7 Linux

Objective

In this lab , the Linux operating system will be explored. Linux is a lesser know operating system, compared to Windows and MacOS, however it is widely used and responsible for hosting much of the internet today. If you use an Android phone, it's actually using Linux under the hood. This lab will introduce the key concepts of the Linux desktop environment and the command line interface.

Theory

The Linux operating systems are built on the Linux kernel, which is a direct interface to low-level compute resource such as the CPU, memory, and other peripherals. On top of the kernel, there are a number of software packages that are typically shipped in collections called distributions, or distros for short. There are many different Linux distros, typically tailored to the end-users application. In the lab we will explore one of the most popular ditros, Ubuntu. A great resource to learn more about other distributions is `distrowatch.com/`.

The Linux file system is organized in a hierarchical structure, with all parent directories originating from root or /. Each user has their own directory for working, typically /home/username. Users are able to create and manipulate files within their home directory freely, whereas system files are owned by the root user and cannot be modified by regular users. The root user, also know as the superuser, can be thought of as the admin account on the system. The root user has the ability to modify any file on the system for administrative configuration, however they also have the ability to destroy the system. In our computer lab, we will only have access to regular user accounts.

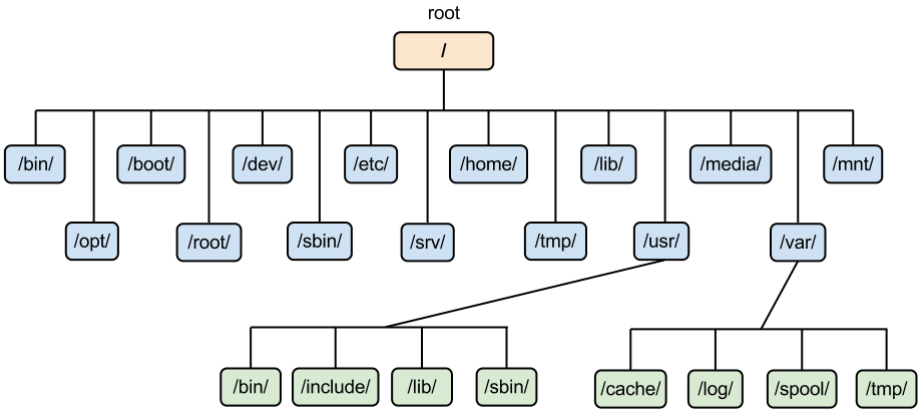

Figure 13: Linux file hierarchy, source: `https://iq.opengenus.org/content/images/2022/01/tree-1.png`

The most common method for interfacing the Linux OS is through a terminal. The terminal is a piece of software that allows a user to issue commands to a shell. The shell interprets the commands and performs the necessary actions with the OS to execute the command. In this lab, we will explore the bourne-again shell, or BASH, which is a commonly used shell often configured as default on Linux systems. Keep in mind that there are many other shells that can be used.

Equipment

- Ubuntu computer

Procedure

Table 2 presents a list of some of the most commonly used commands in the BASH terminal.

Command	Description
pwd	Prints the present working directory
ls	Lists the contents of directory
cd	Change working directory
mkdir	Make a directory
rmdir	Remove a directory
touch	Create or update timestamp of file
rm	Remove a file
mv	Move a file
cp	Copy a file
cat	Print contents of a file
man	Opens manual entry for a command
echo	Prints text to terminal
wc	Prints newline, word, and byte count for a file
history	Prints the entire bash command history
nano	command-line text editor
vi	command-line programmer's text editor
ssh	secure shell remote login client
chmod	Change permissions on a file

Table 2: Common BASH commands

7.1 Terminal navigation and file manipulation

In this section, we will practice in the terminal: navigating the file system, creating files, moving files, and removing files.

Note: commands are presented after the $ symbol below

1. Open a terminal with Ctrl+Alt+T, or select "Terminal" from the Activities menu.
2. Print the path to your present working directory: $ pwd
3. List the contents of your present working directory: $ ls
4. Create a new directory called lab: $ mkdir lab

5. List again, notice the new directory in the listing: $ ls
6. Change your present working directory to the new lab directory: $ cd lab
7. Check the path of this present working directory: $ pwd
8. Create an empty file called test: $ touch test
9. List the contents of lab/, and notice the new file test: $ ls
10. Copy this file to a new name in the same directory: $ cp test test2
11. Use echo to add text to the new file, test2: $ echo "Hello, World!" > test2
12. Print the contents of test2: $ cat test2
13. Print a long listing to display more information about the files in the lab/ directory: $ ls -la
14. Create another directory in lab/ called subdir/: $ mkdir subdir
15. Move test2 into subdir/: $ mv test2 subdir/
16. List your present working directory and notice that test2 is missing: $ ls
17. List the contents of subdir/, note that we can stay in the lab/ directory for this: $ ls subdir/
18. Change directory to the subdir/ directory: $ cd subdir
19. List contents again, use the long listing switch: $ ls -la
20. Navigate back one directory using the built-in links(single-dot corresponds to current directory, double-dot corresponds to the parent directory): $ cd ..
21. Attempt to remove the subdir/ directory: $ rmdir subdir/
22. Remove the file in subdir/: $ rm subdir/test2
23. Attempt to remove subdir/ again: $ rmdir subdir/

7.2 Text editing with nano

In this section, we will practice editing files with the terminal text editor called nano.

1. Navigate back to the lab/ directory.
2. Display all the commands you have ran using $ history
3. Write the history to the test file: $ history > test
4. Long list your current directory: $ ls -la
5. Print the newline, word, and byte count of test: $ wc test
6. Use the nano text editor to modify test: $ nano test

7. In nano, we can see all of our command history exists in the file.
    - Navigate the page using the arrow keys, Home key, and End key.
    - Move to the bottom of the file, add a new line and input *some* text.
8. Save the file in nano with Ctrl+O, do not change the file name, then press Enter. Close nano with Ctrl+x.
9. Count the lines, words, and bytes for test again: $ wc test
10. Display the contents of test with the different commands:
    - $ cat test
    - $ less test
    - $ more test
11. Exit the terminal session to save your bash history with Ctrl+D or type: $ exit

7.3 Connecting to remote servers

As mentioned in the objective, Linux plays a major role in hosting servers on the internet. Many Linux systems are configured to run a service called secure shell (ssh). Secure shell is a means of connecting to a shell on a remote system over a network connection.

In this section, we will use ssh to connect to Mr. Wright's remote server at `justnotes.dynu.net`. Your username will be the first part of your SUNY Poly email. (i.e. email: boned@sunypoly.edu, username: boned)

1. Connect to the remote server, replace 'user' with your username: $ ssh -p 31222 user@justnotes.dynu.net

Questions

1. What is the path, with respect to root (/), to your home directory on the SUNY Poly computers? What about on the justnotes.dynu.net server?
2. Consider the following series of commands. Will the 'sampledir/' directory be successfully deleted with the rmdir command? Why or why not?

```
boned@DESKTOP-R20V1I2:~/lab$ pwd
/home/boned/lab
boned@DESKTOP-R20V1I2:~/lab$ ls
sampledir
boned@DESKTOP-R20V1I2:~/lab$ ls sampledir/
file
boned@DESKTOP-R20V1I2:~/lab$ rmdir sampledir/
```

3. The commands: cat, less, and more, all print text to the terminal. What are the differences between them? (hint: use 'man' to access the manual entries for each command).