

Lab Assignment #1
Copyright©2004 by Bryan Brady and Ivan Kourtev
Copyright©2006 by Steven P. Levitan

University of Pittsburgh
Department of Electrical and Computer Engineering

1 Objective

The objective of this handout is to help you get familiar with the UNIX/Linux environment that will be used throughout this course. Basic definitions and usage examples for most popular UNIX commands are demonstrated. Our version of UNIX is *Linux Red Hat Enterprise Release 4* running on Intel 64 bit quad-core workstations. Throughout this document we use the generic name UNIX and Linux interchangeably.

2 Starters guide to UNIX

UNIX is a large, diverse and very rich operating system. UNIX is memory-efficient, easy to maintain and promotes efficient program development. Unfortunately, UNIX is not as user friendly as some other popular operating systems! Keeping this in mind, this tutorial is organized so that beginners can start with simple functions and progress learning about the UNIX system on their own.

Please try the examples at your workstation as you read through this text. The best way to learn about the UNIX system is by personally using it. Soon, you will be an experienced user in a general-purpose multi-user UNIX system!

2.1 Workstation Information

As you read this, you are hopefully sitting in front of a Intel x86_64 workstation which runs the Linux operating system. Red Hat provides a graphical user interface (gnome) for Linux and is quite easy to use. Assuming one is familiar with Microsoft Windows, it should be relatively easy to get started using Linux. Please look at the keyboard more carefully. You will notice that some of the functional keys are misplaced compared with a typical PC keyboard. We will not be using the dark colored keys on the left side (such as Stop, Again, Copy, and so on). Please avoid pressing these keys by mistake as this may cause unexpected action and confusion. Experienced PC users or fast typers may have a problem with the Control, Back Space, and Caps Lock keys (placed differently than on a typical PC keyboard). Getting used to these may take some time.

2.2 Logging in

You will use the ECE 1192/2192 user account which was given to you in class. If you have problems using it, consult your TA. Enter your user name, Return and password, as it is done in the universal conventional way. **Remember that UNIX is case-sensitive for all operations** so make sure you are using correct letter-case.

2.3 Getting introduced to Red Hat Linux

Welcome to the UNIX world! Linux is pretty much similar to Windows OS as it has windows popping up for different utilities and tools. With the “gnome” windows manager, opening, closing, and resizing windows is done as in Windows environment. Most windows can be resized by dragging their edges.

NOTICE: If you feel uncomfortable in working with windows, like resizing or moving them, or even using the mouse, scroll bars, etc., start reading through the Help viewer, or let your TA know.

Note the large task-bar at the bottom of the screen. On the right are four buttons which are 4 predefined different Workspaces. Go click them all once to see the background change, notice that all are different workspaces, independent from each other. From user's perspective, we could think this is like using four different monitors of a single computer.

Now, start a text editor in Workspace One by clicking on Applications → Accessories → Text Editor. Once the text editor is running, write something in it (don't save yet). Click the button for workspace 'Two' on the taskbar and see the text editor doesn't show up in here! Try 3rd and 4th workspaces, too. You have four independent workspaces on your workstation; you can make use of them if desired. Go back to Workspace One and right click the top center bar of the text-editor window. Choose 'Move to Workspace', click on Two and 'OK'. See the text editor has disappeared. Choose 'Two' as the workspace from the Front Panel, and see your text editor placed there. This way you could change your workspace. Try, but not now, other options for workspace occupying choices, like occupy all workspaces, etc. For now, save the file as *sample.txt* in the *home* folder.

Check out the Applications and Actions menus from the upper left task bar, don't take time to execute them right now. We will be using Snapshot (to capture simulations, schematics, etc. as image files), Image viewer, calculator and text editor, more frequently. There are several menus that have the same applications, just pick one and try launching / killing these applications.

Now, by doing a right click over the desktop, start up a terminal "session". Here we meet a UNIX *shell* at last. This is analogous to popping up a MSDOS prompt in WINDOWS. **Each terminal window/session is a unique environment.** It's almost like you have multiple logins to the computer. What you do in one window does not directly affect any other window, but files created in one, etc., can be seen in others. Each window is running its own shell command interpreter. The "shell" itself is a program that takes commands you type and processes them. There are two popular UNIX shell programs *tsh* and *bash*, we will be using *tsh* by default.

