Lab 0

Working with Linux

The purpose of this part of the lab is to introduce you to the Computer Science computer lab and the UNIX operating system, which includes the use of the UNIX command-line and file system. You will be working with several basic UNIX commands including those for dealing with directories and files.

0.1 Introduction to UNIX

0.1.1 Operating Systems

An *operating system* is a sophisticated program that manages a computer's resources. There are many different operating systems in use today. The most popular ones are Microsoft Windows (TM), MacOS (TM), and UNIX (TM).

0.1.2 UNIX

UNIX is a powerful operating system designed to be a multiuser and multitasking system. The original UNIX was created by Ken Thompson in 1969 at Bell Labs. Today, the term "UNIX" does not refer to a single operating system sold by a single company. Instead, it refers to any operating system that meets certain standards.

Most large-scale computers and some desktop personal computers use an UNIX operating system, which could be a generic system or one written by the computer manufacturer. Some of the more popular UNIX operating systems include Linux, Ultrix (DEC), Irix (Silicon Graphics), and Solaris (Sun Microsystems).

Since each UNIX operating system must meet the same standards, they function similarly. Thus, after you are familiar with the use of one UNIX operating system, you will easily adapt to a different one. The major differences are usually in the administration of the system—meaning, unless you are the administrator of the system, you never have to worry about that aspect.

0.1.3 Linux

The Linux (Lynn-ucks) operating system was created by Linus Torvalds in 1991 while he was a graduate student at the University of Helsinki (Finland). Torvalds created Linux as an alternative to Microsoft Windows and to provide a UNIX operating system for use on the PC. Linux is and has always been an open-source project that allows other programmers to view and modify the source code. Today, hundreds of programmers work on Linux—mostly in their spare time—under the direction of Torvalds.

The Linux operating system is very popular today due in part to its availability and open source status. There are a number of Linux distributions available from different companies and groups such as Red Hat, Slackware, SuSe, and Corel. All of these use the same Linux operating system. The major

differences between the distributions are the services provided and the various applications included with the Linux distribution.

Linux is very powerful and is easy to learn and use. All major distributions provide a graphical user interface frontend that will be familiar to Microsoft and Apple users.

0.1.4 Interface

Users commonly interact with a UNIX system via a text-based command-line interface. In a *terminal*, commands are entered at a *prompt* and results are displayed. Numerous commands are provided for file and directory manipulation, program execution, and file processing. With a text-based interface, users can work with the system directly on the physical machine or connect to a remote system via the Internet. One of the biggest wins with the text-based interface is how easy it is to *automate* complicated processes. We won't be automating processes in this class.

In addition to the text-based interface, most UNIX systems also provide a graphical interface similar to those of the Macintosh and Windows platforms. Unlike those systems, however, the graphical interface under UNIX consists of several layers as illustrated below.
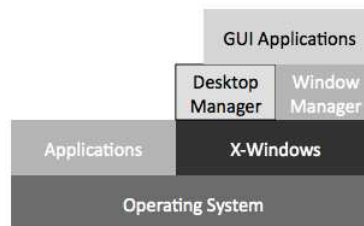
The actual graphical environment is provided by a standalone program called X-Windows (TM). This program is executed on top of the text-based interface. X-Windows is responsible for managing the monitor, keyboard, and mouse in addition to providing the "windowing capability".

The look and feel of the graphical environment is provided by a *window manager*. The window manager is a standalone program that works in conjunction with X-Windows to provide decorations for windows, buttons, and other components in addition to controlling the behavior and action of the various components. There are a number of window managers from which to choose. The default on our systems is the *Metacity* window manager, which is highly configurable and provides a number of "themes" that change the look and feel.

Linux systems also provide a *desktop manager*. The desktop manager is responsible for the icons on the desktop, the panels that provide menus and application launchers, and virtual workspaces. The two most popular desktop managers are GNOME (default) and KDE, both of which are provided on our system.

