Objectives

After completing the lab you would be able to

1. Understand and use the GNOME Desktop Environment.
2. Use LINUX commands such as `look`, `cp`, `ls`, `man`, and `lpr`.
3. Understand the execution of a program by the Itty Bitty Machine.
4. Be comfortable asking questions of a fellow student or the lab instructor when something seems in need of clarification.

References: The following websites and documents of various forms will be used throughout the lab.

- *CSCI 203 course website* `<http://www.eg.bucknell.edu/~csci203/>`
- *The* LINUX *Documentation* (HTML): `<http://www.eg.bucknell.edu/docs/linux>`
- *The* LINUX *Documentation* (PDF): `<http://www.eg.bucknell.edu/docs/linux.pdf>`

Introduction

Welcome to what could very well be your first experience with a UNIX or LINUX computer.

We have designed this first lab for students with various levels of computer sophistication. You may have a lot of in-depth experience with computers, or may only have used one in a very simple, straightforward way; most likely you are somewhere in the middle. This lab was designed with the underlying assumption that you have used a computer, but may not have investigated many of the features which are available: moving windows around, changing their size, entering commands in an system window, and configuring the appearance of your *workspace*. We will touch on these issues in this tutorial, and some others as well.

If you have a lot of previous computer-related experience, you might be able to move through the tutorial pretty quickly. If you don't have that level of experience, we have tried to make the lab doable in two hours or less. Remember, if you get stuck, just raise your hand and help will be there shortly. It might also help to know that the principles behind the LINUX computing environment are very similar to those behind the MS Windows and Mac OS environments.

You can read *The* LINUX *Documentation* by following the link to *The* LINUX *Documentation* (PDF), or by visiting the CSCI 203 course website. This manual was written specifically for new system users and is referred to often herein. This lab will be difficult to complete without having the manual at your side. If you have trouble opening the document, try to get help from a colleague in the lab, your lab TA, or your lab instructor.

Notes about the Lab and Lab Assignments in General

Conventions

Since this is your very first lab assignment in this course, we'll say a few words about the conventions used in the lab descriptions.

To-Do box Lab assignment descriptions such as this one typically consists of sections, some of which discuss the concepts while others give you the work to do. The work for you to complete typically will be enclosed in a **To-Do** box. The **To-Do** box may start with very simple commands or specific mouse clicks. Later the **To-Do** boxes may contain complex tasks that use commands and mouse clicks introduced earlier.

Here is an example of a **To-Do** box.

To-Do: **To-Do Box Example**

1. Read through the instructions in this section.
2. Click the link in the next paragraph.

URLs Lab handouts will frequently refer to certain URLs. If you are reading the web version of the lab, you will be able to click on a link and open it in a browser window. In the printed version, the full URL will be listed and you will have to type it into your browser.

LINUX and UNIX

LINUX and UNIX are very similar from an end-user's point of view. The user interface and the commands available to users from the two systems are almost identical. The actual internal design and implementation of the two operating systems are different, but the design philosophy is very similar.

There are many versions of UNIX and many versions of LINUX in use. Bucknell uses Red Hat LINUX. Throughout the labs we use the two names interchangeably. Most commands are available in both systems. Our focus will be on LINUX.

The GNOME Desktop Environment

The GNOME desktop environment is similar to the windowing environment found in personal computers running Microsoft Windows or Apple Mac OS. If you are familiar with either of these two, then the discussion below should revisit points that you know, at least to a certain extent.

The working environment, that is, your computing *desktop* will appear once you have logged in. It will consist of a red background, sometimes called the *workspace* and a gray bar at the top and bottom of the screen called a *panel*. You will also see three icons on your desktop: a computer representing the computer to which you're logged in and its devices, a little house representing your *home* file folder and a trash can representing the folder for files recently deleted. You should see an arrow on the screen somewhere. This is the mouse pointer, as you would expect with a window-based system such as this.

