

ARCHITECTURE DES ORDINATEURS

TP : 01pre

INSTALLATION DU SIMULATEUR Y86

L'architecture Y86 est un modèle de processeur inspiré de l'architecture dite « x86 » d'Intel. Elle en constitue une version simplifiée, à visée pédagogique. Elle a été conçue par les auteurs du cours *Computer Systems : A Programmer's Perspective* (<http://csapp.cs.cmu.edu/>), en tant qu'outil de travaux pratiques.

Cet outil a été largement adopté par la communauté mondiale des enseignants en architecture des ordinateurs. Cependant, comme ses technologies de mise en œuvre devenaient obsolètes (notamment, TCL/TK), des clones ont été écrits par des tiers, notamment en utilisant des technologies web. Un tel outil, plus simple d'utilisation, a été mis en ligne à l'université de Bordeaux.

Ce document explique comment installer la version TCL/TK d'origine du simulateur Y86.

Installation du simulateur en TCL/TK

Commencez par créer un répertoire `archi` et travaillez toujours dans ce répertoire. Notez que vous pouvez aussi utiliser les étapes suivantes pour installer le simulateur sur votre propre machine.

Sur les serveurs de l'université, TCL/TK 8.6 est déjà installé. Sur votre propre machine, il vous faudra sans doute installer TCL/TK en version « développement » (ces paquetages s'appelleront par exemple `tk8.5-dev` sous Ubuntu/Debian et `tk-devel` sous RedHat/Fedora/Mandriva), ainsi que les paquetages FLEX et BISON. MacOS propose également ces paquetages. Pour Windows, il faudra au moins installer MINGW et TCL/TK (version 8.5, par exemple la version « free » d'ACTIVETCL); il faut par contre corriger le `Makefile` en remplaçant `-ltcl8.5` et `-ltk8.5` par `-ltcl85` et `-ltk85`; il faut également mettre `C:\Tcl\bin` dans le `PATH`.

— Récupérez l'archive sur la page des supports de cours : <http://dept-info.labri.u-bordeaux.fr/ENSEIGNEMENT/archi/supports.html>

Comme vu ci-dessus, il en existe plusieurs versions. Sur les serveurs de l'université, prenez la version pour TCL/TK 8.6.

— Désarchivez le fichier : `tar xvf sim.tgz`

— Allez dans le répertoire `sim` : `cd sim`

— Lancez la compilation : `make`

Il ne devrait pas y avoir d'avertissements ou erreurs.

Si tout s'est bien passé, plusieurs simulateurs ont été créés.

Utilisation du simulateur `ssim`

Le simulateur de processeur Y86 le plus simple est un simulateur séquentiel, situé dans le répertoire `seq` :

— Allez dans le répertoire `seq` : `cd seq`

— Lancez le simulateur : `./ssim -g ../y86-code/prog6.yo &`

(l'option `-g` indique le mode graphique); il faut obligatoirement spécifier un fichier objet. Celui-ci (`prog6.yo`) est fourni à titre de test dans la distribution.

Si tout s'est bien passé, trois fenêtres apparaissent :

- La première contient le code du programme ;
- La deuxième fenêtre est le panneau de contrôle. Cliquez sur le bouton **Step** pour exécuter une instruction, et examinez l’affichage. La partie du milieu (les cadres bleus **Stage**) ne nous intéressent pas immédiatement : ils représentent le fonctionnement interne du processeur simulé, que nous étudierons plus tard. Concentrez-vous surtout sur les boutons du haut, le registre **PC** et la partie **Register File** ;
- La troisième fenêtre décrit le contenu de la mémoire. Elle ne sera pas utilisée pour l’instant, mais ne la détruisez pas pour autant, le simulateur n’aime pas !

Pour éditer le code source, ouvrez le fichier `../y86-code/prog6.y8` avec votre éditeur préféré. Par exemple avec EMACS, il faut faire `M-x asm-mode` pour avoir une coloration syntaxique du code assembleur.

Vous pouvez personnaliser EMACS ou VIM pour avoir une colorisation syntaxique automatique des fichiers `.y8`. Pour EMACS, ajoutez à `~/.emacs` (créez-le s’il n’existe pas) :

```
(add-to-list 'auto-mode-alist '("\\.y8$" . asm-mode))
```

et pour VIM, on ajoute à `~/.vimrc` :

```
au BufNewFile,BufRead *.y8 setf asm
```

Éditez maintenant le fichier objet `../y86-code/prog6.yo`, dans lequel vous trouverez la traduction des instructions assembleur en langage machine.

Utilisation de l’assembleur `y8`

L’assembleur `y8` se trouve dans le sous-répertoire `misc` de l’arborescence. Il est important de se placer dans ce répertoire afin que tous les fichiers nécessaires soient trouvés.

- Allez dans le répertoire `misc` : `cd misc`
 - Lancez la compilation individuellement sur votre fichier source `.y8` :
`./y8 mon-chemin/monfichier.y8 mon-autre-chemin/monfichier.yo`
- Il ne devrait pas y avoir d’avertissements ou erreurs.

Si tout s’est bien passé, le fichier objet `.yo` correspondant a été créé.