System Administration for Beginners

Week 2 Laboratory

February 9, 2009

1 Getting Started

There are several ways to get laboratories done; you must either be at the computing facilities at Soda Hall or remotely connected to the computers in Soda Hall using your computer at home or at other laboratories like the Open Computing Facility. For more information on how to remotely connect to the computers in Soda Hall, see the following:

- `http://inst.eecs.berkeley.edu/connecting.html#network`
- `http://www.ocf.berkeley.edu/~eleen/web/ssh.html`
- `http://docs.ocf.berkeley.edu/wiki/SSH_Secure_Shell`

The second and third URLs were designed as a tutorial on how to remotely connect to the Open Computing Facility (OCF). Most of the steps remain the same, regardless of where you are connecting to. However, you will want to change the hostname like `solar.cs.berkeley.edu` instead of `ocf.berkeley.edu`.

Note that in order to connect to the EECS instructional servers, you will need to use your `inst` account that was assigned to you in class (in the format of `CS198-XX`, where `XX` is a two-letter combination). To connect to the servers at the OCF, you need to use your `ocf` account, which you signed up for as part of homework #1.

2 Submission Guidelines

At the top of each submission, please provide the assignment name, your full name, `inst` login (`cs198-XX`), and your email address. Answer the following questions below. If no specific information is being asked, include any output or answers that you think would help show us that you understand the material (text only). Turn in your *paper* submission at the start of class next week.

3 Laboratory

3.1 Terminal & Shell

The terminal is a tool for entering data into and displaying data from a computer. For now, its main purpose will be to feed your commands to the shell. There are multiple terminals available – older terminals featured a monitor and keyboard that would allow you to interact with a mainframe that was not necessarily in front of you. Current terminals are computer programs (there are terminals developed for almost any operating system) that provide access to other programs. There are different terminals with many features, but they practically work all in the same way.

We won't be using the GUI in this course, not if we can help it at least. A system administrator is most likely to work on the server remotely, and even when they aren't working remotely they will end up using a simple command line. A text-based interface is the most efficient way of working remotely, and also it provides to be less of a distraction when a system administrator is working. In the world of UNIX anything that you want to do can be done on the command line.

The terminal allows us to access the computer, but requires another program to process commands. The shell is a computer program that takes your commands, sends it to the operating system which it will then process and execute. Like terminals, shells come in different variations and feature sets. The two most popular shells are `tsch` and `bash`. `Tcsh` is the default shell on Solaris and BSD systems while bash is the default shell for Linux. You are welcome to choose which shell to use; you can always run a different shell in a shell by typing in the same of the shell you would like to use.

3.2 Basic Commands

NOTE: The commands you will be learning today deal with the Solaris version. There is a slightly different form of syntax in commands between Solaris and Linux, so if any command you use does not work as expected (regardless of which version you are working with), check the documentation. In the end, knowing the syntax of both versions is to your advantage as you will be able to work on either system, a skill-set that is in high demand in mixed-environments.

3.2.1 Documentation and Help

Commands: man, apropos, info

Google as a way of life - online documentation - mailing lists - books

3.2.2 File and Directory Manipulation

Commands: ls, cd, chmod, chown, chgrp, mkdir, rm, rmdir, touch, file, du, cp, pwd, less, more, cat

the UNIX file system - case-sensitivity - lack of file extensions - tab-completion - directories and devices as files - owners and permissions

3.2.3 Process Control

Commands: ps, kill, pkill

process ID (PID) - process owner - signals

3.2.4 Text Editors

Commands: emacs, vim, pico, nano

the plaintext file

3.2.5 Network Utilities

Commands: wget, links

the command-line interface

3.3 Extra Resources

- See the 'Resources' link on the course homepage
- `http://docs.linux.com/search.pl?tid=89` - Some fun guides to familiarize yourself with the command line (scroll to the bottom and start with the first article).

3.4 Questions

1. Describe the process of logging into the Solaris systems at Soda Hall. Suppose you just sat down at a terminal. How would you login? How would you open a web browser and a terminal?
2. What is the command to determine where you are in the UNIX filesystem?
3. What are the commands to make a new file? Do you have to make a new file before editing it?

4. Create a new file. How would you copy the file to a different location? How would you copy multiple files at the same time?

5. Download a text file from the Internet. **HINT:** Use `wget` to get some webpage. What commands could you use to display the contents of the file? **EXTRA:** That file may be very long; is there a command that will allow you to "scroll" through the text?

6. What are the commands to make a new directory? If you wanted a directory hierarchy of `foo/bar`, where `bar` is a sub-directory of `foo`, and `foo` does not exist, do you have to create `foo` first? Read the *man* page for the command to make a new directory and see if there is an option for automatically creating parent directories.

7. Make a new directory and create a few files in that directory. How do you list all the files in that directory? How would you list the files so that all their attributes are displayed (e.g., owner, size, creation time)?

8. Using only one command, how would you delete all the files in the directory created in the previous step?

9. Make another directory and create a few files in that directory. Try deleting the directory. What happens? Can you figure out a method for deleting a non-empty directory?

10. How would you rename a file? **HINT:** There is no 'rename' command. There's an easy and a hard way to rename a file; try looking for both.

11. How would you change the permissions on a file? Describe the three categories of permissions, and the three categories of users to whom these permissions apply. **EXTRA:** How would you change file ownership?

12. Can a file and a directory have the same name? Are file and directory names case-sensitive?

13. Start a web-browser process. Can you figure out the process name and process ID for the web-browser? **HINT:** It probably contains the word "mozilla" or "firefox".

14. How would you kill the process in the previous step? What if you didn't know the process ID?