> *If the screen doesn't appear as described above, something is amiss. Try logging out and logging in again; if that doesn't help, raise your hand and someone will help.*

This section of the lab will describe very briefly each of these desktop elements. Before we get into that, however, a brief word about the use of the mouse. Right-clicking activates buttons and opens menus. When you move the mouse around and the pointer comes close to an icon, that icon is brightened to indicate that you are in its *active zone*. On the desktop, left-clicking in an active zone once will only highlight the icon; to activate it, a double left-click is required. Left-clicking away from the icons' active zones and buttons produces no results. Items on the panels can be activated using a single left-click.

Later when you are using a text editor or other text-based application and you are manipulating text on the screen, you will find that left-clicking and dragging the mouse over text creates a highlighted section of text that can be copied into a different area or application. After you have selected text, pasting only takes a click with the rolling-wheel in the middle of the mouse. (If you have three-button mouse, this would be the click of the middle button.)

Clicking the right-button on the red background of your desktop (the wallpaper, if you will) brings up a menu. Moving the mouse cursor over the menu choices will let you choose an option. Actually left-clicking on that option once asks the system to complete that task for you.

More details about your desktop and its components can be found by going through Section 2.2 of the *The* LINUX *Documentation* which is titled *Introduction to the Desktop Components*.

Windows and Menus

With the mouse pointer on the red background click the right mouse button. This will bring up a menu. One of the menu items is the **Open Terminal** entry. Point to that entry and left-click the mouse, another window will appear. This window is called a *terminal* where one can issue commands and the results of these commands are displayed. Each terminal window has a title bar which reads *Terminal*.

To-Do: Creating A New Terminal Window

1. Point the mouse somewhere on the red background.
2. Click the right mouse button once.
3. Point to the **Open Terminal** entry.
4. Click the left mouse button once while the pointer is on the **Open Terminal** entry.

A terminal window consists of several different parts, the *title bar* which is at the top of the window where a fixed title *Terminal* is displayed, the *menu bar* which is immediately below the title bar where the actions you can take are listed, and the *command window* where you can issue LINUX commands and where the results of the command are displayed. At either end of the title bar of the terminal window are special icons. If you point the mouse on any of the icons (without clicking the mouse buttons), a small text box will appear to explain what this icon is. Typically, the **Windows Menu** icon (a monitor) is at the left side of the title bar. The **Minimize Window**, the **Maximize Window**, and the **Close Window** icons are on the right end of the title bar. If you left-click on the little monitor on the left, a menu will appear. This menu also appears when you right-click anywhere on the window's title bar. If you click on the **X** icon on the far right of a window's title bar, the window will be closed and disappear. Clicking on the icon to the left of the **X** (in the middle), maximizes the window filling the entire screen. If the window is already maximized, clicking on this icon will bring it back to its original size. The dash icon (the third icon from the right) causes the window to be minimized. As you will see the window shrinks down, a corresponding icon will appear on the gray bar at the bottom of the screen. When you want to bring it back, click on the window's icon.

You can change the dimensions of a window: If you put the pointer over any one of the four corners notice that the shape of the pointer changes. If you left-click and drag, the window changes size. A one-directional size change can be

accomplished by placing the pointer on any edge of the window, left-clicking, and dragging. When you do this, the shape of the pointer changes from single arrow to double arrow.

Finally, you can change the position of the window by clicking the left button in the center part of the title bar (that is, anywhere between the icons on the left and on the right) and dragging. The window doesn't change shape, only position.

To-Do: Terminal Manipulation

1. On the terminal window you opened during the previous **To-Do** session, left-click on the monitor icon on the title bar of the window to see the options from the menu.
2. Left-click the dash icon to minimize the window.
3. Left-click the icon corresponding to the window that is just minimized on the icon list at the bottom of the screen to re-open the window.
4. Resize the window to make it bigger and then resize it to its original size (approximately).
5. Drag and move the window to a different place on the desktop.
6. Click the monitor icon on the left of the terminal window's title bar and select the **Close** option from the menu.

Focus

At any time there may be multiple windows open on the desktop but only one active window where actions can take place (editing, issuing command and others). To activate a window, one needs to move the cursor to the window and click the left mouse button just about anywhere in the new window. The top border turns blue when a window becomes active. In this circumstance, we say that the new window became *active* or *in focus*. The commands you type will be recognized by that window alone. When a window becomes active, you will see that its title bar turns to blue and a blinking cursor at the command line, indicating when the commands can be entered.