Before going into the 'core' UNIX world, we should learn to logout from the workstation. **Remember that a user should NEVER power on/off a workstation by pressing the buttons on the station.** We should log out for the sake of security and your own privacy. To logout, click on the Actions → Logout. Choose logout from the options window. **Do not lock the screen, shutdown, or restart the computer.**

Last note on Linux. There are many quite well oriented and basic manuals for Linux, some are part of the Red Hat distribution and some are on the web. Feel free to read through them any time and discover more about Linux. This is no big deal but could make you feel comfortable while working.

Now log into the system again, and get ready to try some basic UNIX commands.

2.4 UNIX

An operating system, abbreviated as OS, is a collection of programs that coordinates the operation of hardware and software. UNIX is one kind of an OS, which is basically broken down into three components: Scheduler, file system, and shell. You will only interact with the shell so let's start running some basic UNIX commands now. To execute the shell, click the right button on the desktop.

2.5 Some Basic UNIX Commands

You have previously popped up an x-term (or terminal/console) window, if you don't have one, open it. There should be a line starting with '[you@machine directory]\$', this is the command prompt. This is where you will enter commands. **Note that in the UNIX world we use the word "directory" rather than the Windows world word "folder".**

2.6.1 The ls command

Type `ls` in prompt line, this command displays the contents of the current working directory. For example:

```
[steve@SB15 ~]$ ls
projects cadence sample.txt
```

All commands in UNIX can get arguments. Try `ls -alF`. `-a` is an option that shows all files, even files that are hidden. `-l` is an option type argument, which provides much more information about a file than the simple version. `-F` is an option that marks directories with a trailing slash (/). Example:

```
[steve@SB15 ~]$ ls -alF
drwxr-x---  3 steve facultynis  4096 Oct  4  2004 cadence/
drwxr-xr-x  5 steve facultynis  4096 Aug 21 11:05 projects/
drwxr--r--  5 steve facultynis  4096 Aug 21 11:05 sample.txt
drwxr----- 2 steve facultynis  4096 Feb  3  2003 .acrobat
drwx----- 6 steve facultynis  4096 Aug 19 21:09 .artist_states
drwx----- 2 steve facultynis  4096 Jan 20  2003 .autosave
-rw-r--r--  1 steve facultynis   124 Aug 20 14:36 .bashrc
drwxr-xr-x  3 steve facultynis  4096 Aug 19 20:25 .cadence
drwx----- 2 steve facultynis  4096 Aug 14  2006 .cdpcache
drwx----- 3 steve facultynis  4096 Aug 14  2006 .cdsdoc
-rw----- 1 steve facultynis 43864 Aug 20 09:21 .cdsenv
-rw-r----- 1 steve facultynis   582 Sep  7  2001 .profile
drwx----- 4 steve facultynis  4096 Aug 21 15:41 .gconf
drwx----- 2 steve facultynis  4096 Aug 21 15:48 .gconfd
```

This listing displays several types of information regarding your files. The first entry on the left refers to the permissions on the file, `steve` is owner, `facultynis` is the group, succeeding number is the file size, date, and file names. The first two files are really directories, so they are shown with a “/”. All the rest of the entrees shown appear only by using `-a` option because these are otherwise “hidden files”. The most important thing to see here is that there are hidden files, some of which we will be using later.

2.6.2 The pwd command

Type `pwd`, this command displays your current working directory.

```
[steve@SB15 ~]$ pwd
/homes/faculty/steve
```

Unix has a file system where files are stored on storage devices such as disks, and file directories are organized in a logical and structured fashion. Directories could have files, programs, and subdirectories in them, like in all usual file systems. Every user has his/her own home directory, where no one but the user can access. In your current working directory, you can create other directories, store, or delete files. Note that our space is limited, so be neat in allocating space for storing your studies.

