

## Linux, X, and Emacs<sup>1</sup>

### 1 Welcome to the Zoo

This semester, we will be using the CS department's Linux workstations for all programming in CS251. You are not expected to have any prior experience with Linux coming into this course. As part of your first labs and assignments, you will become familiar with Linux and related tools (e.g., shells, X Windows, Emacs, and the Sun Java system). This handout provides some information to help you get started, and pointers to places where you can find more information.

There are currently 23 public CS department Linux machines distributed in four areas: (1) the Linux Lab (the open computer area outside of rooms E131, E133, and E135, also known as the **micro-focus**); (2) the Hardware Lab (E125); (3) SCI 173; and (4) SCI E111. Most of the machines are named after animals:

1. **The micro-focus**: cardinal, finch, irwin, jay, lark, minnow, moose, robin, shark, swallow, thrush, warbler, wren
2. **E125** (Hardware lab): baboon, chimp, gibbon, gorilla, lemur, orangutan, tamarin
3. **173** (CS Lounge): bass, trout
4. **E111** (CS Classroom): eel

You may use any of these workstations for your CS251 work. Note that the machines in E125 are dual boot (can be booted into both Windows XP and Linux); you may need to reboot these into Linux in order to use them.

The E125 machines do not boot into Linux by default. To make the machine boot into Linux, reboot it, when the machine powers up you will see a blue GRUB screen that says Fedora Core on the bottom. Press the **Enter** key and then use the arrow keys to highlight the top most Fedora Core line. Then hit the **Enter** key to boot up into Linux.

If you want to stay current on the status of the Linux workstations, you may want to follow the **Vibrant Linux** conference on FirstClass, inside the **Computer Science** conference folder.

If you have taken CS230 recently or have taken one of the CS department's 300-level courses, you may already be familiar with working in the Linux environment and with using tools such as shells, X Windows, and Emacs. If not, you need some pointers to information on how to log in to these machines and use these tools. That's the purpose of this handout.

### 2 Logging In to a Linux Console

In order to use a Linux workstation, you must have an account on Puma, the CS department fileserver. You should already have a Puma account with the same username and password as you used in CS111/CS230. If you do not, or you have forgotten your password, please contact Lyn ASAP.

The easiest way to do your work in CS251 is to log in directly to one of the Linux consoles in the micro-focus or E125. A console that is not in use displays a Linux login screen that has a small window in which to enter your username. To log in, type your **puma** username (the same as your

---

<sup>1</sup>This handout was written by Lyn Turbak and revised by Rebecca Shapiro

“short” FirstClass username) and press the ENTER key. A second window will appear where you should type your password, followed by ENTER.

If the log in is successful, a Fedora<sup>2</sup> desktop screen will appear with several icons (one of which should be labeled with the name of your home directory). There should also be a menu bar along one side of the screen. At one end of the menu bar is a red hat icon that opens a menu, similar to the Start button on Windows machines.

A few things can go wrong when you are logging in:

- If you misspelled or used the wrong username or password, you will be prompted for your username and password again. If you cannot log into the console of a Linux workstation after repeated attempts, send an email message to *all* of Lyn, Stella, and the system administrators at CS-Sysadmin@wellesley.edu.
- If the screen is blank, type any key and the login screen should appear.
- If the screen displays a screensaver of some form, it has been “locked” by another student (see notes on locking etiquette, below) and you cannot use it. Try to log in to another machine.
- If the screen display looks like a window manager, then another student is logged into the machine and may still be using it.
- If you are in E125 and the screen display looks like Windows XP, then the machine has been booted in the wrong mode and needs to be booted into Linux mode.

### 3 Accessing Puma Files

The Puma file system is automatically mounted onto the Linux machine when you log in. This means that you can transparently refer to any of your files on Puma from the Linux machine using the normal Puma file names. For instance, if you are user `gdome`, and you create a file named `~gdome/test.txt`, this file is actually stored on Puma and not on the Linux machine in the directory `/home/gdome/test.txt`. You can later view, edit, or delete any such file from any Linux machine or from Puma itself.

