

Using the Zoo Workstations

Version 1.86: January 16, 2014

If you've used Linux before, you can probably skip many of these instructions, but skim just in case. Please direct corrections and suggestions about this document to Dana Angluin.

1 Logging on to the Suse Linux machines for the FIRST time

1. The Zoo is a collection of workstations at the front of the third floor of AKW. The workstations are named after animals, e.g., frog and tick. There is a card-access area in room 300, and an open area at the south side of the building. Once you have an account, you may use the machines in person or remotely. In person is strongly recommended at first.
2. You must get a Zoo account for *each* Computer Science class that requires it. Course accounts should be used only for course-related purposes. From the Computer Science Department web page, follow the "Course Web Pages" link; at the top of that page is a link "Apply for a Zoo Account." Follow that link and go through the sign up procedure for an account for Computer Science 201. Choose the default shell unless you have a specific preference. Note that it may take up to an hour for your account to be created, so do this in advance (e.g., now.)

It is important for computer science students to learn to use the Zoo machines. The best way to do that is to spend time in the Zoo when others are there, and to ask lots of questions. At first this will feel awkward, but soon you'll be one of those answering questions, and you may find you have some new friends.

3. To log in once you have an account, find a machine that is not being used. It should display a message box inviting login by requesting your netid and password. (Click with the mouse if the screen is dark.) Type your netid and password. Shortly your name should appear in the upper right corner of the screen.
4. It may happen that someone else has previously locked the workstation, in which case the message box asks only for a password and displays someone else's name in the upper right corner of the screen with an icon of a lock. In this case, look for another workstation. It is Very Bad Form to leave workstations locked for extended periods of time!
5. To see menus, move the mouse to the upper left corner of the screen – Firefox and other applications should appear in a vertical bar on the left, and there will be the word "Activities" in the upper left corner. To open a terminal window, type the word "terminal" into the search box at the top. You'll have a selection of terminal applications; click on one to start a terminal window. The one described here is the Gnome terminal. It should display a prompt like: `dca3@tick:~>`. To change to your `cs201` directory, type `cd cs201`; the prompt should become something like `dca3@tick:~/cs201>`. You may type commands at this prompt, for example, `dr racket` to start DrRacket or `emacs` to start emacs. You may open multiple tabs in the terminal window to run different tasks.
6. To log out after a session, make sure to save all your work, and then click on your name in the upper right corner of the screen. Select the option to Log Out. (There is also an option to Lock Screen, which locks the workstation. Please lock the workstation for brief periods only.)

2 Manipulating files in the C-shell

You are expected to acquire a basic facility using Linux on the Zoo machines.

A few useful shell commands are described below.¹ Many of the commands deal with directories. Directories, like folders in the desktop interface, organize your files in a hierarchical way. Directories can contain

¹Also see <http://www.tldp.org/LDP/gs/node5.html> for a quick overview of Linux. Take a look at "3.2 Basic Linux concepts," "3.3 First steps into Linux," "3.8 Wildcards," and "3.10 File Permissions."

files and other directories. When referring to a directory within a directory within a directory, you separate the names of the directories with a “/”.

Upon login, you will be in your login directory; to change to your `cs201` directory, type `cd cs201` at the command line prompt. The “pathname” of your home directory is `/home/accts/your_name`. One of the subdirectories of your home directory is `cs201`; this is where you will store your homework files, as `hw1.rkt`, `hw2.rkt`, and so on.

A convenient abbreviation of the pathname of your home directory is `~`, so files `file1` and `file2` in your home directory could be referred to as `~/file1` and `~/file2`. The following lists some useful commands.

