

CSE 390, Spring 2010

Assignment 1: Basic Unix Shell Commands

Due Tuesday, October 12, 2010, 12:30 PM

This assignment focuses on using the `bash` shell to execute common Unix commands. You will create and electronically turn in a file named `homework1.txt` that contains your answers to several questions below. Some of the questions are Unix commands you must figure out, and others are general questions about the Linux box you will be using.

Note: Unless otherwise specified, the answers to each question in this assignment can be found entirely using commands shown in the lecture notes from the first week. You may use other commands if you like, but you should constrain yourself to those from lecture or from the *Linux Pocket Guide* textbook. Ask the instructor/TA if you are unsure whether a particular command is allowed.

Task 1: Log in to a Linux machine

First, **log in to a machine running Linux**, such as one of the Linux computers in the CSE basement labs. If you prefer, you can instead use your own machine running Linux (such as a "virtual box" as described on the Work @ Home page on the course web site). Or if you know how to remotely connect to the CSE department's `attu` Linux server using the `ssh` program, you may do that. (We will teach you how to do that in a future lecture.) If you aren't able to successfully log in to a Linux machine, please contact us for help or ask a classmate.

Now, launch a **Terminal** window and a **text editor** from the Linux user interface, generally from the top left drop-down application launcher menu. You can usually find the terminal program under System Tools or Accessories.

Task 2: Prepare your home directory

We have set up a ZIP archive full of support files that you must download to your Linux machine. Do the following:

- Create a directory inside your home directory named `390`.
- Create a sub-directory inside `390` named `hw1`.
- Download our support file `hw1.zip` and save it into your new `hw1` directory. You can do this in one of two ways:
 1. By loading a web browser on the Linux box, browsing to our course web site, clicking the Homework link, finding the link to `hw1.zip`, right-clicking it, choosing Save Link Target As..., and browsing to the `hw1` folder;
 2. Or, by typing the following command into your terminal window, when the current directory is `hw1` :
 - `wget http://www.cs.washington.edu/education/courses/cse390a/10au/homework/1/hw1.zip`
- Unzip the `hw1.zip` file's contents into your `hw1` folder. You can do this in one of two ways:
 1. By running a file browser (usually found in the "Places" top menu) and browsing to the `hw1` folder, then double-clicking on the `hw1.zip` file, and using the graphical unzipping program to extract the files;
 2. Or, by typing the following command into your terminal window, when the current directory is `hw1` :
 - `unzip hw1.zip`(We are trying to persuade you that doing things in a terminal can sometimes be the easier way!)

If you did everything correctly, you should now have several files and directories within your `hw1` directory, such as `java/`, `website/`, `animals.txt`, `Burrot.java`, `numbers.txt`, and `verse1.txt`.

(continued on next page)

Task 3: Learn more about the Linux computer

The following are questions about your Linux system for you to investigate and discover the answers. You should create a text file named `homework1.txt` and save it somewhere in your home directory, such as in your 390 folder. **In this file, write your answers to the questions below.** Your file should contain **5** answers total since there are 5 problems below.

Most of the answers to the questions below can be found entirely using commands that you were shown in the week 1 slides. Any question that requires additional commands will have a label of "Self-Discovery". Remember that you can learn more about any command by typing:

```
man command
```

Each Linux system is different, so there is not one "correct" answer to every question. You will receive credit if your answer is plausible for a Linux system in general.

1. Try running each of the following commands on your Linux system, one at a time. (You don't need to turn in the output of the commands.) What is the last command doing? Describe what it does in one sentence. Remember that you can read more about a command and its parameters by looking at its `man` page.

```
ls
```

```
ls -m
```

```
ls -1          (in this case, the -1 is the number 1, not a lowercase L)
```

2. What is the full path of your home directory on this Linux box? Run an appropriate command to find out.
(Note: You can select text on the Terminal window and copy/paste it into your text editor. The terminal program usually has a top menu bar where you can choose Edit → Copy and Edit → Paste.)
3. What is the exact current date and time, as reported by this Linux box? Run a command to find out.
4. What other users have home directories on this Linux box? (Hint: Find out what other user directories exist within the same parent folder as your home directory.) Provide a list of all such users in the format produced by the `ls` command. If there are more than 5 users, listing the last 5 is sufficient.
5. *(Self-Discovery)* What version of Linux does this computer use?

To find out, run the Linux `uname` command, which outputs various information about the current system. It has various parameters that you can learn in its `man` page.

You are looking for the version of the kernel (the core) of the Linux OS, not the name of the "distribution" such as Fedora or Ubuntu. *(Hint: The answer begins with 2.6.)*

(continued on next page)

Task 4: Linux Bash shell commands

For each of the numbered items below, **determine a single bash shell statement that will perform the operation(s) requested**. Each of your solutions must be a single one-line shell statement and should not use Linux's multi-statement joining operators such as `|`, `&&`, `||`, and `;`. (We haven't learned these yet, anyway.)

Write these commands into your `homework1.txt` file. In other words, after completing this task, your `homework1.txt` file should now have **18** answers, including the 5 answers from Task 3. In your file, write the command that will perform the task described for each numbered item; don't write the output that the command produces.

To test your commands, you should have unzipped `hw1.zip` into the current directory. Most of the questions below entirely use commands shown in the week 1 slides. Several questions require you to learn new parameters to those commands; find these out by looking at `man` pages or the *Linux Pocket Guide*.

1. List all files in the current directory, in "long listing format".
2. List all files in the `/var` directory, in reverse alphabetical order.
3. Copy the file `numbers.txt` from the current directory to the `java` subdirectory.
4. Rename the file `Burrot.java` to `Borat.java`. (*Hint: Renaming is done using the same command as moving.*)
5. Delete the files `diff.html` and `diff.css`. (*Hint: Many commands can accept more than one parameter.*) Note that this must be done with a single command and not multiple commands.
6. Print a text calendar for the month of June, 2049.
7. Set the file `MyProgram.java` to have a last-modified date of March 15, 4:56pm.
(*Hint: the man page for the proper command describes the timestamp 'STAMP' format to use. Look for this!*)
(Also note that Linux is case-sensitive when you are specifying file or directory names.)
8. (*Self-Discovery*)
You can use a `*` (asterisk) as a "wild-card" character to specify a group of files. For example, `*foo` means all files whose names end with `foo`, and `foo*` means all files whose names begin with `foo`. You can use a wildcard in the middle of a file name, such as `foo*bar` for all files that start with `foo` and end with `bar`.

List all web page files (files whose names end with the extension `.html` or `.css`) in the current directory. Note that the `ls` command can accept parameter(s) for what files you want it to list.
9. Copy all the text files (files whose names end with `.txt`) from the current folder to the `website` subdirectory.
10. (*Self-Discovery*)
The `cat` command outputs the contents of a file to the terminal.
The `less` command outputs the contents of a file to the terminal, page by page, pausing for you to press a key.

Display the contents of the file `MyProgram.java`.
11. Display the contents of all files whose names begin with `verse` and end with the extension `.txt`, such as `verse1.txt` and `verse2.txt`. (*Write a single command that displays all their contents concatenated.*)
12. (*Self-Discovery*)
The `head` and `tail` commands output only the first or last few lines (respectively) of a file to the terminal.

Display the last 5 lines of the file `animals.txt` from the current directory on the terminal.
13. (*Self-Discovery*)
The `wc` command outputs how many bytes, words, lines, etc. a file occupies.
Display the number of lines occupied by the file `animals.txt`.