

ASR=> Architecture des ordinateurs: Ass2

TP Assembleur ARM

1. Configuration logicielle

Le développement de l'application se fera sur les PC des salles de TP, avec des utilitaires **croisés** pour plateforme ARM (Compilateur C, Assembleur, Editeur de liens) sous le système Linux-Debian.

L'exécution et la mise au point du programme objet se feront sur le même PC en utilisant un simulateur intégré à l'outil de mise au point gdb. Il s'agit bien sûr ici d'une version croisée, comme pour les outils de développement.

2. Développement de l'application

2.1. Editeur de texte

Utiliser un éditeur de texte, par exemple **gedit**.

2.2. Compilateur croisé C pour ARM

arm-elf-gcc -g -c *f.c*

Le fichier *f.c* est le fichier source C à compiler.
Le compilateur génère un fichier objet *f.o*.

2.3. Assembleur croisé ARM

arm-elf-as -o *f.o* *f.s*

Le fichier *f.s* est le fichier source en Assembleur **ARM** à assembler.
L'assembleur génère un fichier objet *f.o*.

2.4. Editeur de liens croisé pour ARM

arm-elf-gcc *f1.o f2.o f3.o* -o *f*

L'éditeur de liens fusionne les divers fichiers objet (*.o*) produits par le compilateur C ou par l'assembleur. Il réalise ensuite les liens avec la bibliothèque standard C (libC) et génère enfin un fichier exécutable *f*.

3. Exécution et mise au point du programme objet

Tous les tests se feront en utilisant le simulateur intégré à l'outil de mise au point gdb.

3.1. Lancement du simulateur

arm-elf-gdb

gdb affiche quelques messages dont son numéro de version et sa configuration croisée et termine par son invite qui est :

(gdb)

3.2. définition du programme exécutable

file prog

où *prog* est le nom du fichier exécutable.

3.3. Connexion au simulateur

target sim

3.4. Chargement du programme exécutable

load

3.5. Exécution

run

3.6. Arrêt du simulateur

quit