Y86 Tools

Assembler and Simulator

Documentation

simguide.pdf (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/y86/simguide.pdf) from CMU

What's Installed in 003

Code (modified to build and run cleanly on our Macs, and to fix some bugs): **sim-cs51-v1.taz (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/y86/sim-cs51-v1.taz)**

To install:

- Download it and put it somewhere
- Uncompress it:

  ```
  tar -xvzf sim-cs51-v1.taz
  ```

- cd down into sim and edit the beginning of the Makefile, as noted.
  - SIMPATH should be absolute pathname of where this directory lives
  - BINPATH should be the path to where you would like your executable binaries to go
- ```
  make
  ```

  to build.
- then

  ```
  make install
  ```

  to copy the binaries to your bin dir, if that's how you like to do things. (That's what I like to do, so that I can reach the tools from the command line no matter what dir I'm in). You might want to tell your shell that there are new things in bin; on csh/tcsh, you can do this with "rehash"; on bash, with "hash -r"

You get:

- yas, the assembler
  ```

- yis, an instruction-level simulator
- ssim, a simulator for the sequential Y86 datapath "SEQ" from the textbook
- psim, a simulator for the pipelined Y86 datapath "PIPE from the textbook
- plus lots of sample assembler code down in sim/y86-code/

If you want scrollbars in your program code window

Try replacing your seq.tcl with **this one (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/y86/with-scrollbars/seq.zip)** from **tianyu.cheng@utexas.edu (mailto:tianyu.cheng@utexas.edu)**

Ryan Amos' installation scripts for Linux and Windows

README.txt (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/y86/ryan/README.txt)

install.sh (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/y86/ryan/install.zip)

setup.sh (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/y86/ryan/setup.zip)

FYI: The Simulator I Use in Class

In class, I use a simulator I've been further customizing for various pedagogical purposes. So far:

- a "-d" option for ssim, to invoke it with a smaller window, so it fits on the projector screen
- a "-i" option for ssim, to invoke it with support for a memory-mapped keyboard and display

You can obtain the code here: **sim-cs51-v2.taz (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/y86/sim-cs51-v2.taz)** .
But don't forget to make, make install, and then rehash (or equivalent). Then

```
ssim -gi foo.yo
```