

RH124 Report

Ruojia Zhang

## Module summary

### **Module1**

This is the first module of the training, and it briefly introduced what is Linux, Advantages of Linux, why should I learn Linux, what is Open-Source Software, Types of Open-Source Licenses, Who Develops Open-Source Software, who is Red Hat, what is a Linux distribution, And Red Hat Enterprise Linux to me. I have got the basic idea of what Linux is and have understand what I should do to start learning to use Linux. Also, I get to know several different Linux distributions like Fedora, Red Hat Enterprise Linux, and CentOS. Finally, I learnt that Red Hat is commercially-supporting community for their projects.

I've learnt: Open-source software is software with source code that anyone can freely use, study, modify, and share. A Linux distribution is an installable operating system constructed from a Linux kernel and supporting user programs and libraries. Red Hat participates in supporting and contributing code to open-source projects, sponsors and integrates project software into community-driven distributions and stabilizes the software to offer it as supported enterprise-ready products. Red Hat Enterprise Linux is Red Hat's open source, enterprise-ready, commercially supported Linux distribution.

### **Module2**

This is the second module of the training. It mainly discussed the Fundamentals of Linux System and how to run commands using the shell, Including Log in to a Linux system on a local text console and run simple commands using the shell. Log in to a Linux system using the GNOME 3 desktop environment and run commands from a shell prompt in a terminal program.

Save time by using tab completion, command history, and command editing shortcuts to run commands in the Bash shell.

In this chapter, I learned: The Bash shell is a command interpreter that prompts interactive users to specify Linux commands. Many commands have a `--help` option that displays a usage message or screen. Using workspaces makes it easier to organize multiple application windows. The Activities button located at the upper-left corner of the top bar provides an overview mode that helps a user organize windows and start applications. The `file` command scans the beginning of a file's contents and displays what type it is. The `head` and `tail` commands display the beginning and end of a file, respectively. You can use Tab completion to complete file names when typing them as arguments to commands.