### 4 The Linux Shell

You can accomplish many tasks in Fedora’s windows manager by pointing and clicking, just as on a Mac or in Windows. However, some tasks can only be accomplished (or can be accomplished more conveniently) by typing textual commands in an interactive program called a *shell*. For instance, in a shell, you can easily copy all files whose name match a certain pattern from one directory to another – something that can be more tedious using point-and-click interfaces.

One way to open a shell in Linux is to select the **Applications>Accessories>Terminal** the menu at the top left-hand side of the screen. A window will appear containing a shell prompt that looks something like

```
[username@hostname current-directory]
```

where *username* is your username and *hostname* is the name of the machine into which you logged in. The *current-directory* is the directory that you are currently connected to, and is initially set to your username when you first log in.

---

<sup>2</sup>Fedora is the name of the Linux operating system that we use. It is a free operating system distributed by Red Hat.

At the shell prompt, you type a Linux command to execute, followed by the ENTER key. Linux then executes this command, and upon finishing the execution, presents you with another shell prompt. This mode of interaction may be unfamiliar if you have only had experience with a point-and-click, drag-and-drop window system.

There are a plethora of shell commands for tasks such as navigating through and modifying the file system, searching for files that match certain criteria, finding documentation, and invoking programs like text editors and compilers. For a quick introduction to some very basic commands, see Section 4 (Shell Commands) of Scott Anderson's article *Introduction to Unix and the X Window System*, which can be found on-line at:

<http://cs.wellesley.edu/~anderson/writing/unix/unix-intro.pdf>

For a more detailed introduction, read Chapter 4 (The Unix Shell) of Larry Greenfield's *The LINUX Users' Guide*. There are a few red-bound copies of this guide next to the Linux workstations in the Linux Lab. The guide is also available on-line at the following URL:

PDF version: <http://cs.wellesley.edu/user-info/handouts/linuxUsersGuide.pdf>

Once you have mastered the simple shell commands and are comfortable with X Windows and Emacs as described below, you are encouraged to learn more powerful shell commands. A good starting point is *The LINUX User's Guide*, particularly the following chapters: Chapter 6 (*Working with Unix*); Chapter 7 (*Powerful Little Programs*); Chapter 9 (*I Gotta be Me!*); and Chapter 11 (*Funny Commands*). Also, the Linux `man` command can be used to find detailed documentation on any command. For example, executing `man ls` gives documentation on the file-listing command `ls`.

In addition to using the **Applications>Accessories>Terminal** menu option, there are two other standard ways to create a new shell window:

1. Execute `xterm &` from an existing shell window.
2. Create a shell within Emacs via `M-x shell` (see Sec. 6 on Emacs, below).

## 5 X Windows

When you first log in to a Linux machine, the X windows system is launched automatically. If for some reason, you are not running X on your machine, you can launch X via the shell command `startx`. This will change the display from a strictly text-based interface to a graphical windows interface similar to that on Macs and Windows machines.

The particular window manager we are using this semester is called Gnome. Using Gnome is fairly intuitive. As described above, at the top left corner of the screen is a set of menus together whose purpose similar to the Start button in Windows. Feel free to explore these menus to see what is available. Some particularly important options are **Applications>Programming>Emacs** (to launch Emacs) and **Applications>Internet>Firefox Web Browser** (to launch Firefox, a web browser).

## 6 Emacs

Emacs is an extensible, customizable, self-documenting text editor created by Richard Stallman. Many consider it to be one of the greatest programs of all time. It is one of the flagship programs of Stallman's Free Software Foundation and GNU project.

You will be doing most of your work this semester – writing, executing, and debugging programs in OCAML and other languages – using Emacs. In fact, it is possible to do all your work in the

course entirely within Emacs. It is very important to become a proficient Emacs user because this will save you a lot of time during the semester.

There are two standard ways to launch Emacs:

- Select the **Applications>Programming>Emacs** option from the Fedora menu.
- Execute `emacs &` from within a shell. The `&` will create a separate Emacs window. If you are working remotely and do not wish a window to be created, instead execute `emacs -nw` (the `-nw` means “no window”).

