**Chapter 03 Lab: Managing Files from the Command Line**

Marko Shaffer

Information Technology, Franklin University

ITEC 200: Linux Fundamentals

Professor Kagan Ulucay

5/28/2023

Red Hat System Administration I 8.2

Chapter 03 Lab: Managing Files from the Command Line

**Performance Checklist:**

In this lab, you will efficiently create, move, and remove files and directories by using the shell and a variety of file name matching techniques.

**Outcomes:**

You should be able to:

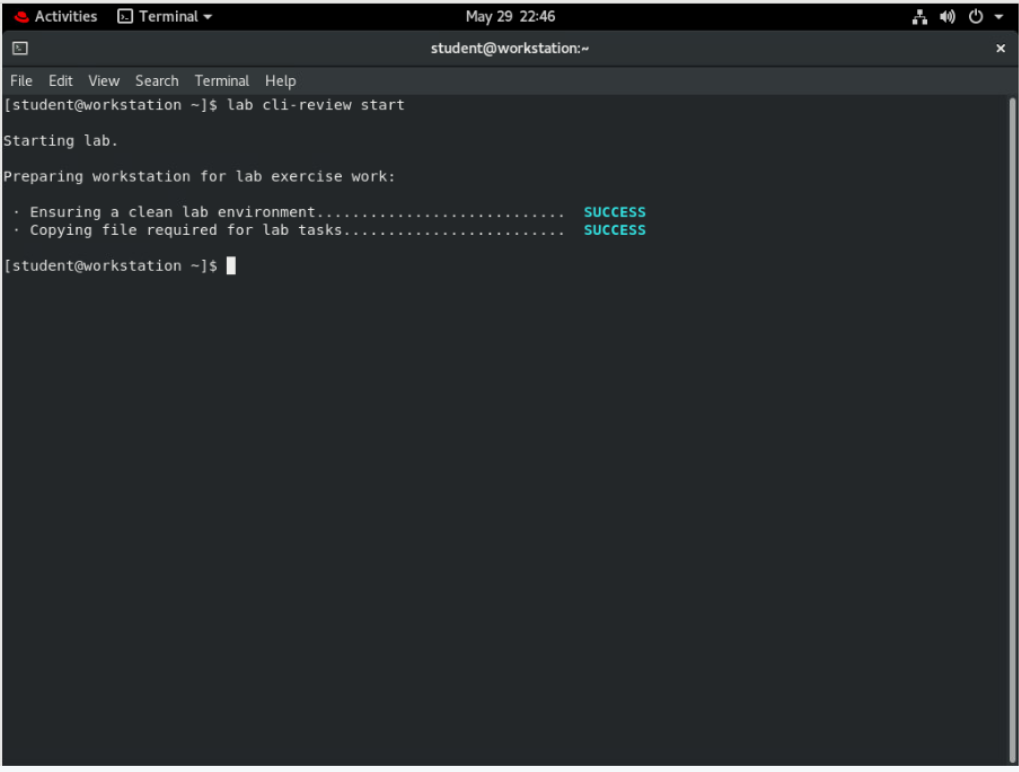
* Use wildcards to locate and manipulate files.

Log in to workstation as student using student as the password.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Franklin VM: | Standard User Account: | The Student's Root Account: |
| Username | kiosk | student | root |
| Password | redhat | student | redhat |

On workstation, run the **lab files-review start** command. The command runs a start script that determines if the serverb machine is reachable on the network.

