des commandes « `more` » et « `less` » soit sensiblement différente de celle de « `cat` ».

- d) L'idée ici est le montrer que « `more` » et « `less` » présente un fichier page par page, contrairement à « `cat` » que le fait défiler sans arrêt à l'écran. Attention : les commandes « `more` » et « `less` » ne sont pas toujours disponibles (parfois seulement un ou l'autre est disponible).
- e) Exemple de redirection avec le « `|` » (*pipeline*). Les étudiant doivent comprendre qu'on peut exécuter deux commandes de suite, en faisant la redirection de la sortie du premier (le texte afficher par `cat`) vers l'entrée du second (le `more/less`) avec le pipeline.
- f) L'objectif ici est de montrer la commande `man` comme une aide importante.

- g) L'idée ici est de montrer l'usage du joker « \* » (*mask*).
- h) Usage de la commande « zip » pour la création d'archives. Faire remarquer aux étudiants qu'ils peuvent envoyer par mail les programmes qu'ils auront créés pendant les séances de TD plus facilement s'ils envoient un seul fichier zip (au lieu de plusieurs fichiers .c).
- i) L'objectif ici est qu'ils puissent voir une autre fonctionnalité utile de gedit (menu Rechercher→Aller à la ligne) et de préparer l'exercice suivant.
- j) Les étudiants doivent utiliser l'option « -l » avec la commande « unzip » (exemple sur le guide de référence) pour éviter d'écraser leurs fichiers. Il faut qu'ils soient conscients que la commande « unzip » ne va pas poser des questions avant d'écraser un fichier.

## Exercice 5

L'objectif de cet exercice est d'abord de montrer les effets de la compilation : la création d'un fichier binaire « a.out » dans le premier cas, « helloworld » dans le second. C'est également l'opportunité de leur parler que les programmes ne s'appellent pas forcément « quelquechose.exe » sous Linux, qu'ils peuvent avoir n'importe quel nom.