## LAB1

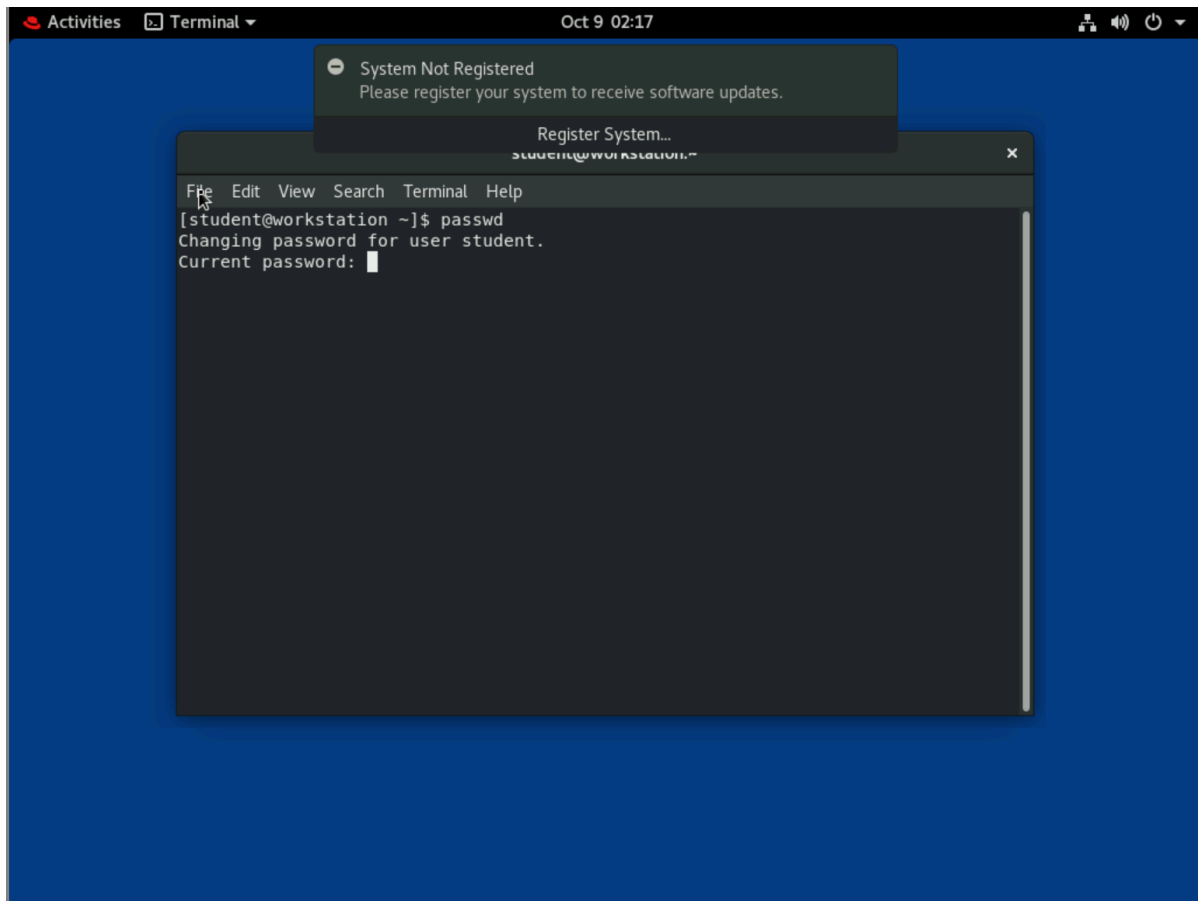


Figure 1. Work station is successfully setup and is able to change password using the passwd operation.

```
02:25:24 AM
[student@workstation ~]$ lab cli-review grade

Grading the student's work on workstation:

· You completed attempting these tasks..... PASS
Overall lab grade..... PASS

[student@workstation ~]$ lab cli-review finish

Cleaning up the lab on workstation:

· Removing the files created..... SUCCESS
Lab finished.

[student@workstation ~]$
```

*Figure 2.* Running simple commands on the terminal shell and pass task grading.

### **Module3**

In this third module of the training, the narrator mainly discussed Basic command operations on Bash shells, including Describe how Linux organizes files, and the purposes of various directories in the file-system hierarchy. Specify the location of files relative to the current working directory and by absolute location, determine and change your working directory, and list the contents of directories. Create, copy, move, and remove files and directories. Make multiple file names reference the same file using hard links and symbolic (or "soft") links. Efficiently run commands affecting many files by using pattern matching features of the Bash shell.

In this chapter, I learned: Files on a Linux system are organized into a single inverted tree of directories, known as a file-system hierarchy. Absolute paths start with a / and specify the location of a file in the file-system hierarchy. Relative paths do not start with a / and specify the location of a file relative to the current working directory. Five key commands are used to manage files: mkdir, rmdir, cp, mv, and rm. Hard links and soft links are different ways to have multiple file names point to the same data. The Bash shell provides pattern matching, expansion, and substitution features to help you efficiently run commands.

## LAB2

```
[student@workstation ~]$ lab files-review grade

Grading the student's work on serverb:

· Checking files in chapters directory..... PASS
· Checking files in editor directory..... PASS
· Checking files in season1 directory..... PASS
· Checking files in season2 directory..... PASS
· Checking hard link..... PASS

Overall lab grade..... PASS

[student@workstation ~]$ lab files-review finish

You have completed attempting the lab exercise.

· Checking ..... SUCCESS
· Removing lab content from serverb..... SUCCESS

Lab finished.

[student@workstation ~]$
```

Figure 3. The success grading of file operations.

## Module4

In this forth module of the training, in this chapter, I learned: Man pages are viewed with the man command and provide information on components of a Linux system, such as files, commands, and functions. By convention, when referring to a man page the name of a page is followed by its section number in parentheses. Info documents are viewed with the pinfo command and are made up of a collection of hypertext nodes, providing information about software packages as a whole. The navigational keystrokes used by man and pinfo are slightly different.

## LAB3

```

student@workstation:~
File Edit View Search Terminal Help
[student@workstation ~]$ firefox /usr/share/doc
[student@workstation ~]$ lab help-review grade

  · Ensuring /home/student/passwd.ps exists: ..... PASS

Overall lab grade..... PASS
[student@workstation ~]$ lab help-review finish

Completing the lab on workstation:
  · Remove /home/student/passwd.ps: ..... SUCCESS

Lab finished.

[student@workstation ~]$ █

```

Figure 4. The success grading of help operations.

## Module5

In this module 5, I'm learning about Creating, viewing, and editing text files and learn about Save command output or errors to a file with shell redirection, and process command output through multiple command-line programs with pipes. Create and edit text files using the vim editor. Use shell variables to help run commands, and edit Bash startup scripts to set shell and environment variables to modify the behavior of the shell and programs run from the shell.

In this chapter, I learned: Running programs, or processes, have three standard communication channels, standard input, standard output, and standard error. You can use I/O redirection to read standard input from a file or write the output or errors from a process to a file.