To-Do: Active Terminal

1. Open two terminal windows.
2. Move the two new windows on desktop and resize them so they don't overlap but are next to each other.
3. Make one window active and type the LINUX command `ls` at the prompt. Now make the other window active and type the LINUX command `date` at the command prompt.

Configuring the Workspace

GNOME is fun to use partly because it can be easily tailored to the user's tastes and needs. You can create icons on your desktop corresponding to application programs by dragging a menu item from any program menu and leaving it on the desktop. When you do this, you will not destroy the original menu item on the program menu, but just create an additional copy on the desktop. At this point we will show you how to create a couple of useful icons on the desktop.

To-Do: Configuring Workspace Applications

1. Left-click once on the **Applications** menu on the top panel.
2. You see a series of icons for programs and folders similar to the **Start Menu** of Windows. Move your mouse point over the **Accessories** icon.
3. Left-click on the **Calculator** icon, but continue to keep the mouse button pressed.
4. Slide the mouse to move the pointer to the desktop and then let go of the mouse button.

You have dragged the **Calculator** item from the menu onto the desktop and created a new icon on the desktop. This is a short-cut for the calculator program, just like any Microsoft Windows short-cut. Removing the short-cut simply means taking it away from the desktop, not actually deleting the program.

The rest of the top panel contains (from left to right) some commonly used applications and a clock. The bottom panel contains (from left to right) a **Show**

Desktop button, tabs for all of your open applications, and a workspace partition (the box with four partitions, one of which is blue).

The first application on the top panel is the **Firefox** web browser (a blue globe with a red fox draped around it). This is a web browser similar to Microsoft's **Internet Explorer**. If you hover the mouse over the application icons, a text box will appear with some information about the icon. A single left-click on one of these icons will activate its program.

On the bottom panel, you will see a button that will hide all open applications and show your desktop.

To the right of that you will see icons for the current programs in this workspace. A single left-click on one of these programs will expand it if it is not active or minimize it if it is active.

Further to the right on the bottom panel, you will see an icon that looks like a box with four different partitions inside. One partition is blue with some smaller boxes inside, the rest are grey. The blue partition contains the *active* workspace you have been working on. Think of it as one desktop. If you left-click on any of the other partitions, you will be switched to a different desktop. It may be nice to have multiple workspaces when you need to have many windows open all at the same time and each focus on a different application.

To-Do: Working with Multiple Workspaces

1. Left-click each of the four workspace icons once.
2. In each of the workspaces, create one new terminal window.
3. Switch back to the original workspace where you have multiple terminals open.

Locking a Session

In an environment such as a public laboratory, it is often common for people to step away from their work temporarily for only a few minutes. When a circumstance like that arises, it is counter-productive to exit all the applications you are using only to have to restart them again upon your return.

You could always step away from your computer and leave your session running, but that has its drawbacks. Imagine that you are in the middle of something that requires a good level of privacy. If you leave your computer alone for some time, any passersby will be able to read your screen. An even worse scenario is

that a malicious user could sit at your keyboard and do all sorts of evil things, such as erasing your files, or sending email to anyone using your identity.

What comes in handy in these situations is the option that the system gives you to lock your computer. This allows you to freeze your computing session as it is, so that your work will be there waiting for you when you're ready to continue. You can do this using **Lock Screen** under the **System** menu. Once your computer is locked, you will have to enter your password again to unlock it and resume your work.

To-Do: Locking the Screen

1. Left-click the **System** menu on the panel at the top of the screen.
2. Left-click the **Lock screen** icon.

You screen should now turn dark with nothing on it. The computer essentially goes into a screen-save mode. In order to get back into your account again, you first need to activate the screen from the screen-save mode, then enter your password to get back to where you were before leaving the computer the last time.

To-Do: Unlocking the Screen

1. Press any key on the keyboard or click any of the mouse buttons to activate the screen.
2. Type your password in the password box and re-enter the workspace.

