

# Chapitre 5

## TP Langage de commande II

### I Exercices

Réalisez à l'aide de tubes les opérations suivantes :

1. comptez le nombre de fichiers se trouvant dans le répertoire courant,
2. comptez le nombre d'utilisateurs connectés,
3. en utilisant le fichier `/etc/passwd`, donnez la liste, classée par ordre alphabétique, des utilisateurs pouvant travailler en bash,
4. affichez la liste des processus de tous les utilisateurs, classée par nom d'utilisateur.

### II Manipulation de fichiers et de programmes

Les éléments sur les éditeurs joe et emacs sont minimalistes. Ils doivent juste vous permettre d'éditer, enregistrer et quitter. Leur maîtrise viendra avec le temps. Vous pouvez en choisir d'autres : vi, nedit ou en environnement graphique : kwrite, gedit, kate...

#### 1. L'éditeur Joe

Pour obtenir de l'aide `CTRL k h`

TAB. 5.1: Les commandes de base

Rechercher	CTRL k f
Rechercher suivant	CTRL k l
Copier un block	
Début de block	CTRL k b
Fin de block	CTRL k k
Copier le block	CTRL k c

*Suite page suivante*

Déplacer le block	CTRL k m
Supprimer le block	CTRL k y
Ecran précédent	CTRL u
Ecran suivant	CTRL v
Début de document	CTRL k u
Fin de document	CTRL k v
Début de ligne	CTRL a
Fin de ligne	CTRL e
Sauvegarder et quitter	CTRL k x
Sauvegarder	CTRL k d
Lire un fichier	CTRL k e
Insérer un fichier	CTRL k r
Accéder au shell	CTRL k z (Tapez fg pour revenir)
Quitter	CTRL x c

## 2. L'éditeur Emacs

TAB. 5.2: Les commandes de base

C-x s	Sauvegarder
C-x c	Quitter

# III TP

## 1. Le compilateur gcc

Vous allez utiliser le compilateur C de GNU, `gcc` pour la programmation en C ANSI. C'est également un compilateur C++. Voici les principales commandes :

```
gcc -o hello hello.c
```

S'il n'y a pas d'erreurs de compilation, `gcc` génère un exécutable `hello`. Le compilateur a généré tout d'abord un objet (fichier objet `hello.o`), puis a réalisé l'édition de lien, pour obtenir l'exécutable.

```
gcc -c hello.c
```

donne le fichier objet

```
gcc -o resultat premier.c deuxieme.c
```

Cette commande montre qu'il est possible de compiler plusieurs programmes sources C, en un seul fichier résultat exécutable.

```
gcc -O -o hello hello.c
```

Cette commande lance l'optimiseur de code de `gcc`. Vous pouvez également voir l'option `O2`. Le programme est plus long à compiler, mais le code sera plus compact et plus rapide.

```
gcc -g -o hello hello.c
```

Active toutes les informations de débogage du programme. Cette option n'est à utiliser qu'en phase de programmation et de test.

## 2. Vérification de votre environnement

Linux offre cette possibilité (et cette richesse aussi) de pouvoir disposer sur la même machine de plusieurs versions d'une même application (ici gcc). Vous pouvez avoir un gcc 2.x, gcc 3.x, gcc 4...

Pour éviter de compliquer les lignes de commandes, on utilise souvent gcc.

On crée donc un lien entre la commande gcc et le compilateur qu'on souhaite utiliser. Nous allons vérifier que ce lien existe bien. Vous allez taper dans une console la commande `gcc -help`, si vous avez une erreur

```
bash: gcc: command not found
```

nous allons la corriger. Si tout se passe bien vous pouvez lire (par curiosité) la suite ou sauter au paragraphe suivant.

Faites un

```
locate gcc
```

Cela vous produit plusieurs pages de résultat. Ce qui nous intéresse, c'est tout ce qui commence par `/usr/bin`. Vous devez avoir quelque chose du style `/usr/bin/gcc-4.0`. Il ne reste plus qu'à créer le lien sur ce binaire avec une commande gcc.

```
sudo ln -s /usr/bin/gcc-4.0 /usr/bin/gcc
```

C'est terminé. Chaque fois que vous taperez gcc, le système lancera `/usr/bin/gcc-4.0`.

## 3. Compiler un programme

Cet exercice explique la réalisation puis la compilation d'un premier programme en C avec gcc. Avec un éditeur, tapez le code ci-dessous puis enregistrez le sous le nom `hello.c`.

```
#include <stdio.h>
int main() {
    printf ("Hello_World\n");
    return (0);
}
```

Vous allez réaliser la compilation avec la commande `gcc -o hello hello.c`. Vous pouvez exécuter le programme en tapant la commande `./hello`. Avec la commande `ls -al hello*`, vous pouvez voir que le programme `hello` est bien exécutable.