0.1.5 The Computer Lab

The Computer Science department maintains a system of networked UNIX workstations in both the teaching lab (Parmly 405) and the advanced lab (Parmly 413). The machines in Parmly 405 are to be used for this course. If you need to work when a class is using Parmly 405, you may use Parmly 413.

The computers in Parmly 405 run both Linux and Windows; they run Linux natively and Windows virtually. See the instructions in the lab for how to switch between them. The machines in the advanced lab are Linux-only PCs.

The computers in Parmly 405 are uniquely named after programming languages (e.g., c, basic, fortran, etc.), while the machines in Parmly 413 are uniquely named for an individual who made a major contribution to the field of computer science. All of the machines are connected to our central file server.

UNIX workstations are not like stand-alone PCs. The Central Processing Unit (CPU) of one machine can be utilized by a user sitting at the terminal, by a user at another workstation in the lab, or by a remote user (or by many of these users simultaneously). Therefore, when using a Linux PC (like those in Parmly 405 and 413), it is important that you do **not** turn the machines off as you would a stand-alone PC.

0.1.6 User Accounts

To use a UNIX system, you must have an account and password. You will log on to gain access to the system and then log off after you are done. The account/password provides protection to your data stored on the system. Thus, it is important that you protect your password and not allow others to use it. Likewise, it is important that you **log off the system after you are finished**. If you forget to log off, another person can gain access to the system as if they were you. **Do not lock your machine** so that other students cannot use the machine.

0.2 Getting Started

You should have received a paper with your username and password. You will need this to log onto the Linux system. Note that while your username is the same as that assigned by the University, you have a *different* account and password on the Computer Science Department server and the two accounts are not interchangeable.

0.2.1 Login

You are now ready to log onto the system for the first time. After clicking on "Other", you should see a screen containing a dialog box similar to that shown below.
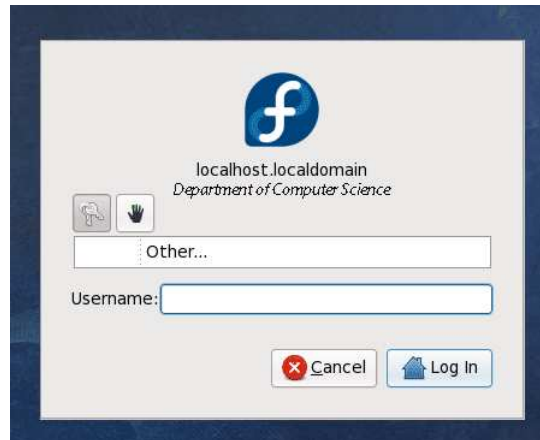
This is the Linux login prompt. If you do not see this prompt, then someone may have forgotten to log out of this machine or to change the machine to Linux. Ask for assistance on how to log out the previous user or move to a different computer.

Enter your account name in the "Username" text field and press ⎡Enter⎤.

If you have trouble entering text into the box, you will need to move the mouse over the dialog box. Next, you will be prompted for your password.

Enter the password provided on the "new account" sheet.

You should now be logged in to the system. If not, ask your instructor or lab assistant for help.

0.2.2   Exploring the Desktop

Open the *Firefox* web browser by clicking on the Firefox browser icon (shown below), which is located on the desktop panel at the top of the screen.

Using the browser, navigate to the course Web page. There are lots of different ways to navigate to the course Web page; you should be able to find one that works. Bookmark the course Web page. You could even edit your preferences to make it your home page!

There is a link at the top of the Web page labeled "Resources". Click on the link to open the Resources page. Then scroll down to the "UNIX" section and select "Lab Resources".

The Lab Resources page contains information about the system and various applications.

Click on the "Labs" link in the top right.

Among other things, this page links to two documents that provide an introduction to UNIX and the Linux desktop. The documents give you a basic introduction to the Linux desktop used on the computers in the Computer Science Department. You may want to refer to these documents later in the semester to learn more.

Play with the Linux Desktop yourself to see what you can do!