Important: Your computer will automatically lock itself if it is inactive for more than 5 minutes. If your computer is locked for more than 10 minutes, a logout button will appear that enables other users to end your session so that they can use the computer. Use locking only if you plan to return in a few minutes. Otherwise, you may lose some of your work.

Logging Out

When you have finished a work session and are ready to leave the lab, you should log out of the computer so other users may use it.

To-Do: **Logging Out**

1. Left-click on the **System** menu and then on the **Log Out** item.
2. Left-click the **Log Out** button. (If you change your mind during this process, you can always click the **Cancel** button and get back to the computing session you were in.)
3. Log back into the computer. You now should be back to the screen that you left before. There should be some terminals open in the current workspace.

A Few Useful LINUX Commands

In the sections to come, you will be asked from time to time to write something down, such as the results of a command or what you think of the results. In this lab, these writings will be sent to the instructor/TA through email for grading. Before you learn how to use an editor or send email, you might want to write these results down until you are ready to submit them. Now, let's move onto a few things you are perhaps not familiar with.

The LINUX operating system is a complex and powerful operating system and the backbone of Bucknell's extensive network of computers. In this lab we introduce you to just a few basic LINUX commands. Then as the semester progresses, we will introduce a few more commands and/or new options for previously discussed commands.

Here we refer you to *The* LINUX *Documentation*. Read section 3.2 *Interacting with Linux from the Command Line* and do the examples in that section.

Once you have worked through the examples which involve the `date` command, and you think you have a handle on what you did, try the `look` command as follows — be sure to leave a space between `look` and `frus`.

To-Do: **The LINUX Command `look`**

1. Type `look frus` at a command prompt.
2. Explain what the command `look` would do for any input; not just the specific output of the parameter `frus`.

In your explanation, try to point out the connection between what the *word*

"look" means and what the *command* `look` does in general. *Please Note*: this is not a trick question and can be answered in a few short sentences.

The next few commands are among the commands you will use the most. *It is very important that you pay attention to spacing in these commands: many of your early problems will stem from putting spaces where they don't belong or leaving spaces out where they do belong.*

A more complete discussion about LINUX commands is available to you in Section 3 of *The* LINUX *Documentation*.

Copying Files

Now that you have seen the concept of the LINUX command line from doing the examples in the *The* LINUX *Documentation*, we will introduce some essential UNIX commands.

The first is the `cp` (copy) command, which is discussed in section 3.4.5 *Copying Files* of *The* LINUX *Documentation*. Later in this lab, you will be asked to load an IBM (Itty Bitty Machine) program into the IBM and run it. The filename of this program is `lab1.ibm`. (The IBM program requires IBM files to have the extension `ibm`.)

The file `lab1.ibm` is currently located in a directory named

```
~csci203/2009-fall/student/labs/lab01
```

(We know we haven't told you what a directory is yet — it's just another name for a *folder* in Windows or Mac terms — we're saving that for next week.) The `cp` command takes two parameters, the source file and the destination file. If you intend to keep the file name the same after copying it, the destination file name can simply be a dot (**.**) when you issue the copy command.

To-Do: Copying the IBM Program

1. Type the following LINUX command:

   ```
   cp ~csci203/2009-fall/student/labs/lab01/lab1.ibm .
   ```

Beware! There is a space after the command `cp`. There is no space between the tilde character (`~`) and `csci203`. The only other space on the line is between the `ibm` and the final period (**.**) which we will learn indicates the destination file takes the same name as the source file. Thus that final period is an important part of this command line!

The `cp` command instructs the system to copy the file `lab1.ibm` from the

`~csci203/2009-fall/student/labs/lab01`

directory into your account and name the copy of the file `lab1.ibm`.

More on copying files is available in Section 3.4.5 of *The* LINUX *Documentation*.

Checking on Files

To verify that this file was correctly copied into your directory, you will use the `ls` (list) command. A discussion of this command can be found in Section 3.4.6 of *The* LINUX *Documentation*. Try out the examples in the section **Listing** of the document and make sure you have a fair idea of what the command does. You should see the file `lab1.ibm` in your list of files. If this file is not there, look at what you typed and compare it to the appropriate command above. If everything matches and you don't have the file, *ask for help*. If you typed in the command incorrectly, try it again. *Make sure you have this file in your directory before you continue.*

At this point, you need to make some minor changes to the default settings about your account. The Engineering Computing Support Team (ECST) has made this easy for us by including necessary actions in a single command.

