CS301 Compiler Design and Implementation **Handout # 2**
Prof. Lyn Turbak **September 3, 2003**
Wellesley College

Linux, X, and Emacs

1 Lions and Tigers and Bears – Oh My!

This semester, we will be using the CS department's Linux workstations for all programming in CS301. There are currently 24 such machines distributed in three areas:

1. **The micro-focus**: druantia, grizzly, leopard, lion, lynx, ocelot, panda, polar, rhodes, sloth, teddy, tiger
2. **121B** (Network/security lab, behind the mini-focus consultant's desk): caribou, dudley, ibex, kudu, santorini
3. **E125** (Hardware lab): baboon, chimp, gibbon, gorilla, lemur, orangutan, tamarin

You may use any of these workstations for your CS301 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.

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 not in use should display a Red Hat login screen, which prompts you first for a username, then a password. Enter your Puma username and password at the prompt. If things go right, you should be presented with the Red Hat window manager.

A few things can go wrong when you are logging in. 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. 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 Connecting to a Linux Machine 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 any of the department's Linux machines remotely via a `telnet` client, such as BetterTelnet on a Mac or QVTerminal on a PC, or an `ssh` client, such as NiftyTelnet on a Mac or `ssh` on another Linux box or a Mac running OS X. You can use one of these clients to connect to one of the machines with your Puma username and password. (A list of all machine names appears at the top of this handout.) Connecting via ssh is preferable to telnet because it is more secure.

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, via ssh or telnet 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 of using telnet is that a telnet clients provide only a text-based interface, so you will not be using the graphical user interface familiar from the console. However, since nothing in the course depends on graphics (Emacs/Scheme/ML/Haskell all work wonderfully well in a purely text-based environment), you can work on any assignment in the course via telnet. (It helps to be familiar with Emacs control- and meta- key commands!)

Ssh clients are better in this regard. Many such clients perform X forwarding and allow you to display a remote X window on your local machine (if your local machine has X).

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.

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 Using a 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.

If you are unfamiliar with shells, you should read Chapter 4 (*The Unix Shell*) of Larry Greenfield's *The LINUX User's Guide*. This guide is available on-line at

`http://cs.wellesley.edu/~cs301/linuxUsersGuide.pdf`

which is linked from the *Resource Links* section of the CS301 home page. There are often a few hardcopies of the guide sprinkled amongst the Linux machines. You should also read Scott Anderson's *Introduction to Unix and the X Window System*, available at :

`http://cs.wellesley.edu/user-info/handouts/unix-intro.pdf`

There are three standard ways to create a new shell window:

- Right-click on the background and select **New Terminal**.
- Execute `xterm &` from an existing shell window.
- Create a shell within Emacs via `M-x shell` (see the section on Emacs, below). It is very convenient to have a shell within Emacs, because then any shell command can be easily executed without leaving Emacs. This is especially important if you are telneting into a Linux machine.

  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 instance, 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.

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 also happens to be free; in fact, it is 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 – 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 menu sequence `Red Hat:Programming:Emacs` (where `Red Hat` stands for the Red Hat icon in the bottom left corner of the screen).
- 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.[1] 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 via telnet.

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 CS301 web page. You should also read Scott Anderson's *Introduction to the Emacs editor*, available at:

`http://cs.wellesley.edu/user-info/handouts/emacs-intro.pdf`

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

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

`http://cs.wellesley.edu/~cs301/emacs-refcard-letter.pdf`

You may want to print out a copy of this card and carry it with you for handy reference.

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`.[2] 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.

7 Printing

There are two 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 sequences. (The latter gives nicer looking output.)
- Within a shell, execute `lpr` *filename*.

Either of these options will print your document on printer `psci1l`, which is the smaller printer near the mini-focus consultant's desk. If you use `lpr`, 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`.

In the past, the connections from some of the cluster machines to `psci1l` have been flaky (e.g., nothing prints out when you try the above). It remains to be seen if the flakiness persists this year.

8 Logging Out of Linux Machine

After you are done using a Linux workstation, you need to logout.

- If you are using 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.
- If you are just using a text-based interface (no windows), just type `logout` or `exit` at the Linux shell prompt.

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

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 `Red Hat:Lock Screen`. 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.

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