0.3   The Terminal

Before you can begin working with UNIX commands, you need a *terminal window*, which runs a program called a *shell*. The shell provides an interface between you and the operating system.

⚠ Open a terminal window from the Applications menu, under "System Tools".

⚠ Go through the Applications menu to the "System Tools" menu, grab the terminal icon, and drag it to the top panel to use as a shortcut throughout the course.

The terminal window on your desktop should contain some characters that look something like

```
[username@lisp ~]$
```

This is called the **prompt**. The prompt gives you information about the account and machine being used. For example, the prompt above is for someone whose username is *username* and who is using the computer named *lisp*. The prompt indicates that UNIX is waiting for (or "prompting") you to type something. Whenever you type something after a UNIX prompt, UNIX tries to understand it as a command. If you type a command that UNIX understands, UNIX carries out the command. Otherwise UNIX displays a message indicating that the command was not recognizable.

Note: Commands and filenames in UNIX are case sensitive.
Spaces are required between commands, arguments, and options.
The basic template for a UNIX command is

```
command-name [options] [argument1] [argument2] ...
```

To enter a UNIX command, the terminal window must be the *active* window. To make a window active, simply move the mouse so that the mouse pointer is located within the limits of the window and click on the window. When a window is active, its border changes color.

⚠ Make the terminal window the active window.

⚠ At the prompt, type in your last name and press Enter .

You should see the error message that UNIX displays when it cannot interpret what you type as one of its commands (unless your parents happened to name you "ls" or "cp" or some such thing).

In the rest of this lab, we'll focus on commands that UNIX *does* recognize and on how to use them.

0.4   The File System

The basic unit of storage in UNIX is a **file**. A file may contain many kinds of information, including a Python script, an HTML document, a research paper, an image, or an executable program. Files are organized in a hierarchical system of directories. A **directory** may contain files and other directories. The directory at the "top" of the file system, in which all other directories are located, is called the **root** directory.

Every account owner has their own **home directory** within the file system. All of the work that you do in lab will be stored here.

The shell remembers the current **working directory**, which is the directory that commands will, by default, be executed in when you type them. When you start a terminal, the working directory is your home directory.

Your home directory is currently empty except for several configuration files, a "public_html", and "Desktop" directory. You will be creating new directories and adding files throughout this lab assignment.

0.4.1   Listing Files

Most UNIX commands are abbreviations for a verb that describes what the command does. Some of these abbreviations may seem strange at first, but as you use them, they will become more natural.

Let's start by looking at the files in your home directory. To view the contents of the current directory (which is your home directory at this point), use the UNIX command `ls`, which is short for "list".

> To see the `ls` command in action, type
>
>   ls
>
> and press Enter .

> **Note:** From this point on, if you are instructed to "enter" a command, you should type the given command and then press Enter .

A list of names should appear in the terminal window. These are the names of the files and directories contained in the current directory, which happens to be your home directory. Since you're just starting,

there aren't many files or directories in your home directory (the current directory). Let's look at another directory.

⚠️   `ls /home/courses/cs111`

Now you see the names of the files and directories contained in the directory /home/courses/cs111. The name of a subdirectory will have a / following its name. For example, `turnin` is a subdirectory of the /home/courses/cs111 directory.

In this example, /home/coures/cs111 is the **argument** to the `ls` command. By default, i.e., when you don't give `ls` an argument, `ls` lists the names of the files in the current directory.

0.4.2 Command Options

There is more to most commands than just the commands themselves. You can change the behavior of most commands with options. **Options** consist of a minus sign (-) followed by a letter or two, and are typed on the line with the command just to the right of the command itself.

There are many options that you can apply to the `ls` command. One such option is `-l` (where the option -l is a lowercase L).

⚠️   Add the `-l` option to the `ls` command by entering

       `ls -l /home/courses/cs111`

Question 0.1 *Describe briefly how the `-l` (lowercase L) option modifies the action of the `ls` command. (You do not have to understand the information displayed.)*

💡   **Note:** Answer this and future questions on the Question Worksheet at the end of this lab. You will hand in the Question Worksheet at the end of this lab session.