| | |
|-------------------------------|--|
| <code>pwd</code> | See what directory you are in. (Upon login, you will be in your login directory <code>/home/accts/your_name</code> .) |
| <code>ls</code> | List what is in the current directory. |
| <code>cd blah</code> | Change current directory to the one named “ <code>blah</code> ”. Note that “ <code>blah</code> ” may be an absolute path name (like <code>~/cs201</code>) or a path name relative to the current directory. |
| <code>cd</code> | Change current directory to your home directory. |
| <code>cd ..</code> | Change current directory to the one that contains this directory. |
| <code>mkdir foo</code> | Make a directory named “ <code>foo</code> ” in the current directory. |
| <code>rmdir bar</code> | Remove a directory named “ <code>bar</code> ” in the current directory (the directory must be empty before you can remove it). |
| <code>rm file1</code> | Remove the file called “ <code>file1</code> ”. |
| <code>cp file1 file2</code> | Make a copy of “ <code>file1</code> ” and call it “ <code>file2</code> ”. If “ <code>file2</code> ” is the name of a directory, a copy of <code>file1</code> called “ <code>file1</code> ” will be created under the “ <code>file2</code> ” directory. |
| <code>mv file1 file2</code> | Rename “ <code>file1</code> ” to “ <code>file2</code> ”. If “ <code>file2</code> ” is the name of a directory, “ <code>file1</code> ” will get moved to the “ <code>file2</code> ” directory. |
| <code>more file1</code> | Show the contents of “ <code>file1</code> ” one screen at a time. |
| <code>man some_command</code> | Display the online manual pages for “ <code>some_command</code> ”. |

Finally, you can print out a text file with the “`enscript`” or “`lpr`” command. Use this to print out your homework if a paper copy is required.

```
enscript file1
```

```
---OR ---
```

```
enscript -2r -Pzoo1 file1
```

The “`-2r`” will conserve paper (it fits two pages side by side on a single sheet), and “`-Pzoo1`” will print out on the main printer in the Zoo. There is a second printer called `zoo2`.

Since “`enscript`” sometimes has problems, if it doesn’t work, you can also print out your programs with the more primitive “`lpr`”:

```
lpr file1
```

or

```
pr file1 | lpr -Pzoo1
```

“`pr`” adds simple headers to `file1` and “`-Pzoo1`” will print out on the main printer in the Zoo.

3 Using Racket

The recommended development environment for Racket for this course is DrRacket. You may start DrRacket by typing `dracket` at the command prompt. This should open a window and instruct you to choose a language (if one has not already been chosen.) For this course the language choice is Racket. There should be a line `#lang racket` at the top of your file.

The window is generally divided into an upper part (for procedure definitions) and a lower part (for interacting with your program and evaluating expressions.) The Run button evaluates the expressions in the definitions area and clears the interaction area; use it after you have changed things in the definitions area.

As the course progresses, you'll be creating, running and debugging programs and saving them in files. You can do this by editing the procedures in the definitions area and saving the results to a file, or by using a text editor to create your program, and opening it in the definitions area. Please name your Racket homework files `hw1.rkt`, `hw2.rkt`, and so on, and save them in the `cs201` subdirectory of your home directory. The appendix of this document gives a quick introduction to emacs, a text editor available in Linux.

To become more familiar with the DrRacket interface, go to the Help menu, choose the Help Desk option, and do the Quick tutorial up through Section 8 (and beyond, if you are curious.) You can quit DrRacket by using the Quit command in the File Menu. You'll be returned to the shell prompt in the window where you started DrRacket.

4 Ready to depart the Zoo?

Make sure to save all your work in files. Click on your name in the upper right corner of the screen, and select "Log Out" on the drop-down menu. The display should return to the login screen.

5 Administrative problems

If you have problems with one of the computers, or if you have problems with your account that are not CS 201 specific problems, first ask a friend for help. If that fails, send mail to "`cs.support@cs.yale.edu`". E-mail sent to that address is handled by the staff members who are in charge of systems administration for the Zoo.

6 Final comments

This handout should tell you much of what you need to know to get you started in this course. Please go to the Zoo and ask lots of questions to fill in any gaps. There is a lot more to Linux and to Emacs, however, so you may want to play around with them a bit, go through the tutorials, and ask knowledgeable friends for help before you get your first real assignment.

The TA(s) will be happy to answer any questions you may have about anything related to the course (if the instructions here don't make sense or don't work, if lecture material is unclear, if you get stuck on a problem, etc.), but they might be less willing to deal with questions like "How do I change the color of my Emacs window?" and "Where is the Tetris game in Emacs?"

