***TIC – Filière Informatique***

**Jigé Pont**

**Nicolas Fuchs**

**Microprocesseur 3**

**Labo 1**



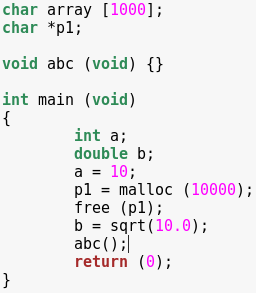
20/02/2018

# Configuration

Système d'exploitation : Kali Linux Rolling 64 bits  
Processeur : Intel Core i7-4510U CPU @ 2.00GHz

# Compilation

file.c



gcc -Wall -Wextra -c file.c -o file.o

Commande compilation :

Sortie console : 

Analyse : Aucune erreur n'est apparue par contre plusieurs warnings :

* Les variables a et b sont déclarées mais pas utilisées
* Déclarations implicites des fonctions malloc, free et sqrt

Ces warnings sont dus à l'absence de header qui déclare les prototypes des fonctions utilisées. Pour les supprimers, il suffit d'inclure les headers math.h et stdlib.h.

# Linker

gcc -Wall -Wextra file.o -o file

Commande linker :

Le linkage fonctionne indépendamment de l'inclusion des headers. Le linker connaît le chemin par défaut vers les librairies préinstallées.

# Options

gcc -Xlinker -Map=file.map -Wall -Wextra file.o -o file

Commande linker avec map file :

gcc -S file.c

Commande compilation avec sortie assembleur :

gcc -g -Wall -Wextra -c file.c -o file.o

Commande compilation avec information debugging :

Analyse : Le map file se nomme file.map, le fichier assembleur se nomme file.s et les informations de debugging sont incluses dans le fichier objet file.o qui est passé de la taille 1992 octets à 4120 octets.

Remarque : Le fichier file.map // TODO expliquer pourquoi ca fontionne par offset et pas par valeur adresse complète….

# Adresses et mémoire

L’adresse du début du code est l’adresse du début du code segment. Pour l’obtenir il suffit de prendre l’adresse de la fonction void abc(void)  via un printf (voir plus bas). Comme l’implémentation de la fonction abc est la première chose qui est faite dans le code (les deux déclarations de array et p1 sont des globales et iront donc dans le data segment) ce là correspond bien à l’adresse du début du code.

Les adresses des variables array, p1, a, b, abc et main ont été obtenues en faisant un printf :



Ce qui nous donne le résultat suivant :



Une représentation de la mémoire pour ce programme :

