SCIENCES SORBONNE UNIVERSITÉ

Travaux Pratiques

UE: MU5SP05

Lab. 10 : Le Makefile

1. Recopiez toujours les fichiers max.c, min.c, var.c, max.h, min,h, var.h et test de la partie 7.

2. Créer un makefile permettant de compiler les fichiers sources et de créer un binaire de sortie. Dans ce makefile, créer les cibles suivantes en activant à chaque fois l'option –Wall :

Install : permettant d'installer le binaire créer dans $HOME (normalement c'est /home/student dans votre cas)

My-binaire **:** crée l'exécutable final de ce projet

max.o

min.o

var.o

test.o

clean : permettant d'effacer les fichiers dont on n'aurait pas besoin pour créer le binaire.

3. Exécuter les commandes suivantes :

   - make
   - make install
   - make min.o
   - make clean

   Analyser ce qui ce passe pour chacune des étapes. Lorsqu'on ne précise pas de cible, qu'est ce qui se passe ?

4. Modifier le makefile en rajoutant les macro suivants.

   CC = gcc
   OBJ = max.o min.o var.o
   CC_OPTIONS = *// (Laisser vide pour le moment)*

5. Redéfinissez la macro CC_OPTIONS à –g-Wall sur la ligne de commande qui permet d'appeler le makefile. Vérifier que cette modification à été prise en compte.

6. Proposer un nouveau makefile en remplaçant le nom de la cible et la liste des dépendances par les variables automatiques. La liste des variables automatiques que vous avez à rajouter dans votre makefile est donnée comme suit :

   - $@ : nom de la cible ;
   - $< : première dépendance de la liste des dépendances ;
   - $^ : toutes les dépendances ;

7. Nous souhaitons pouvoir créer un binaire en mode debug ou en mode distribution release suivant le souhait de l'utilisateur. Définissez une variable DEBUG dans votre makefile et rajouter les modifications suivantes :

   - Si DEBUG = yes, la compilation sera en mode debug.
   - Sinon la compilation sera en mode release.

8. Tester vos modifications et vérifier le bon fonctionnement de votre makefile.

9. Afficher le mode de compilation utilisé en rajoutant la commande echo dans votre makefile.