Pipelines can be used to connect standard output from one process to standard input of another process, and can be used to format output or build complex commands. You should know how to use at least one command-line text editor, and Vim is generally installed. Shell variables can help you run commands and are unique to a particular shell session. Environment variables can help you configure the behavior of the shell or the processes it starts.

### Lab4

```
lab_review.txt
[student@workstation ~]$ lab edit-review grade

Grading the student's work on workstation:

· Verifying that the file exists: ..... PASS
· Verifying that the first 3 lines are removed: ..... PASS
· Verifying permissions column is removed: ..... PASS
· Verifying group column is removed: ..... PASS
· Verifying time column is removed: ..... PASS
· Verifying Desktop and Public lines are removed: ..... PASS
· Verifying dashed line is added: ..... PASS
· Verifying directory listing added: ..... PASS

Overall lab grade..... PASS
```

Figure 5. Grading after all file text operations.



```

cat: editing_final_lab: No such file or directory
[student@workstation ~]$ cat editing_final_lab.txt
-rw-  1 student    49 Oct  9  .bash_history
-rw-  1 student    18 Aug 30  .bash_logout
-rw-  1 student   141 Aug 30  .bash_profile
-rw-  1 student   312 Aug 30  .bashrc
drwx 11 student   253 Oct  9  .cache
drwx 10 student   274 Nov  9  .config
drwx  2 student    28 Oct  9  Documents
drwx  2 student     6 Sep  1  Downloads
-rw-  1 student     0 Oct  9  editing_final_lab.txt
-rw-  1 student    16 Sep  1  .esd_auth
-rw-  1 student  1550 Oct  9  .ICEauthority
drwx  3 student    19 Sep  1  .local
drwx  5 student    66 Oct  9  .mozilla
drwx  2 student     6 Sep  1  Music
drwx  2 student     6 Sep  1  Pictures
drwx  3 student    19 Sep  1  .pki
drwx  2 student    73 Sep  1  .ssh
drwx  2 student     6 Sep  1  Templates
drwx  2 student     6 Sep  1  Videos
-rw-  1 student  1402 Nov  9  .viminfo
-----
lab_review.txt

```

Figure 6. Output of the edited file

## Module6

In the module6, which is mainly about Managing Local Users and Groups. Including Describe the purpose of users and groups on a Linux system. Switch to the superuser account to manage a Linux system, and grant other users superuser access using the sudo command. Create, modify, and delete locally defined user accounts. Create, modify, and delete locally defined group accounts. Set a password management policy for users, and manually lock and unlock user accounts.

In this chapter, I learned: There are three main types of user account: the superuser, system users, and regular users. A user must have a primary group and may be a member of one or more supplementary groups. The three critical files containing user and group information are

/etc/passwd, /etc/group, and /etc/shadow. The su and sudo commands can be used to run commands as the superuser. The useradd, usermod, and userdel commands can be used to manage users. The groupadd, groupmod, and groupdel commands can be used to manage groups. The chage command can be used to configure and view password expiration settings for users.

### Lab5

```
logout
Connection to serverb closed.
[student@workstation ~]$ lab users-review grade

Grading the student's work on serverb:

· The group consultants with GID 35000 exists in serverb..... PASS
· Verifying the group membership of consultants on serverb.... PASS
· Verifying the sudo access of consultants on serverb..... PASS
· Verifying password expiry of consultants on serverb..... PASS
· Verifying account expiry of consultants on serverb..... PASS
· Verifying default password expiry of users on serverb..... PASS
· Verifying password change date of consultants on serverb.... PASS

Overall lab grade..... PASS

[student@workstation ~]$ █
```

Figure 7. The output of grading of user group operations

```
[student@serverb ~]$ sudo groupadd -g 35000 consultants
[student@serverb ~]$ sudo vim /etc/sudoers.d/consultants
[student@serverb ~]$ sudo useradd -G consultants consultant1
[student@serverb ~]$ sudo useradd -G consultants consultant2
[student@serverb ~]$ sudo useradd -G consultants consultant3
[student@serverb ~]$ date -d "+90 days" +%F
2023-01-07
[student@serverb ~]$ sudo chage -E 2023-01-07 consultant1
[student@serverb ~]$ sudo chage -E 2023-01-07 consultant2
[student@serverb ~]$ sudo chage -E 2023-01-07 consultant3
[student@serverb ~]$ sudo chage -M 15 consultant2
[student@serverb ~]$ sudo chage -d 0 consultant1
[student@serverb ~]$ sudo chage -d 0 consultant2
[student@serverb ~]$ sudo chage -d 0 consultant3
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$ lab users-review grade

Grading the student's work on serverb:
```

Figure 8. Operations done during the user group operations