

# Software for CSE 341

Compilers or interpreters for all of the languages we'll be using this quarter should be installed on the undergrad computers on both Windows and Linux. You can also download and install them on your own machine if you have one. (Make sure you get the right version, which will generally be the latest one.) Specific directions for each language will be in the materials for that language.

## Using Linux and Emacs in CSE 341

### Reflection X

We'll use Reflection X to give you a graphical interface for your Unix session on [attu.cs.washington.edu](http://attu.cs.washington.edu)

### Logging in from the basement labs

If you're going use the labs, everything is set up for you:

- Linux machines: Click on Applications->System Tools->Terminal, and skip down to the Basic Unix commands section.
- Windows machines: Simply double-click the "SSH-X attu" icon on the desktop. Type in your CSE account password and you're all set. Now type "`xterm &`" without the quotation marks to get an xterm. Skip down to the Basic Unix commands section.

**OR**

Open the start menu and type "ssh attu" into the search bar. Two options should appear. Click on the option with the X icon. Type in your CSE account password and you're all set. Skip down to the Basic Unix commands section.

### Working elsewhere

It's probably simpler to work in the basement labs where we have checked that the right software is already installed and the directions we provide are correct. If you're willing to put in a bit of extra work to get things set up, you have a few options:

1. If you want to run something on a department linux machine, install Reflection-X on your computer, using the [CSE department instructions](#). What a Reflection-X session does is connect you to a CSE UNIX server. Everything you type or click is sent to the server, which sends back to your computer what to display. So you're using the department's software installation.
2. Install the necessary software (emacs and haskell (and possibly haskell-mode for emacs) at first, Racket later, Ruby after that) on your computer. The links on the course web page should help. Make sure you get the right versions.
3. For those savvy with terminals and ssh, run emacs and Haskell directly in terminals after connecting via ssh to the department.

### Basic Unix commands

To move around through the directory structure in your xterm window, you'll need to know a few basic Unix commands.

```
attu4% mkdir mydir
attu4% ls
mail mydir
attu4% cd mydir
attu4% pwd
/homes/iws/sbfan/mydir
attu4% exit
```

In the above (the things within the quotation marks are commands to type, don't type the quotation marks themselves):

- To make a directory, and name it, say "mydir", you would type "mkdir mydir".
- "ls" lists what you have in your current directory.
- "cd" will change directories. To change to a directory "mydir", for example, type "cd mydir". To move back out of a directory, type "cd ..", and to get back to your home directory, "cd ~".
- To check where you are, type "pwd".
- "exit" exits.
- To get help on some command, say you want to know how to use "ls", type "man ls" and you will get the manual pages for that command. (Alternatively, you can use "info" in the same way.)

These are just the very minimum basics, but you won't need too much more in 341. For more info on UNIX, see your ACM chapter's [tutorials](#), [this UNIX tutorial](#), material from the first week of [CSE303](#), or many other good resources.

## Accessing your UNIX home directories from Windows 7

If you're in the basement labs, your O: drive should already be mapped to your Unix home directory. Look under for the O: drive under "My Computer" in the Start Menu. To access your files, go to O:\unix\homes\iws\userid, replacing "userid" with your own CSE account username. All the files you saved while working on Attu should be there.