All Emacs documentation, including a tutorial and reference information, is available online. If you are unfamiliar with Emacs (or have used it before but are rusty), you should take the online Emacs tutorial. You can do this by typing the **Control** and **h** keys at the same time, followed by the **t** key.<sup>3</sup> This will load an interactive tutorial, whose directions you should follow. When you complete the tutorial, you will know how to do basic editing in Emacs.

The tutorial teaches you keystroke commands for basic Emacs functionality. If you prefer, most of this functionality can instead be accessed by using a combination of the mouse, menu items, and arrow keys. However, we strongly recommend that you learn the keystroke commands, as they will save you lots of time and make it easier for you to work remotely from PCs or Macs (see Sec. 10).

In addition to taking the tutorial, you should read Scott Anderson’s article *Introduction to The Emacs Editor*, which can be found online at:

<http://cs.wellesley.edu/~anderson/writing/emacs/emacs-intro2.pdf>

Another useful introduction to Emacs is Chapter 8 (*Editing Files with Emacs*) of Larry Greenfield’s *The LINUX User’s Guide*. You will find links to this and several sites containing more detailed Emacs documentation on the CS251 home page.

A particularly useful link is the Emacs reference card you can find at

<http://cs.wellesley.edu/~cs251/emacs-refcard-letter.pdf>

This is so handy that I have provided it to you as Handout #6. You should have this card with you for handy reference every time you use Emacs.

It turns out that Emacs even has its own hypertext information system. This system contains detailed documentation on Emacs itself, and is worth exploring to find out more about Emacs. In order to access this information system, type the **ESCAPE** key, followed by the **x** key, followed by the character sequence `info`.<sup>4</sup> This will load up an editor buffer that contains a top-level menu of the system documentation. You can browse this system via mouse clicks, much as you browse web pages in a web browser.

The Emacs command `M-x shell` creates a shell that runs inside an Emacs buffer. It is convenient to have a shell within Emacs, because then any shell command can be easily executed without leaving Emacs. This can be important if you are accessing a Linux machine remotely from PCs or Macs (see Sec. 10).

One minor drawback of running a shell under Emacs is that Emacs sometimes interprets or prints character sequences in a different way than a separate shell window would. For example, an Emacs shell will echo passwords that a normal shell would not. Also, the `ls` command in an Emacs shell may print a lot of annoying formatting characters; these can be removed by first executing `unalias ls` in the Emacs shell.

## 7 Printing

There are three standard ways to print your files from a Linux cluster machine:

---

<sup>3</sup>In Emacs notation, this keystroke combination is usually written `C-h t` and pronounced “control-h t”.

<sup>4</sup>This keystroke combination, pronounced “meta-x info”, is usually notated as `M-x info`.

- Within Emacs, select either the `File:Print Buffer` or `File:Postscript Print Buffer` menu options. (The latter gives nicer looking output.)
- Within a shell, execute `lpr filename`.
- Within a shell, execute `a2ps -1 filename`.

All of these options will print your document on printer `psci11`, which is one of the printers near the mini-focus consultant's desk. If you use `lpr` or `a2ps`, you can print to a different printer using the `-P` option. For instance, you can print to `psci1r` using `lpr -Ppsci1r`. Note there is no space between the `-P` and the `psci1r`.

If you experience printing problems, please report them to our Linux system administrators by posting to the `CS-SysAdmin FirstClass` conference.

## 8 Saving Work

In addition to saving work in your `puma` home directory, you should make backup copies of your work on a thumb drive, on `FirstClass`, or on your own personal computer. To use a thumb drive, simply insert your thumb drive into a usb port in the machine. The drive should then be automatically mounted and you should see a file browser window pop open showing the contents of your drive. If you would like to access your thumb drive through the command line, the contents of it are likely to be found at `/media/disk/`. When you are done, be sure to eject your thumbdrive. To do this find the icon on your desktop that links to your thumb drive, right click on it and click `Unmount Volume`.

