Manuel d’utilisation

TABLE DES MATIÈRES

1. Présentation de l'application

LaTeX est un langage et un système de composition de documents. Il regroupe une collection de commandes destinées à faciliter l'utilisation du « processeur de texte ».Le programmeur a aussi la possibilité de créer ses propres commandes.

Le sujet du projet est d'écrire en d'introduire deux programmes qui illustrent les commandes **\ASIPRO** et **\SIPRO** en Latex.

Le programme **algo2asm** qui implémente la commande ASIPRO prend en argument un fichier Latex avec l'extension **.tex** contenant la description d'une fonction f dans la syntaxe du paquetage ALgo. Elle traduit la description de cet algorithme en ASIPRO, en produisant un fichier **.asm** correspondant au .tex.

Pour simplifier, il est dit dans le sujet qu'on supposera que le nom du fichier est le même que la fonction décrite dans le fichier (dans l'exemple précédent, on suppose donc que le fichier puissance.tex contient la description ALgo d'une fonction nommée puissance).

Le programme **run** qui implémente la commande SIPRO prend en argument une fonction avec ses arguments s'il y en a. Ensuite, il génère un code assembleur exécutable avec sipro qui calcule la valeur de la fonction.

2. Pré-requis

Afin d'utiliser les deux programmes convenables, plusieurs outils sont nécessaires à son fonctionnement.

- Système d'exploitation Linux : pour éxécuter l'application écrite en langage C.
- Version du langage C : c11 publiée en 2011 avec l'API POSIX 2008.
- Make : compiler et générer un fichier éxécutable grâce à un ensemble de règles écrites dans un fichier nommé Makefile.
- Flex : analyseur lexical
- Bison : analyseur grammaticale

3. Installation de l'émulateur assembleur SIPRO

SIPRO est le processeur qui permet d'éxécuter les programmes écrits dans un langage proche de l'assembleur afin d'éxécuter nos programmes LaTeX. SIPRO a été créée par M.Nicolas Bedon.

Lien du dépôt : https://github.com/NicolasBedon/asipro

SIPRO gère le codage des entiers, la manipulation des octets, de mots machines, et un bon nombre d'éléments dont nous avons besoin pour exécuter nos programmes.

Premièrement, il vous faut installer ASIPRO qui se charge de compiler le fichier **.asm** pour en donner un exécutable ainsi que SIPRO qui exécute le fichier exécutable obtenu par ASIPRO.

Pour ce faire, rendez-vous dans le dossier **asm** se trouvant à **/asipro/asm** et exécuter la commande **make**.

```
~/Documents/GitHub/Algo2AsmRun/asipro/asm$ make
gcc -DDEBUG -g -pg -std=c11 -pedantic -Wall -Wpointer-arith -pg analyse.c -c
gcc -DDEBUG -g -pg -std=c11 -pedantic -Wall -Wpointer-arith -pg asm.c -c
gcc -DDEBUG -g -pg -std=c11 -pedantic -Wall -Wpointer-arith -pg analyse.o asm.o -o asipro
~/Documents/GitHub/Algo2AsmRun/asipro/asm$
```

Ensuite, rendez-vous dans le dossier **emul** se trouvant à **/asipro/emul** et exécuter la commande **make**.

```
~/Documents/GitHub/Algo2AsmRun/asipro/emul$ make
gcc -posix -g -pg -std=c11 -pedantic -Wall -Wpointer-arith exec.c -c
gcc -posix -g -pg -std=c11 -pedantic -Wall -Wpointer-arith emul.c -c
gcc -posix -g -pg -std=c11 -pedantic -Wall -Wpointer-arith memory.c -c
gcc -posix -g -pg -std=c11 -pedantic -Wall -Wpointer-arith exec.o emul.o memory.o  -o sipro
~/Documents/GitHub/Algo2AsmRun/asipro/emul$
```

Maintenant que les fichiers exécutables pour exécuter asipro et sipro sont générées, il faut les ajouter à la variable d'environnement **PATH** pour éxécuter ces deux programmes à partir de n'importe quel répertoire.

Afin de réaliser cela, ouvrez le fichier **.bashrc** (CTRL+H dans le répertoire utilisateur, ex : /home/username).

Récupérer les chemins des dossiers asm et emul avec la commande pwd.

```
~/Documents/GitHub/Algo2AsmRun/asipro/asm$ pwd
/home/    /Documents/GitHub/Algo2AsmRun/asipro/asm
```

```
~/Documents/GitHub/Algo2AsmRun/asipro/emul$ pwd
/home/    /Documents/GitHub/Algo2AsmRun/asipro/emul
```

Dans le fichier .bashrc, ajouter à la variable PATH, les deux chemins récupérer en séparant chaque chemin par le caractère deux points **:** puis enregistrer et ouvrer nouveau terminal. Tapez aspiro pour tester si la commande est trouvée.

```
132  # Variable d'environnement PATH
133  export PATH="$PATH:/usr/local/bin/jdk-14.0.2/bin:/home/joo/Documents/GitHub/Compilation/asipro/asm:/home/joo/Documents/GitHub/Compilation/asipro/emul"
```

```
                ~$ asipro
ASIPRO V2.0
Usage: asipro input_file output_file
        Assembles input_file to output_file
                ~$ sipro
SIPRO V2.0
Usage: sipro [-t] [-d] [-h] file
        Runs <file> on SIPRO
        -t: trace mode
        -d: dump memory at end of execution
        -s: print stack in trace mode
        -h: print this message and exit
                ~$
```

Remarque :

Si vous ne souhaitez pas ajouter les commandes à la variable PATH, vous pouvez directement dans le terminal, tapez le chemin du répertoire pour exécuter le programme.

```
                ~/Documents$ ./GitHub/Algo2AsmRun/asipro/asm/asipro
ASIPRO V2.0
Usage: ./GitHub/Algo2AsmRun/asipro/asm/asipro input_file output_file
        Assembles input_file to output_file
                ~/Documents$ ./GitHub/Algo2AsmRun/asipro/emul/sipro
SIPRO V2.0
Usage: ./GitHub/Algo2AsmRun/asipro/emul/sipro [-t] [-d] [-h] file
        Runs <file> on SIPRO
        -t: trace mode
        -d: dump memory at end of execution
        -s: print stack in trace mode
        -h: print this message and exit
                ~/Documents$
```

4. Utilisation

Pour exécuter les deux programmes algo2asm et run, vous devez être dans le répertoire racine soit Algo2AsmRun.

1. make algo2asm : compilation de la commande \ASIPRO
2. ./algo2asm fichier.tex : génére la description assembleur du fichier.tex
   Le format minimale du fichier donnée en argument doit respecter la forme suivante :
   \BEGIN{algo}{nom_fonction}{argument1,argument2,...}
   \RETURN{resultat}
   \END{algo}
3. make run : compilation de commande \SIPRO
4. ./run '\SIPRO{nom_fonction}{argument1,argument2,...}' : assemble le code de description de la fonction avec le code principale pour ne faire qu'un code éxécutable.
5. asipro fonction_nom_main nom_executable
6. sipro nom_executable

Dans les fichiers sources, il y a un dossier **tests** permettant de tester un programme LaTeX.

Exemple avec le programme factorial.tex:

1. make algo2asm
2. ./algo2asm tests/recursion/factorial.tex
3. make run
4. ./run '\SIPRO{factorial}{5}'
5. asipro factorial_main executable
6. sipro executable