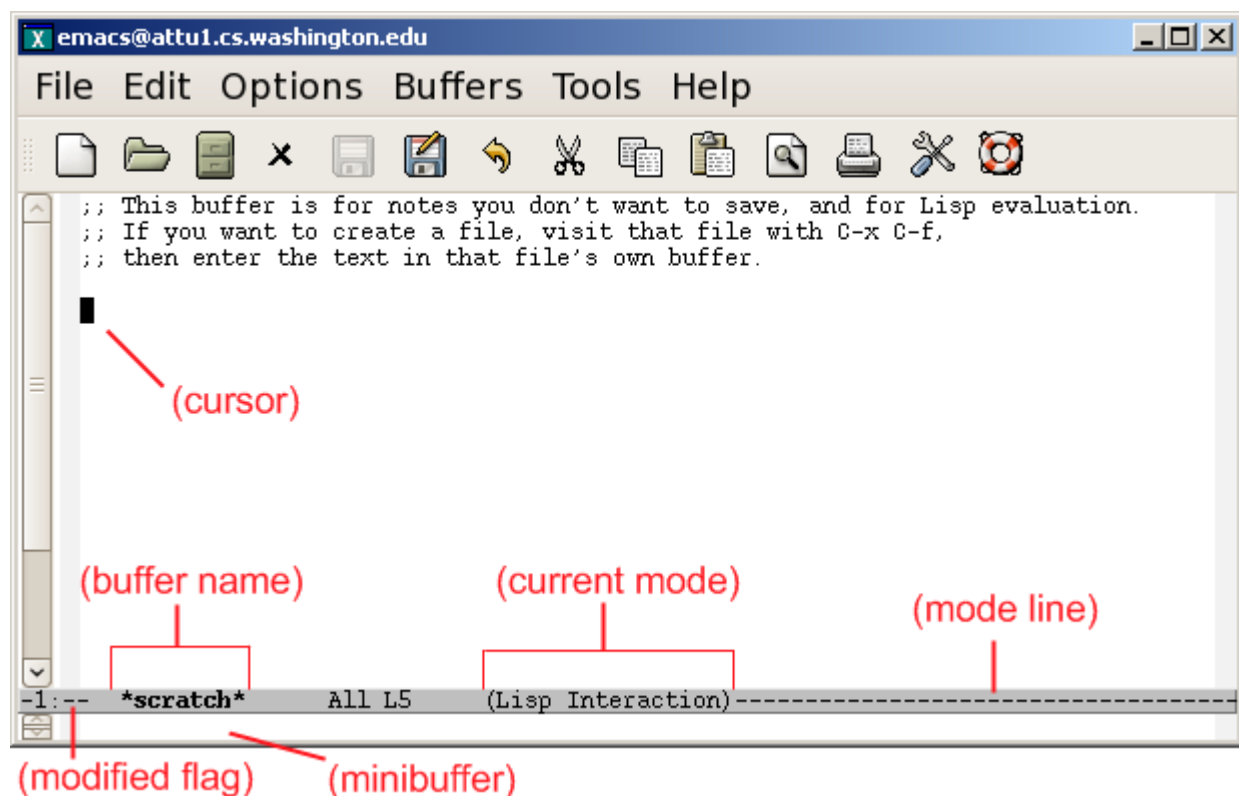
If you're in the labs and your O: drive isn't mapped, open up a Windows Explorer window (by clicking the "My Computer" from the Start Menu) and in the Tools menu item, select "Map Network Drive" and enter O: for drive and \\ntdfs\cs for folder. Click Finish. If you're at a computer that does not use CSE's name servers, for example in Mary Gates Hall or at home, you will need to use the fully qualified name, \\ntdfs.cs.washington.edu\cs.

For more info on this, see [the CSE support page that discusses Microsoft DFS](#), which is what we just used.

## Emacs

Now that you have your xterm open, let's open Emacs.

Type "emacs &" to open emacs in a new window. Note what happens if you don't type the "&"--you can't do anything in your xterm window. The "&" runs your program, in this case, emacs, in the background.



- The cursor is a rectangular block and is referred to as the **point**.
- The **modeline** displays information about the **buffer** displayed in the current **window**. A buffer is a logical "thing" that you are working on. When you open a file, it will be loaded into a buffer, typically with the same name as the file.
- Every buffer is edited in a **mode**. The most basic mode is **Fundamental**, which provides only the most basic Emacs editing features. There are modes for many different languages.
- There are many "special" buffers that do not correspond to loaded files. The one above is called **\*scratch\***. This buffer runs in **Lisp Interaction** mode, which means that you can interactively type and evaluate expressions in the Lisp programming language.

Emacs uses many key combinations involving the **Control** and **Meta** keys. Such key combinations are denoted **C-x** (Control-x (lowercase)) or **M-x** (Meta-x). On keyboards that don't have Meta, **Alt** is usually an acceptable substitute. If Alt doesn't work, ESC-x is equivalent to **M-x**.

A sequence of key presses is written like **C-a C-b M-x**, which would mean do the three actions in sequence.

The most important keys in Emacs

- **C-x C-c**: Quits Emacs.
- **C-g**: Cancels the current action.
- **C-x C-f**: Opens a file.
- **C-x C-s**: Saves a file.
- **C-x C-w**: "Save as" (save contents to file with a name you type).

Some other useful keys:

- **C-x 2**: Split the window into 2 buffers, one above the other. (Use the mouse or **C-x o** to switch between them.
- **C-x 0**: Undo window-splitting so there is only 1 buffer.
- **C-x b**: Switch to another buffer by entering its name.
- **C-x C-b**: See a list of all current buffers.

Getting help within Emacs: In addition to the help button on the right...

- **C-h**: Help. Hitting this will display a short message in the minibuffer: `C-h (Type ? for further options)`.
- **C-h t**: Built-in interactive tutorial. Some people don't like this tutorial, but some people find it helpful.
- **C-h b**: Key **b**indings. This lists all key bindings that are valid for the current mode. Note that key bindings change from mode to mode!
- **C-h a**: Command **a**propos. After hitting `C-h a` you can type a symbol and a window will appear that lists all symbols and functions that match that phrase.

---

Finally, there are other resources, including the (older but not necessarily outdated) [course help pages](#) as well as the [CS Lab](#) pages.

---

Acknowledgment: These notes have evolved over the last few years, but the linux/emacs notes are largely based on notes written by Keunwoo Lee in 2001. Last updated by Alan Borning for *Autumn 2012*.