Question 0.2 *List the subdirectories and files within the directory* /home/courses/cs111/handouts/lab0.

As with other operating systems, wild cards can be used with the UNIX file and directory commands. The * can be used to replace 0 or more characters within a filename, while the ? can be used to replace a single character.

Question 0.3 *Enter the command*

       `ls /home/courses/cs111/handouts/lab0/*.py`

       `ls /home/courses/cs111/handouts/lab0/example.*`

For each command, what files are displayed and what do they have in common?

> **Note:** Because of several advanced features of the shell program, avoid using blanks (spaces) and other special characters (e.g., &, /, <, and >), within file and directory names.

0.4.3 Shortcuts

Recall that the **shell** is a program that provides an interface between you and the operating system. The shell provides a method for recalling previous commands entered at the UNIX prompt. To recall a previous command, use the $\uparrow$ and $\downarrow$ keys to traverse through a list of the previous commands you entered at the prompt. In addition, you can edit a command at the prompt by using the $\leftarrow$ and $\rightarrow$ keys. After selecting a command using the arrow keys, the command can be executed by pressing Enter .

⚠ Try using the $\uparrow$ to retrieve some of your recent commands. This feature is very useful when you repeat a command frequently.

Another useful shortcut provided by the shell is filename completion, which will complete a filename or directory in a UNIX command for you so that you do not have to type the entire filename.

⚠ For example, type the following at the prompt
```
ls /home/c
```
and press the Tab key.

The shell should have completed the name of the directory to /home/courses/ since that is the only possible completion within the /home/ directory.

⚠ Press Enter to execute the command.

**Question 0.4** *List the contents of the /home/courses directory.*

If there is more than one possible filename that could complete the partial filename that you typed, the shell will finish as much of the name as possible and then display the possible completions.

⚠ Type the following at the prompt
```
ls /home/courses/cs
```
and press the Tab key.

It is unclear which file or directory you want to list as there are several directories in the /home/courses/ directory starting with the letters cs.

⚠ Press the Tab key again.

This time, the shell should display a list of all the files and directories in the `/home/courses` directory starting with `cs`. At this point, you can either type in the entire filename yourself or as much as necessary to distinguish it from the others.

These shortcuts can *really* reduce the amount of typing that you need to do. Using the shortcuts makes using UNIX much easier.

0.4.4   Changing Your Password

When you logged into the system for the first time, you used a computer-generated password. This password is not something that is easy to remember. Don't worry: you can change your password to something you prefer.

Enter the command:
```
passwd
```

For your security, what you type for your passwords will not show up in the terminal.

You will be prompted to enter the current password (refer to your sheet) and then enter a new password and verify the new password. The new password must be at least 6 characters and have both upper and lowercase letters, or non-letters.

> **Note:**   This account is different from your University account. Thus, changing this password only affects your CS Department account.

0.5   Working with Directories

Most file and directory manipulation can be performed through the "File Browser" as is done in Windows (TM). However, there are many instances where you will need to work from the command-line. To give you practice, you are going to create and then use several new directories.

0.5.1   Creating Directories

When you work on assignments for this course, you will be told what directories to make and what to name them so that everyone's lab directories will be similar.

The command `mkdir` stands for "make directory".

Enter the following command to make a directory named `labs` in your home directory
```
mkdir labs
```

Now enter `ls`.

By default, the `ls` command shows the contents of the current directory. Since your current directory is your home directory, the contents of your home directory is shown in the terminal. The `labs` directory will help to keep your lab files organized throughout the semester.

### 0.5.2   Navigating Directories

You now know how to create directories and view their contents. The next thing you need to know is how to change to a given directory or more specifically how to change your current working directory[1].