## 9 Setting Up and Customizing Your Account

If you desire to use `FirstClass`, you must set it up for use in your account. To do this first open up the client, either by running the command:

```
/opt/firstclass/fcc &
```

in a shell or by opening it through the menu: `Applications>Internet>FirstClass Client`. Once `FirstClass` opens, click `Setup...` and under `Server` type `firstclass.wellesley.edu`, then you can click `save` and login to `FirstClass`. Please note that `FirstClass` on Linux is not very stable, so it may crash on you more often than you would like.

You can play around with your background to customize your account a little. The customizations you make will show up on any Computer Science Department client machine you use. To change your background go to `Applications>System>Preferences>Desktop Background`. Another way to customize your account is to change your Terminal colors. Open your Terminal, then go to `Edit>Current Profile>Colors` you can then play with the palette to customize your terminal. Don't be afraid to browse through the Preferences menu to see how else you can customize your account!

## 10 Using Linux Machines Remotely

You do not have to be physically seated in front of one of the Linux workstations in order to use it. You can access the department's Linux machines remotely from any PC, Mac, or Linux machine on the Internet:

- From PCs running Windows XP, you can connect to Linux machines using the PuTTY program, which can be found inside the `Program Files>PUTTY` folder on the local disk. After

launching the program, enter the name of one of the Linux workstations (listed at the beginning of this handout) in the **Host Name** field and be sure to check the **SSH** radio box.

- From Macs running OS 10.4, first launch the X11<sup>5</sup> program in the **Applications/Utilities/** folder. In the terminal window, you can then invoke the **ssh** program. To connect to the machine named **moose**, for example, enter the following command:

```
ssh -Y username@moose.wellesley.edu
```

(Use **ssh -X** if you are using 10.3)

Note that you may receive a warning indicating that the authenticity of the host cannot be established, and a question about whether you want to continue connecting. If you respond with **yes**, the **ssh** program should then connect you to the Linux machine.

- You can also connect from one Linux machine to another, **puma**, for example, via the following **ssh** command

```
ssh -X username@puma.wellesley.edu
```

There are two key advantages of connecting to a Linux machine remotely. First, you can access the **puma** from any other machine on the Internet, and any of the client machines (**finch**, **bass**, etc) from any other machine in Wellesley – a fact which is important when you don't wish to go to the Science Center. Second, you can still use the machines even when all consoles are actively being used (several people can be logged into the same Linux machine at once). This is important to know when the lab areas are crowded with people near a problem set deadline.

On many systems, it is even possible to display remote X Windows on your local machine. If you use **ssh -X** to connect from one Linux machine to another, any windows created on the remote machine will automatically be displayed on the local Linux machine! On Macs with X11, any remote X Windows created under **ssh** will display on your local machine as long as you have used the **-X** or **-Y** flags with **ssh**. On PCs, the free **Xming** server can be used to get a similar effect. Using these kinds of connections, you can remotely do almost *anything* on a micro-focus Linux machine that you could do if you were seated directly in front of it.

If you have a text-only remote connection, it helps to be familiar with Emacs control-and meta-key commands!

You may want to install Linux on your own computer. If you would like to borrow a Fedora installation DVD or CD, please send an email to **cs-sysadmin**.

## 11 Logging Out of a Linux Machine

After you are done using a Linux workstation, you need to logout. From the Fedora window manager, logging out is a two-step process:

1. Select the menu sequence **System>Logout**
2. Select **OK** in the resulting pop-up window.

You know that you have succeeded in logging out when you see the Linux login prompt appear.

If you are logged in remotely, you can log out by executing **logout** in the shell created by your remote access program.

---

<sup>5</sup>Note: If the Mac does not have X11 installed, you will need to SSH using **Terminal**. In **Terminal** you do not have a graphical user interface, so you must **ssh username@moose.wellesley.edu**

It is important not to accidentally leave yourself logged in to a Linux machine when you are done. If you do so, someone may accidentally or purposely read, modify, or delete your files. Also, you will be tying up an important resource.

If you want to leave a Linux console for a *short* break, you can “lock” your console by selecting **Lock Screen** from the System menu. This will lock the screen in such a way that your password is required to unlock it. You should only lock machines for *short* breaks (as a rule of thumb, no more than 15 minutes). Otherwise, you will be tying up an important resource that someone else may need to use.