To-Do: Setting Up Your LINUX Environment

1. Type the following command at the LINUX prompt:

   ```
   csci203setup
   ```

This will alter your `.cshrc` file (an initialization file that is read when you open a terminal window) so that you can run the "Itty Bitty Machine" software. In order for this change to take effect, you must either start a new terminal session or reread the `.cshrc` file.

To-Do: Making the Environment Change Effective

1. Exit your current terminal session by typing

   ```
   exit
   ```

   at the prompt.

2. Open a new terminal window.

To-Do: Making the Environment Change Effective

1. Type the following command to make the change in `csci203setup` effective in your current terminal. You need to do this for every terminal you have opened. This will reread the initializatiion file `.cshrc`.

   ```
   source ~/.cshrc
   ```

Using the LINUX Manual Pages

Perhaps the most important command we can tell you about today is right now the least useful to you. However, trust us when we say that this will be your very best reference when working with LINUX in the time to come.

Give it a try right now! In a terminal window, try a few LINUX manual page commands.

To-Do: LINUX Manual Page (`man`)

1. Type the following command in a LINUX terminal

   ```
   man man
   ```

2. Hit spacebar to get to next page.

3. Type `q` to quit.

The first word in the line above represents a command to consult the LINUX online manual. It's a simple mnemonic: `man` for *manual*. The second word in the line represents some LINUX command that we would like to look up in the manual pages. In this case, we're using the manual to look up the command `man`. At this point, the explanations in the manual page may look a little cryptic to you, but as you gain more experience with LINUX, things will make more and more sense.

Try now to look up the other LINUX commands we used before.

To-Do: **LINUX Manual Page (`cp` and `ls`)**

1. Type the LINUX command

   ```
   man cp
   ```

2. Type the LINUX command

   ```
   man ls
   ```

or try any other commands you may know. Skim the pages for your own benefit, so that you can start getting used to this kind of language. While viewing the manual pages using the `man` command, you can press the spacebar to get to the next page or press `q` to quit the manual pages.

Printing Files

The final command we will discuss is one that allows you to print files. To send the file `lab1.ibm` to the printer in your lab you would issue the following command. **Don't issue this now!**

```
lpr lab1.ibm
```

It is convenient that the system is set up so that when you use a command with this form the output will always go to the printer in the lab where your current machine is located.

However, if you want to print to a printer in a different room, you must slightly change the form of the print command to indicate the printer to use. If you take a look at section 3.7 *Printing Files* of *The* LINUX *Documentation* you will find a list of the names of printers in various rooms and labs. For example, to send a file to the printer in Dana 213 (from any room) you would issue the following command. **Don't do it now!**

```
lpr -Pdana213-lp1 lab1.ibm
```

Now, print a copy of `lab1.ibm` to the printer in your lab by using the first command above.

To-Do: **Printing in LINUX**

1. Type the following command

   ```
   lpr lab1.ibm
   ```

Knowing how to send files to any one of the printers listed in *The* LINUX *Documentation* is an essential skill. More on printer names is available to you from Section 3.7.1 of *The* LINUX *Documentation*.

Throughout the semester, there may be times when one or more of these printers is not working; to make matters worse, the room where the only working printer is located may be hosting a class at the time you need to print. Since students have access to so many printers, the line "The printer wasn't working" is rarely an acceptable excuse for not handing in programs on time.

Make absolutely sure you understand the printing commands before continuing.

The Mailer

When sending or reading mail, there are many options such as Outlook, a browser-based email system such as Hotmail, or the Bucknell Webmail system. Outlook is not available on the LINUX systems. If you want to use email while in the lab (or otherwise logged into the LINUX systems) you will have to use one of the browser-based alternatives. The Bucknell Webmail system is probably the most convenient. You can access Bucknell's Webmail from anywhere on the Internet.

