Licence 1 - section B
TP 4 d'éléments d'informatique

Catherine RECANATI – Département d'Informatique – Institut Galilée

Semaine du 28 novembre au 4 décembre 2016

L'objectif de ce TP est de vous présenter la commande gcc, et de vous faire adopter de bonnes habitudes concernant l'organisation des fichiers sources de vos programmes. Cela peut paraitre insignifiant aujourd'hui, mais dès que vous écrirez des programmes plus importants, cela sera indispensable.

Nous allons mettre tous les programmes de ce TP dans un répertoire appelé `TP4`. *Pour savoir comment créer de nouveaux répertoires et comment naviguer d'un répertoire à l'autre, reportez vous aux préambules des feuilles de TP précédentes. Vous pouvez aussi consulter le manuel utilisateur de Linux en utilisant la commande* `man` *(pour manuel), qui vous renseignera sur les commandes de manipulation de fichiers (cf.* `cp,mv,cd` *et* `rm`*). Ainsi,* `man mv` *vous expliquera la syntaxe et la sémantique de la commande* `mv`.

1 Compilation séparée de fonctions

Exercice 1.1 Compilation d'un programme utilisant une fonction compilée séparément.

1. Vous pouvez commencez par vous placer dans le répertoire TP4 pour éditer le texte source d'une fonction `fonct(int tab[], ...)` dans un fichier intitulé `fonct.c` (par exemple, écrire la définitions de la fonctions `initTabInt` du TD7 dans le fichier `initTabInt.c`).
2. Compilez ensuite ce fichier avec une commande gcc qui utilise l'option -c indiquant au compilateur de ne créer que des codes objets (.o) :
   `gcc -c fonct.c` (ou `gcc -Wall -c fonct.c` pour obtenir des *warning*). Ces commandes, une fois les erreurs corrigées, entrainent la création d'un fichier intitulé fonct.o.
3. Nous allons maintenant écrire un programme utilisant la fonction compilée séparément. Ce programme sera écrit dans un fichier nommé par exemple `main.c`. On incluera la déclaration de la fonction que l'on souhaite utiliser dans la fonction `main` au début du programme et on compilera le fichier main.c avec la commande :
   `gcc fonct.o main.c` (ou `gcc fonct.o main.c -o progexe`). On obtient alors un code exécutable dans le fichier `a.out` ou `progexe` respectivement.
4. On peut aussi n'avoir qu'un seul fichier comportant toutes les définitions de fonctions sur les tableaux, appelé tableaux.c, et constituer un fichiers de déclarations de ces fonctions (tableaux.h). Dites comment vous modifierez le programme main.c, et quelle commande gcc vous utiliserez pour compiler séparément tableaux.c, et créer ensuite un exécutable avec main.c.

Remarque : Si on a besoin de plusieurs fonctions sur les tableaux dans le programme principal, on peut aussi procéder de la même manière qu'en 3., en compilant séparément les fichiers fonction1.c fonction2.c ...fonctionN.c avec l'option -c de gcc, puis assembler tout le code avec la commande :
`gcc fonction1.o fonction2.o ... fonctionN.o main.c` [optionnel `-o progexe`].

Exercice 1.2 Application à l'exercice 2.1 de la feuille du TD8
On mettra en oeuvre la compilation des fonctions et programme définis dans la section 2 de la feuille de TD8.

2 Compilation d'une bibliothèque statique de fonctions

Le compilateur gcc vous permet aussi de compiler les bibliothèques (certains disent librairies, car bibliothèque se dit *library* en anglais).

Il existe deux types de bibliothèques : les bibliothèques statiques et les bibliothèques dynamiques. Nous ne parlerons ici que des bibliothèques statiques. (Pour des informations sur les bibliothèques dynamiques on pourra consulter le site de cours gratuit en ligne https ://openclassrooms.com/courses/compilez-sous-gnu-linux).

Exercice 2.1 Préparer une bibliothèque statique de fonctions sur les tableaux

Créez à partir de fonctions (que vous choisirez dans le TD7 ou le TD8) une bibliothèque statique de fonctions prenant en argument des tableaux d'entiers.

Indications : Une bibliothèque statique (généralement d'extension .a) est une bibliothèque qui sera intégrée à l'exécutable lors de la compilation. L'avantage est que l'exécutable produit est autonome et ne nécessite rien de plus pour fonctionner. La bibliothèque se comporte comme un autre fichier objet.

Si vous voulez faire une bibliothèque statique avec fonctTab.c et fonctTab.h, il suffit de faire :

```
gcc -c fonctTab.c -o fonctTab.o
ar -q libtableaux.a fonctTab.o
```

La première commande crée le fichier objet comme on l'a vu dans l'exercice précédent. La commande `ar` archive tout simplement ce fichier, comme son nom l'indique. (Pour plus de détails, lisez `man ar`).

Une bibliothèque se liera ensuite comme n'importe quel fichier objet :

```
gcc bidule1.o bidule2.o bidule3.o libtableaux.a -o Programme
```