UNIX file systems are organized in a hierarchical system of **directories** that resembles an upside down tree. Consider the following incomplete sample file system:
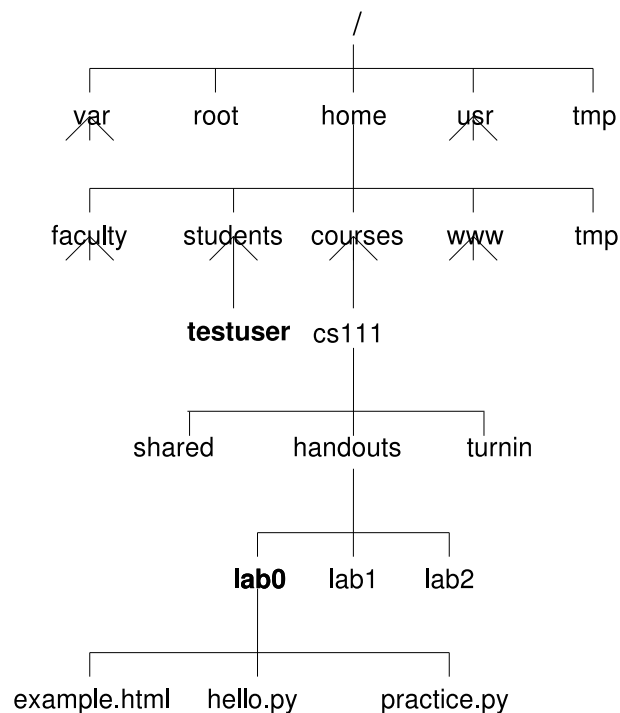


The complete path name for a file or directory starts at the root of the tree. The home directory of the *testuser* user would be `/home/students/testuser` while the path to the `handouts` directory within the `cs111` courses directory would be `/home/courses/cs111/handouts`.

> ⚠️ At the prompt, enter the command `pwd`, which displays your current working directory, i.e., the command **p**rints your **w**orking **d**irectory.

**Question 0.5** *What is the full name of your current working directory?*

To make your `labs` directory the current directory, use the `cd` or "**C**hange **D**irectory" command.

---

[1]As indicated earlier, the *current working directory* is the default directory used by the various UNIX commands.

Enter the command
```
cd labs
```

**Question 0.6** *What is the full name of your current working directory now?*

File and directory path names can be given *relative* to the current working directory or *absolute* by providing the complete path. In the above example, cd labs, the directory name labs is provided relative to your current working directory and, thus, it must exist within your directory. On the other hand, the command could have been given as

```
cd /home/students/testuser/labs
```

In this case, the path name is an absolute or complete path starting at the root of the tree. Typically, you will use relative names to change to a subdirectory within the current directory and an absolute names to change to a subdirectory within a different part to the tree.

Make a new subdirectory named lab0 within your labs directory.

Go into (i.e., change to) this new directory.

Question 0.7 *Are there any files within your* lab0 *directory?*

We have seen that the cd command allows you to change the current directory by specifying a subdirectory. The cd command also allows you to enter the directory that contains the current directory, called the **parent directory** or "one up" directory. To change directories to the parent directory, we use the special .. argument.

Enter
```
cd ..
```
at the command-line prompt.

Question 0.8 *Which directory are you now in and what is its contents?*

To return to your home directory use the cd command with no arguments.

Enter the command
```
cd
```
which should return you to your home directory.

0.6 Copying Files

The command to make a copy of an existing file is the cp command. It accepts 2 arguments: (1) what you want to copy and (2) where you want to put the copy.

⚠ Change to your home directory if that is not your current directory.

We are going to copy the file `hello.py` from the course's `lab0` folder to your `lab0` folder.

⚠ Enter the command (CAREFULLY! Use file completion to help.)
`cp /home/courses/cs111/handouts/lab0/hello.py labs/lab0`

The first argument is the source file (given with an absolute path) to be copied and the second is the destination directory (relative to the current directory).

⚠ Make your `lab0` directory the current directory.

To copy a file into the current directory, use a period, which is the special name for the current directory.

