

TP1 : Introduction en Linux et rappels de C

Exercice 1: Dans cet exo nous allons explorer l'interaction avec Linux *en ligne de commande*. Utiliser la commande `man` est fortement conseillé!

1. Connectez-vous sur votre machine, lancez un "terminal virtuel" (`xterm` ou `terminal`) et tapez `pwd`. Qu'est-ce que cela vous indique? Tapez ensuite `cd ..` et encore une fois `pwd`. Où vous trouvez-vous? Pour revenir dans votre répertoire "maison" (*home directory*), tapez `cd` (sans arguments).
Pour répondre aux questions de ce point, vous pouvez ouvrir une fenêtre avec votre répertoire d'accueil et naviguer dans celui-ci, en parallèle avec les commandes données dans le terminal.
 2. Listez le contenu du répertoire maison, avec la commande `ls`, ainsi que le contenu du répertoire parent de votre répertoire maison et celui du répertoire racine (`/`). Changez de répertoire dans le répertoire "racine" / (*root*). Affichez les entrées dans ce répertoire.
 3. Lancez en exécution un éditeur de texte (`kedit` ou `gedit` ou `kwrite` ou `emacs`), en tapant le nom du programme et `<enter>`. Est-ce qu'on peut encore lancer de commandes dans le terminal? Tapez `CTRL-C` dans le terminal – que se passe-t-il? Lancez le même éditeur en exécution maintenant en rajoutant le `&` à la fin du nom du programme, avant le `<enter>` – y a-t-il une différence?
 4. Créez, avec `mkdir`, dans votre **répertoire maison**, le sous-répertoire `progimp`. Créez un sous-répertoire `tp1` dans le répertoire `progimp`. Créez aussi un fichier dans ce répertoire en utilisant un éditeur de texte. Trouvez la taille des fichiers créés, avec la commande `ls -l`. Expérimentez d'autres options de la commande `ls`, trouvées dans la page du manuel de la commande, que vous pouvez afficher avec `man rm`.
 5. Supprimez les fichiers et répertoires que vous avez créé (sauf le `tp1`) en utilisant la commande `rm`. Pour comprendre son fonctionnement et les options qu'on peut lui passer en ligne de commande, lisez la page du manuel de `rm` avec la commande `man rm`.
-

Exercice 2: Transcrivez les deux programmes suivants à l'aide d'un éditeur de texte, puis compilez-les, exécutez-les et confirmez le fonctionnement discuté en TD. Utilisez le compilateur `gcc`.

```
#include<stdio.h>
main(){
    int a=1000;
    char b;
    b=a;
    printf("a=%d\nb=%d\n",a,b);
}

#include<stdio.h>
main(){
    char a=100;
    int b;
    b=a;
    printf("a=%d\nb=%d\n",a,b);
}
```

Pour rappel, l'option `-o nomdefichier` de `gcc` vous permet de donner le nom voulu à votre exécutable. Pour lancer en exécution l'exécutable, placez-vous (à l'aide de `cd`) dans le répertoire où se trouve l'exécutable, puis lancez-le en tapant `./nomdelexecutable`.

Exercice 3: Écrivez un programme C qui lit un octet et affiche ses bits, un par ligne, compilez-le, corrigez les erreurs et vérifiez son fonctionnement correct.

Rappel : L'opérateur `/` calcule le quotient d'une division entière, l'opérateur `%` calcule le module d'une division entière et l'opérateur de décalage de bits est le `>>`.

Exercice 4:

- Écrire une fonction C qui prend en paramètre une chaîne de caractères et renvoie la taille de la chaîne. (Pour rappel : la fin d'une chaîne de caractères est marquée par un octet de valeur 0).
 - Écrire un programme C qui lit une chaîne de caractères (avec `scanf` format `%s` ou avec `gets`), et affiche chaque caractère composant cette chaîne et son code ASCII (sous la forme caractère `i = ...`, code `ascii = ...`). Le programme devrait utiliser la fonction écrite au point précédent.
-

Exercice 5: (Si le temps le permet) Transcrivez les programmes créés pour les exos 3 et 5 de la feuille de TD1, compilez-les, lancez-les en exécution et vérifiez leur fonctionnement.
