

Bonjour cher utilisateur, nous t'expliquons dans ce document le principe de notre programme et comment t'en servir.

## 1) Principe du programme

Le but de ce projet est de simuler un processeur fictif. Nous disposons d'une mémoire de 64 kilo-octets (64 fois 1024 soit 65 536 octets) et de 32 registres adressés de 0 à 31 qui contiennent 16 bits et le registre r0 est toujours mis à 0, il pourra donc servir si nécessaire. Une case mémoire est donc sur 8 bits. Notre programme aura pour but d'effectuer deux principales tâches. Premièrement l'assembleur va transformer un programme écrit en langage assembleur (dans un fichier texte) en un programme écrit en langage machine. Et ensuite on simule le programme qui exécute instruction par instruction. Notons qu'une instruction est codé sur 32 bits donc une instruction occupe donc 4 octets et donc 4 emplacements d'adresse. La machine comprend aussi 3 registres d'état Z, C et N qui sont mis à jour (à 1) si respectivement le résultat est nul, s'il y a une retenue et si le résultat est négatif. Notre simulateur effectue comme un processeur des instruction classique (arithmétique, de transfert, de saut, d'entrée-sortie).

## 2) Exécution

### Étape 1 : Téléchargement

Tout d'abord téléchargez notre programme sur votre PC. Vous trouverez un exécutable, son nom est « simulateur ».

### Étape 2 : Remplir un fichier

Afin d'exécuter le programme il faut au préalable remplir un fichier assembleur selon le cahier des charges. Donnez un nom à ce fichier.

Pour cela créer un fichier « txt » et remplissez-le en respectant la syntaxe suivante :

- Une instruction par ligne.
- Chaque ligne est composée d'une étiquette optionnelle, de 0, 1 ou plusieurs espaces ou tabulations, du code opération de l'instruction (en minuscule), suivi d'une ou plusieurs espaces, et des opérandes séparés par des virgules et éventuellement des espaces.

Voici un exemple :

```
ici: in r1
     jzs fin
     sub r1, r0, r1
     out r1
     jmp ici
fin: hlt
```

### Étape 3 : Exécution du programme

Exécuter le programme avec la commande suivante :

*./simulateur nom\_fichier\_assembleur.txt*

Le fichier hexadécimal sera créé dans votre répertoire.

FIN, BONNE SIMULATION !