2.6.3 The more and less commands

There are two “pager” programs for looking at text files: `more` and `less`. `More` is more universal, `less` is less so.. but has the advantage of letting you page forward and backward through a file. To run the programs type:

```
[steve@SB15 ~]$ more sample.txt
```

To exit from either, the command is “q” for quit.

2.6.4 The cd command

This command is used to change the working directory. Type `pwd` to see your working directory. Now type `cd sampledir`. Use `pwd` again to see the new working directory.

In general, `cd <directoryname>` is used to change to another directory and `cd ..` to pop out of a directory.

NOTE: you do NOT need the angle brackets around the directory name.

Type `cd ..` (note the space between `cd` and `..`). We are back in our home directory, check typing `pwd`. **If you are in a different directory and you want to come back to your home directory, type `cd` or type `cd ~`.**

```
[steve@SB15 ~]$ cd projects
[steve@SB15 ~/projects]$ pwd
/homes/faculty/steve/projects
[steve@SB15 ~/projects]$ cd CDK
[steve@SB15 CDK]$ pwd
/homes/faculty/steve/projects/CDK
[steve@SB15 CDK]$ ls
ade_viva.log      foo                libManager.log.cdslck  setup_all      test2
casLog           lib.defs          logs_steve             setup_all~
cds.lib          lib.defs~         LVS                   stevelib_06
cds.lib~         lib.defs-s        nands                 stevelib_16
cdsLibEditor.log libManager.log    S_analogLib           test_06

[steve@SB15 CDK]$ cd ..
[steve@SB15 ~/projects]$ pwd
/homes/faculty/steve/projects
[steve@SB15 ~/projects]$ cd ..
[steve@SB15 ~]$ pwd
/homes/faculty/steve
```

2.6.5 The mkdir command

The `mkdir` command is used to create a directory into your current working directory.

```
[steve@SB15 ~]$ mkdir sampledir
```

A directory named “sampledir” is created in your current working directory, in the hierarchical structure.

2.6.6 The cp command

This is used in order to make a copy of a file. Type:

```
[steve@SB15 ~]$ cp sample.txt sample_2.txt
```

The first argument is the name of the file to be copied (source file) and the second argument is the place to copy to (destination file). Type `ls` to see `sample_2.txt` file. Full path names could be used as source and destination files.

2.6.7 The mv command

This is used to move a file between directories and/or renaming a file/directory.

To rename, type:

```
[steve@SB15 ~]$ mv sample_2.txt sample_3.txt
```

Check the results. Now type:

```
[steve@SB15 ~]$ mv sample_3.txt ./sampledir/any_name
```

Switch to `sampledir` (using the `cd` command) and see the file renamed under `sampledir`. You could leave `any_name` place empty, and just carry the file into the directory, leaving the name the same. Now rename the `sampledir` as `enoughdir` by typing:

```
[steve@SB15 sampledir]$ cd ..  
[steve@SB15 ~]$ mv sampledir enoughdir
```

Up to now we have covered choosing our working directory, listing the files in it, creating & copying new files, and making new directories.

2.6.8 The `rm` command

This command is used to remove files. Type `ls` to see `sample.txt` in working directory. Type:

```
[steve@SB15 ~]$ rm sample.txt
```

Now try removing `sampledir`, type:

```
[steve@SB15 ~]$ rm enoughdir  
.....
```

You get an error message!

The way to delete a directory recursively with the files and subdirectories under it in the hierarchical structure is to use `rm` command with `-r` for “recursive” option.

```
[steve@SB15 ~]$ rm -r enoughdir
```

2.6.9 The `chmod` command

This command is used to change the permissions over a file that you own. Type `ls -l` to see the permissions of the files stored in your current directory. The leftmost column indicates whether the file is a file(-), directory(d) or a link(l). 'b' and 'c' can also appear in this position, if the file is a special file used to control a hardware device. This is not our point of interest. The rest of the nine characters in the first strings shows the permissions or mode of the file.

A file can have three sets of permissions: user, group, other, abbreviated as `u,g,o` respectively. Each of these three groups has three parts: read access, write access, and execute access. Read means the subject (`u,g` or `o`) can read the file, write means the subject can edit the file, and execute means execute the file as a command.