⚠ Enter the command
`cp /home/courses/cs111/handouts/lab0/practice.py .`

The file `practice.py` should have been copied into your `lab0` folder. You can also create a duplicate copy of a file within the same directory using the copy command.

⚠ Enter the command
`cp practice.py practice.py.bak`

This should have created a duplicate (a backup) copy of the file `practice.py` within your `lab0` folder.

Question 0.9 *What are the contents of your* `lab0` *folder?*

Sometimes, we want to copy the entire contents of a directory to another directory. The `cp` command can be used to perform this operation. You'll do this at the end of most labs this semester when you copy your lab directory into your "Turnin Folder" on the Department's file server.

Look at the contents of your turnin folder (`/home/courses/cs111/turnin/username` , where username is your `username` that you used to log into the system) contains.

⚠ Change to your `labs` directory and enter the command
`cp -r lab0 /home/courses/cs111/turnin/username`
where username is **your** username that you used to log in to the system.

This command copied the contents of your `lab0` directory (including the directory itself) to your "Turnin Folder" on the Department's file server. You will execute this command at the end of most labs.

0.7   Deleting/Removing Files

You can remove files using the `rm` command, followed by the name of the file(s) to be deleted.

⚠ Navigate back to your `lab0` directory.

⚠️ Enter the command
        rm practice.py.bak

The shell will request that you verify that you really want to delete the file. Type 'y' or 'yes' to delete the file. Any other letter will cancel removing the file.

0.8   Getting Help

UNIX provides an online manual that you can access to obtain more information about any UNIX command. Each manual page contains information about a command's purpose and correct use.

The man command allows you to find information in the command's manual. To get more information on a UNIX command, simply type man followed by the name of the command about which you want information. The man command returns the name of the command, a description of what the command does, a list of options that can be used with the command, and information about any known bugs associated with the command. This command is most useful when you want more information about how to use a command.

⚠️ Try the command by entering
        man ls
      at the command-line prompt.

Information about the ls command should be displayed. Scan through this information a line at a time by pressing ⌈Enter⌋ or a page at a time by pressing ⌈Space⌋. When you have finished viewing the information you may exit the man page by pressing ⌈q⌋ (for "quit").

Question 0.10 *Use the* man *command and describe what the* -r *option does when used with the* ls *command.*

UNIX does not have a file attribute for hiding system and configuration files. Instead, the convention is that all files and directories whose name begins with a period are hidden with the normal use of ls and other file browsers.

Question 0.11 *List all files within your home directory whose name begins with a* .b *(that's a period followed by the letter b). Note that you can do this a few ways—using an option described in the* man *entry or using a technique described earlier in this lab.*

0.9   Finishing Up This Exercise

When you are finished with this part of the lab, turn in in your completed "Question Worksheet" and be sure your name is on the first page.

Then, return to the lab in the web browser.

0.10 Question Worksheet

Name:

_____

Question 0.1 *Describe briefly how the* -l *(lowercase L) option modifies the action of the* ls *command. (You do not have to understand the information displayed.)*

Question 0.2 *List the subdirectories and files within the directory* /home/courses/cs111/handouts/lab0*.*

Question 0.3 *Enter the commands*

> ls /home/courses/cs111/handouts/lab0/*.py
>
> ls /home/courses/cs111/handouts/lab0/example.*

For each command, what files are displayed and what do they have in common?

Question 0.4 *List the contents of the* /home/courses *directory.*

Question 0.5 *What is the full name of your current working directory?*

Question 0.6 *What is the full name of your current working directory now?*

Question 0.7 *Are there any files within your* lab0 *directory?*

Question 0.8 *Which directory are you now in and what is its contents?*

Question 0.9 *What is the contents of your* `lab0` *folder?*

Question 0.10 *Use the* `man` *command and describe what the* `-r` *option does when used with the* `ls` *command.*

Question 0.11 *List all directories within your home directory whose name begins with a* `.b` *(that's a period followed by the letter b).*