To-Do: Bucknell's Webmail

1. Activate one browser, if you haven't done so.
2. Type the following in the Firefox browser that is open.

   `http://webmail.bucknell.edu/`

3. Type in your user name and password to make sure you can get into the Bucknell webmail system.

We will assume that you know how to use one of these email systems. In fact, we expect that you read your email frequently, that is, at least a couple of times a day. At this point, if you are unsure how to use email on the LINUX systems, please ask the instructor or TA for help.

The Itty Bitty Machine

We've talked about the IBM in class. Now you will get to see it work.

To-Do: **Starting IBM**

1. At the LINUX command prompt, type the following:

```
ibm &
```

A window should open; you are now looking at the IBM. Notice that all 100 memory locations contain "000." The window starts out with two drop-down menus at the top, labeled **File** and **Help**.

The IBM program is similar to many other GUI-based (Graphical User Interface) programs. One can click on menu items and issue commands. IBM is a program that simulates the workings of a simple computer with a CPU (Central-Processing Unit) which has registers and instruction execution unit, an array of memory cells, one input device (a simulated keyboard) and one output device (a simulated screen). With this simulation program, one can load a program written in a language that is understood by IBM. The execution of the program can be in one of two modes, single-step, or continuous execution. In single-step execution mode, one can watch the change of the simulated computer's state shown by changes in memory cells and in registers. When an input is needed from the program, the user would type one character at a time from the real keyboard into the simulated keyboard which is shown at the lower left corner of the GUI. The results of the program are displayed in the simulated screen which is shown at the upper right corner of the GUI.

In the following **To-Do** box, you are asked to work with this program to accomplish some simple tasks. You will learn more details about IBM as the semester goes on. For more detailed discussion of IBM, please read the booklet *An Introduction to the Structure of Computers and Programs* which is available from the course website.

Pay very close attention as you do the following exercise. You will be asked to explain what happens a little later as a part of the exercises.

To-Do: **Working with IBM**

1. With the IBM program running, click on **File** and then select the **Load** menu option.
2. When a file browsing window opens with the title *Load Which File?*, look for the list which is labeled **Files** and select the file named `lab1.ibm`.
3. Click on the **OK** button at the bottom of this window.
4. After clicking on **OK** you should see the interface change, with numbers being written into the memory and CPU registers.
5. Move the mouse pointer over the red word **Step**. The letters should turn green, then turn back to red if the mouse moves away from the word.
6. Move the mouse pointer into the region labeled `Keyboard`. A green border lights up around the region which means you *may* type characters from the real keyboard as the input to IBM. Don't do this yet.
7. Move the mouse pointer to the **Step** button so the word **Step** turns to green.
8. Click the **Step** button twice in succession. Note the changes in the **Instruction Register** and the **Program Counter**, as well as the value displayed in the simulated screen.

To-Do: **Working with IBM (continued)**

*Continued from the previous **To-Do**.*

9. Click the **Step** button one more time, the IBM now is waiting for your input.
10. Move the mouse to the simulated keyboard area (you should see the board of the **Keyboard** area become green), type the digit 5 on your real keyboard. Click the **Step** button twice.

    The digit 5 should appear in the simulated screen.
11. Repeat steps 9 and 10 two more times, entering the digits 7 and 9, respectively.
12. Repeatedly click the **Step** button and watch the changes in IBM, until the **Step** button changes into a **Halted** button. This means the program is completed.

The previous **To-Do** box lets you watch the execution of the IBM program `lab1.ibm` once. You can restart the program by going to the **File** menu and selecting the **Reload** option. Repeat this **To-Do** if necessary until you feel comfortable you can explain what the program is doing.

Hand In

1. Send your *lecture professor* an e-mail message containing the following:
   - Your description of what the `look` command does. Remember, only a few sentences are required. If you get help from *any* source other than your own intellect, make sure to state what it was. Get used to giving credit when credit is due!
   - A *brief* explanation of what the IBM program you loaded and stepped through did. Again, we are not expecting an essay.
2. Remember to log out of the system before leaving the lab. Leave the machine on for the next user.