For a directory the meanings are slightly different: read access means that the subject is allowed to look at the contents of the directory (with `ls` for example), write access means that the user can create a file in the directory, and execute means the user can go through the directory searching through the subdirectories.

The UNIX system provides the `chmod` command to change the permissions of a file that you own. The syntax for `chmod` is user class (`u,g`, or `o`), followed by the action to take (`-` or `+`), followed by the permission to change (`r,w`, or `x`).

Now create a file and a directory, using `cat` and `mkdir` commands, and see their default permission status by using `ls -la <created_file_name>`.

```
[steve@SB15 ~]$ chmod o+x <created_file_name>
```

Now others have permissions to execute your file after using the above command. You can type `ls -la <created_file_name>` to see the new permissions.

```
[steve@SB15 ~]$ chmod o-rx <created_file_name>
```

The above command will remove the read and execute permissions for others.

2.6.10 The `ps` command

This is used to give information on the running processes. Type:

```
[steve@SB15 ~]$ ps -al
```

In the command arguments, `-a` means all, `-l` denotes long listing. After running the above command, you will see PID (Process ID number) for all the processes as one of the options.

2.6.11 The `kill` command

If a process is stuck, for instance the netscape browser, you can identify that application from the `ps` list and stop that execution by using the `kill` command.

```
[steve@SB15 ~]$ kill <PID of the running application>
```

Don't try to experiment with this command right now, but remember that there is such a command to stop running programs. This is used only for cases where there is no other way to stop the program, since it might leave files in a corrupt state.

2.6.12 The `man` command

This is perhaps the most important command in UNIX. It is the help command for UNIX environment. For example, type:

```
[steve@SB15 ~]$ man ls
```

This formats and displays a page (actually a lot of pages) concerning the usage of `ls` command and its arguments. Try to find out what an `ls -al` command does.

The `man` command provides an extensively useful online tutorial for UNIX. If you are interested, you can take your time to go through some popular commands, but it might take too long. Remember you can also type `man man` to get information on the usage of `man` command.

2.6 Password and Home Directory

After logging in, you can change your password by typing "`yppasswd`" command. You will be required to enter your old password and twice enter the new password.

Your home directory would be:

```
/classes/ece1192/fall106/<username>
```

NOTE: you do NOT need the angle brackets around your username. Type `pwd` to see your home (current directory).

2.7 Snapshots

After completing the assignments, you will definitely need to hand in some simulation results and schematics/layout drawings. It would sometimes be a problem to use CAD tools' built-in print commands, so we could use the snapshot function of Linux. Choose Actions → Take Screenshot or even better use Applications → Graphics → KSnapshot, which allows you to take a portion of the screen.

2.8 Print outs

The built in 'Print' commands of many applications may fail, as they would individually require printer path definitions or captures. To take a printout from Linux workstations, prepare your file and save it with a predefined, known suffix, like `.ps`, `.doc`, `.txt`, etc. Then use the terminal window and `lpr` command to take printouts. For example to print the `sample.txt` file, type:

```
[steve@SB15 ~]$ lp sample.txt
```

This will send the request (`sample.txt` file) to the printer across the room.

When you complete your work, don't forget to logout.

3 Class Account Setup

Before you are able to use the tools which we will use in class, you must first setup your account. This can be done by 'sourcing' a configuration file. To do this, type the following command in the Terminal:

```
[steve@SB15 ~]$ source ../CLASS/dot_cshrc
```

The `source` command reads through the specified file and sets variables and other options which you will need. This must be done every time you login. However, you can modify your `.cshrc` file so that it is no longer necessary to execute the command each time you login. To do this, open your `.cshrc` file in your favorite text editor (if you don't have a favorite text editor, `dtpad` will work). Add the following line to your file, then save it:

```
source ../CLASS/dot_cshrc
```

Note: you will need to open another shell window to see your changes take effect.

Q: Why?

To Turn In: Printout of your new `.cshrc` file with your name on it.