

Setting Up Your Work Environment

Log into one of the EWS machines. Open `terminal` (right click on the desktop and choose the application) and execute the `ece190` command. This will move you into the `/work2/ece190/ <net_id>` directory where you will have access to all the tools for the course.

Getting Around in UNIX

man <command>	prints the manual page for <command> (try <code>man man</code>).
ls [-al] [<dir_name>]	lists contents of <dir_name>.
cd [<directory>]	change the current working directory to <directory>.
rm [-rf] <filename>	remove the file named <filename>.
pwd	list the current working directory.
cp <src_file> <dest_file>	copy the file named <src_file> to a file named <dest_file>.
mv <src_file> <dest_file>	move the file named <src_file> to a file named <dest_file>.
mkdir <dir_name>	create a directory named <dir_name>.
rmdir <dir_name>	remove the directory named <dir_name>.
./<executable>	executes <executable>.
CTRL+c	terminate the execution of the currently running program.

I/O Piping

`./<executable> > <out_file>` causes the output of <executable> to be stored in a file named <out_file>.
`./<executable> < <in_file>` causes the <executable> to read its input from <in_file>.
You may combine these two. For example:
`./<executable> < <in_file> > <out_file>`
will cause <executable> to read input from <in_file> and write output to <out_file>.

Creating and Modifying Files

You can use `vim` or `emacs` to create and modify files. Without any command line arguments an new unnamed document will be created. Passing a file name on the command line will open that file for editing. `gvim` and `xemacs` are graphical versions of the two text editors. Here we cover some basic `vim` commands.

i	puts vim in <i>insert</i> mode.
ESC	puts vim in <i>command</i> mode.
:w	while in <i>command</i> mode writes the current file.
:q	while in <i>command</i> mode exits vim.
:wq	while in <i>command</i> mode writes the current file and exits.
:<line_num>	go to line <line_num>.
/<search_string>	while in <i>command</i> mode searches for <search_string> in the current file.

LC-3 Tools

Binary to Object Conversion

lc3convert <file>.bin
Converts an binary .bin file to an executable named <file>.obj.

Assembler

lc3as <file>.asm
Assembles the input file to an executable named <file>.obj.

Simulator

lc3sim <file>.obj
Simulates the execution of <file>.obj.

Inside the simulator you can use the following commands:

break (clear list set <label address >)	sets, clears, or lists breakpoints.
step/next	move to the next instruction.
continue/finish	run the code up until the next breakpoint (or until completions).
list <label address>	print the code or data around the label or address.
dump <label address>	display the values in memory near the argument location.
memory/register	set the value of a register or a memory location.
reset/quit	restart the simulation or quit the simulator.

You may also try using the graphical version of the simulator using the command:

lc3sim-tk

gcc

gcc <src_file>.c -o <out_file>
Compiles <src_file>.c to an executable named <out_file>.

Handins

handin --MP <mp_num> <file>

Where <mp_num> is the number of the MP that is due and <file> is the file you are handing in.