

Marion Fillaudeau et Ekaterina Galkina

Groupe 5

Documentation pour l'utilisateur

Sommaire

Cette documentation a pour but de guider l'utilisateur dans l'emploi de notre simulateur d'une machine virtuelle.

LANGAGE ASSEMBLEUR	3
Structure de la machine virtuelle	3
Syntaxe du langage assembleur	3
Commandes	3
COMPILATION EXÉCUTION DU PROGRAMME	4
Compilation	4
Exécution	5

LANGAGE ASSEMBLEUR

Structure de la machine virtuelle

Le machine simulée est composée d'une mémoire de 64 ko. Dans le processeur se trouvent plusieurs registres :

- un registre PC qui contient l'adresse de la prochaine instruction
- un registre d'état comprenant 3 bits : Z (mis à 1 si le résultat d'une opération est nul, sinon 0), C (mis à 1 si le résultat d'une opération génère une retenue, sinon 0) et N (recopie du bit de poids fort du résultat d'une opération).
- 32 registre généraux sur 16 bits. De r0 à r31, r0 est toujours nul.

Syntaxe du langage assembleur

Les programmes en langage assembleur doivent respecter des règles de syntaxe.

- Le programme contient une instruction par ligne. Chaque ligne est composée d'une étiquette optionnelle suivie de ':' puis de 0 ou de 1, d'espaces ou de tabulations précédant une commande écrite en minuscule. Les arguments de la commande doivent être séparés par des virgules et des espaces optionnels.
- Les valeurs immédiates peuvent être passées en les précédant de '#' si elles sont exprimées en décimal (peuvent être négatives) et de '#h' si elles sont exprimées en hexadécimal.

Commandes

- **add** rd, rn, S → met dans rd le résultat de l'addition de rn et S (registre ou valeur immédiate).
- **sub** rd, rn, S → met dans rd le résultat de la soustraction de rn et S (registre ou valeur immédiate).
- **mul** rd, rn, S → met dans rd le résultat de la multiplication de rn et S (registre ou valeur immédiate).
- **div** rd, rn, S → met dans rd le résultat de la division entière de rn et S (registre ou valeur immédiate).
- **and** rd, rn, S → met dans rd le résultat du ET logique de rn et S (registre ou valeur immédiate).
- **or** rd, rn, S → met dans rd le résultat du OU logique de rn et S (registre ou valeur immédiate).

- **xor** rd, rn, S → met dans rd le résultat du OU EXCLUSIF logique de rn et S (registre ou valeur immédiate).
- **shl** rd, rn, S → met dans rd le résultat de rn décalé S fois à gauche (registre ou valeur immédiate).
- **ldb** rd, (rn)S → met dans rd le contenu de l'adresse rn + S (registre ou valeur immédiate) sur 1 octet.
- **ldw** rd, (rn)S → met dans rd le contenu de l'adresse rn + S (registre ou valeur immédiate) sur 2 octets.
- **stb** (rd)S, rn → l'adresse rd+S (registre ou valeur immédiate) reçoit le contenu de rn sur 1 octet.
- **stw** (rd)S, rn → l'adresse rd+S (registre ou valeur immédiate) reçoit le contenu de rn sur 2 octets.
- **jmp** S → saut inconditionnel, à l'adresse S (registre ou valeur immédiate).
- **jzs** S → saut si le registre Z = 1, à l'adresse S (registre ou valeur immédiate).
- **jzc** S → saut si le registre Z = 0, à l'adresse S (registre ou valeur immédiate).
- **jcs** S → saut si le registre C = 1, à l'adresse S (registre ou valeur immédiate).
- **jcc** S → saut si le registre C = 0, à l'adresse S (registre ou valeur immédiate).
- **jns** S → saut si le registre N = 1, à l'adresse S (registre ou valeur immédiate).
- **jnc** S → saut si le registre N = 0, à l'adresse S (registre ou valeur immédiate).
- **in** rd → met dans rd une valeur entrée au clavier en décimal.
- **out** rd → affiche le contenu en décimal de rd
- **rnd** rd, rn, S → met dans rd un nombre aléatoire entre rn et S-1 (registre ou valeur immédiate).
- **hlt** → met fin à l'exécution du programme.

COMPILATION EXÉCUTION DU PROGRAMME

Compilation

Pour compiler le projet de machine virtuelle, il faut auparavant avoir installé gcc sur votre ordinateur.

Après avoir extrait le fichier *FillaudeauGalkina.zip*, déplacez vous dans ce dossier sur votre terminal.

```
> cd /chemin/FillaudeauGalkina
```

```
> gcc -Wall *.c -o simulateur -lm
```

Exécution

Si vous souhaitez simuler l'exécution d'un programme se trouvant dans un fichier nommé `nomFich.txt`.

`> ./simulateur nomFich.txt`

L'exécution provoquera la création d'un fichier `hexa.txt` qui contient la traduction de votre programme en "langage machine". Le résultat des instructions s'affichera dans votre terminal.