CS251 Programming Languages **Handout # 3**
Prof. Lyn Turbak **January 31, 2005**
Wellesley College

Linux, X, and Emacs

1 Welcome to the Zoo

This semester, we will be using the CS department's Linux workstations for all programming in CS251. There are currently 23 such 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 birds, apes, cats, and bears:

1. **The micro-focus**: cardinal, druantia, finch, grizzly, jay, lark, panda, robin, swallow, teddy, thrush, warbler, wren,
2. **E125** (Hardware lab): baboon, chimp, gibbon, gorilla, lemur, orangutan, tamarin
3. **173** (CS Lounge): gazelle, impala
4. **E111** (CS Classroom): fox

You may use any of these workstations for your CS251 work. Note that most of the machines in the micro-focus and 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.

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 use one of the Linux workstations is to directly log in to one of the workstation consoles. A console that is not in use displays a Linux login screen that has a redhat icon in the upper left corner and a small window to enter your username. To log in, type your `puma` username (the same as your "short" First-Class 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 new screen will appear with three icons in the upper left corner (one will be labeled with the name of your home directory), and several icons along a menu bar at the bottom of the screen. At the far left end of the menu bar is a red hat icon that opens a menu, similar to the Start button on Windows machines. In this document, this menu will be referred to as the "red hat menu".

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 Lyn.
- 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 in Sec. 11) 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 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 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.
- From Macs running OS X, first launch the `Terminal` program in the `Applications` folder. In the terminal window, you can then invoke the `ssh` program. To connect to the machine named `robin`, for example, enter the following command:

  `ssh` *username*`@robin.wellesley.edu`

  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 via the `ssh` command described above.

There are two key advantages of connecting to a Linux machine remotely. First, you can access the Linux machines from any other machine on the Internet – 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.

A disadvantage to accessing the Linux machines remotely is that the PC and Mac clients provide only a text-based interface, so you will not be able to use the graphical user interface familiar from the console.[1] For this reason, it helps to be familiar with Emacs control- and meta- key commands!

[1]Note: there are programs (such as Hummingbird's Exceed software) for both Macs and PCs that can display X windows from a remote Linux machine, but these are not standard on the public cluster machines at Wellesley.

However, if you use `ssh` to connect from one Linux machine to another, any windows created on the remote machine will automatically be displayed on the local Linux machine! For this reason, it is *extremely* handy to install a Linux partition on your personal computer. For instance, using a Linux partition on my laptop, I can remotely do almost *anything* on a micro-focus Linux machine that I could do if I were seated directly in front of it.

4 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.

5 The Linux Shell

You can accomplish many tasks in Red Hat 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 `System Tools>Terminal` option from the red hat menu. 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.

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. Although many tasks are not as convenient in a text-based shell as in a graphical interface, some tasks, such as finding all files that satisfy a non-trivial specification or automating a sequence of tasks, are actually more convenient in the text-based interface.

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/user-info/handouts/unix-intro.pdf`

and is linked from the CS251 home page. 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 URLs, which are linked from the CS251 home page:

HTML version: http://espc22.murdoch.edu.au/~stewart/guide/guide.html
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 `System Tools>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. 7 on Emacs, below).

6 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 bottom left corner of the screen is a red hat icon that serves a purpose similar to the Start button in Windows. Click on this icon for a menu of options. Some particularly important options are `Programming>Emacs` (to launch Emacs) and `Internet>Web Browser` (to launch Mozilla, a web browser).

7 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 `Programming>Emacs` option from the red hat 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, are on-line. If you are unfamiliar with Emacs (or have used it before but are rusty), you should takes the on-line Emacs tutorial. You can do this by typing the `Control` and `h` keys at the same time, followed by the `t`

key.[2] This will load a 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, I 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. 3).

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

`http://cs.wellesley.edu/user-info/handouts/emacs-intro.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`.[3] 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. 3).

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.

8 Printing

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

- 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 `psci1l`, 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

[2] In Emacs notation, this keystroke combination is usually written `C-h t` and pronounced "control-h t".

[3] This keystroke combintation, pronounced "meta-x info", is usually notated as `M-x info`.

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.

9 Saving Work

In addition to saving work in your `puma` home directory, you should make backup copies of your work on a Zip/floppy disk, on FirstClass, or on your own personal computer. For instructions on how to use a Zip disk with the Linux machines, read Section 8 (*Using Removable Disks*) of Scott Anderson's *Introduction to Unix and the X Window System*.

10 Logging Out of a Linux Machine

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

1. Select the menu sequence `Red Hat>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.

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.

11 Locking a Linux Machine

If you want to leave a Linux console for a *short* break, you can "lock" your console by selecting `Lock Screen` from the red hat 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.