A Appendix: Using Emacs

A text editor allows you to write and edit text files, including the Scheme programs you will be writing for this course. It is recommended that computer science students become familiar with a Linux text editor: emacs or vi. This section gives a brief introduction to emacs. To start emacs, type `emacs` at the shell prompt; if you are sitting in the Zoo, you will get a new emacs window; what happens with ssh will depend on how it is configured. You can learn about emacs by going through the tutorial (Select "Emacs Tutorial" under the "Help" menu in the Emacs window); the whole tutorial may take about an hour. Of course you can exit before finishing the tutorial.

Emacs uses the Ctrl and Alt (or Esc) keys a lot. Locate them before you get going. It also offers a lot of functionality in point and click style, if you prefer that. The concept of a “buffer” is that it holds a copy of your file while you edit it – the changes are not made permanent until you explicitly save. You can be viewing several files simultaneously in several buffers; copy text between them, etc. Emacs can recognize Scheme syntax and can help you balance parentheses and indent code in a reasonable style (use tab.)

Here are some useful commands.

- **CTRL-x CTRL-c** (type them in succession) will let you exit Emacs, asking if you want to save any files that haven’t been saved.
- **CTRL-x CTRL-s** will save the file in the currently active buffer (do this often when you start programming, so that you don’t lose anything).
- **CTRL-x CTRL-f** will prompt you at the bottom of the Emacs screen to name a file. You can either type the name of a file that exists, in which case Emacs will load it into the current buffer, or you can type the name of a new file, in which case Emacs will create a new file by that name. You can keep loading more files into a single window—they will be hidden, but they are still there. To find other buffers, try:
 - You can use the pull down menu “Buffers” at the top left, *or*
 - **CTRL-x b** to switch buffers. This will prompt you to enter the name of a file that has already been loaded into Emacs.
- Use the arrow keys on the keyboard to move up, down, left, and right. You can also use **CTRL-p** to move up, **CTRL-n** to move down, **CTRL-b** to move left, and **CTRL-f** to move forward.
- The “Page Down” button or **CTRL-v** moves you a page forward.
- The “Page Up” button or **ALT-v** moves you a page back. (**ALT-x** is also written **M-x** in Emacs documentation).
- **CTRL-l** centers the page so that whatever line the cursor is on will be in the center of the window.
- The “End” button moves you to the end of a line, and **CTRL-End** moves you to the end of a file.
- The “Home” button moves you to the front of a line, and **CTRL-Home** moves you to the front of a file.
- **ALT->** moves you to the end of the file.
- **ALT-<** moves you to the beginning of the file.
- **CTRL-d** deletes the character under the cursor.
- “Delete” deletes the character before the cursor.
- “Insert” switches Emacs between “overwrite mode” and “insert mode.”
- **CTRL-k** kills everything at and after the cursor on that line (and sticks it in the clipboard).
- **CTRL-@** sets a “mark.” Once you set a mark, you can define a “region” by moving the cursor to the other end of the region.
- **CTRL-w** cuts a region (and puts it into the clipboard).
- **ALT-w** copies a region into the clipboard.
- **CTRL-y** yanks whatever is in the clipboard to wherever the cursor is (like pasting).
- You can also copy and paste regions by selecting them with the mouse. Press and drag with the left mouse button to select a region. Click on the middle button to paste it.
- **CTRL-_** (**CTRL-SHIFT-hyphen**) lets you undo the last operation. Pressing it twice in succession causes Emacs to undo the last two operations, and so on.

- `CTRL-x 2` (type “CTRL-x”, then “2” by itself) will split the Emacs window into two. You can have two different files displayed at once using this feature. To go back to one window, same command with 1 instead of 2.
- `CTRL-x o` will move the cursor into the other window, in the case that you have two windows.
- `CTRL-x 1` will hide the other window, in the case that you have two windows.
- `ALT-x` will prompt you to enter some command which is not necessarily bound to a keystroke, but which is still a command in Emacs.
- `CTRL-g` will stop any operation—if you get stuck in Emacs, it often helps to hit `CTRL-g` a few times.

If Emacs beeps at you, see what it says at the bottom of the window. Often, it will be prompting you for a “yes” or “no”.