**[student@workstation ~]$ lab files-review start**

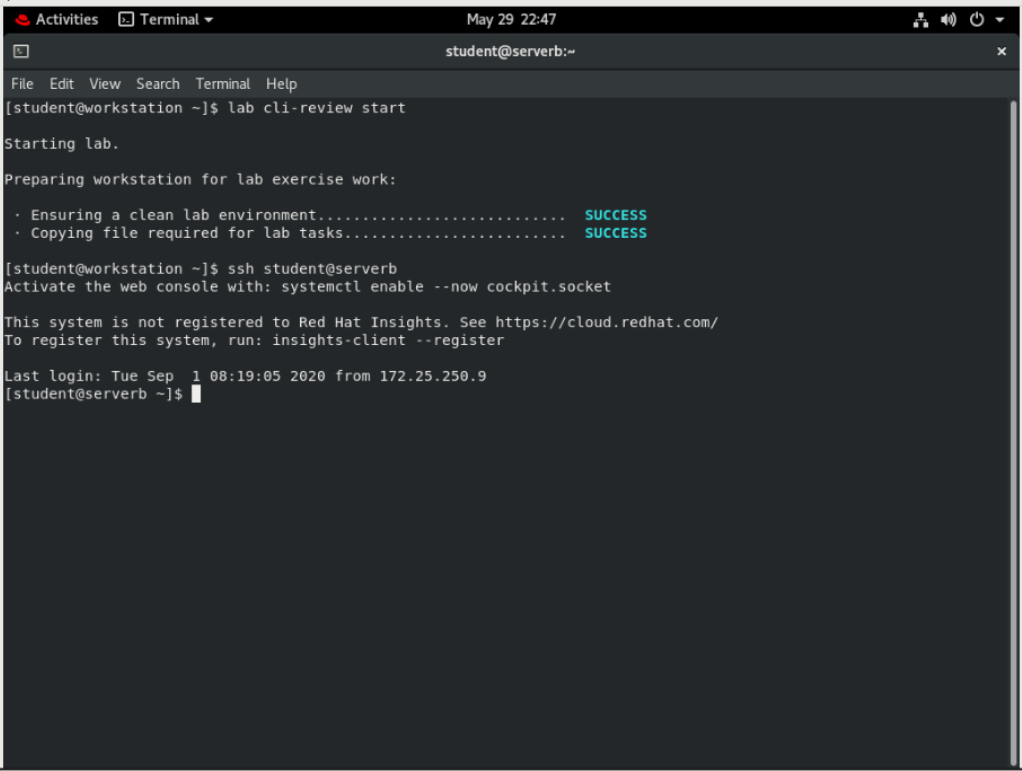


1. Use the **ssh** command to log in to serverb as the student user. The systems are configured to use SSH keys for authentication, and therefore a password is not required.

**[student@workstation ~]$ ssh student@serverb**

*...output omitted...*

**[student@serverb ~]$**



1. Before you create project files, use the **mkdir** command with brace expansion to create empty project planning documents in the /home/student/Documents/project\_plans directory. (Hint: if ~/Documents does not exist, the -p option for the **mkdir** command will create it.)

Create two empty files in the ~/Documents/project\_plans directory: season1\_project\_plan.odf and season2\_project\_plan.odf.

**[student@serverb ~]$ mkdir -p ~/Documents/project\_plans**

**[student@serverb ~]$ touch \**

**~/Documents/project\_plans/{season1,season2}\_project\_plan.odf**

**[student@serverb ~]$ ls -lR Documents/**

Documents/:

total 0

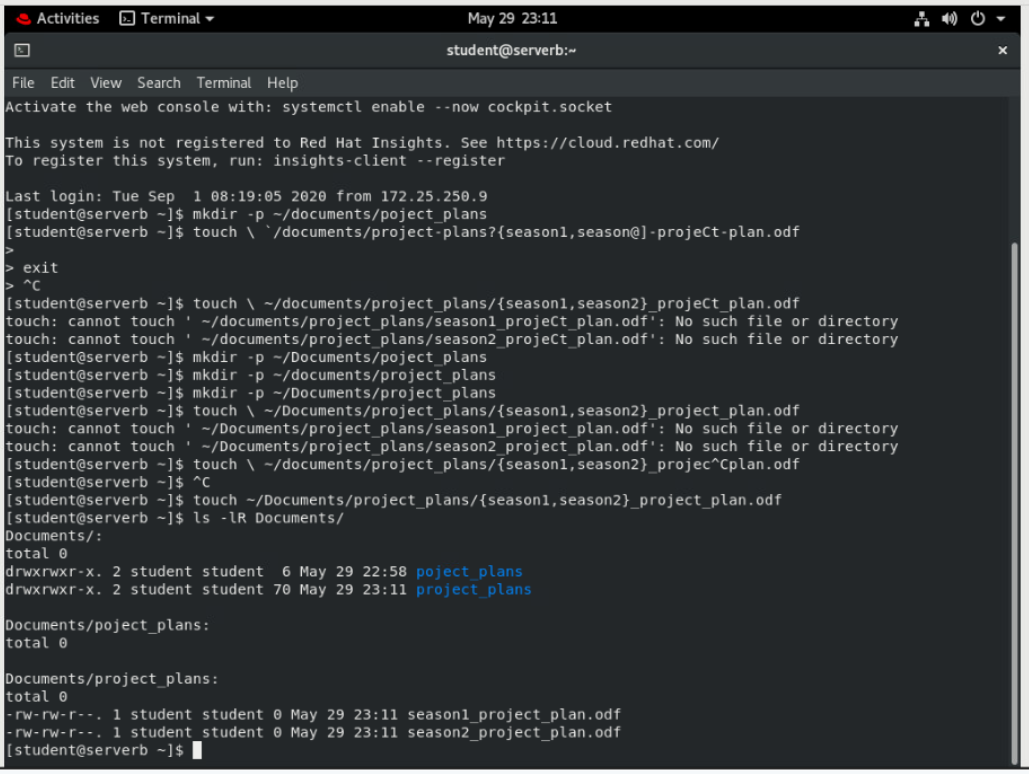
drwxrwxr-x. 2 student student 70 Jan 31 18:20 project\_plans

Documents/project\_plans:

total 0

-rw-rw-r--. 1 student student 0 Jan 31 18:20 season1\_project\_plan.odf

-rw-rw-r--. 1 student student 0 Jan 31 18:20 season2\_project\_plan.odf



1. Create sets of empty practice files to use in this lab. If you do not immediately recognize the intended shell expansion shortcut, use the solution to learn and practice. Use shell tab completion to locate file path names easily.

Create a total of 12 files with names tv\_seasonX\_episodeY.ogg. Replace *X* with the season number and *Y* with that season's episode, for two seasons of six episodes each.

**[student@serverb ~]$ touch tv\_season{1..2}\_episode{1..6}.ogg**

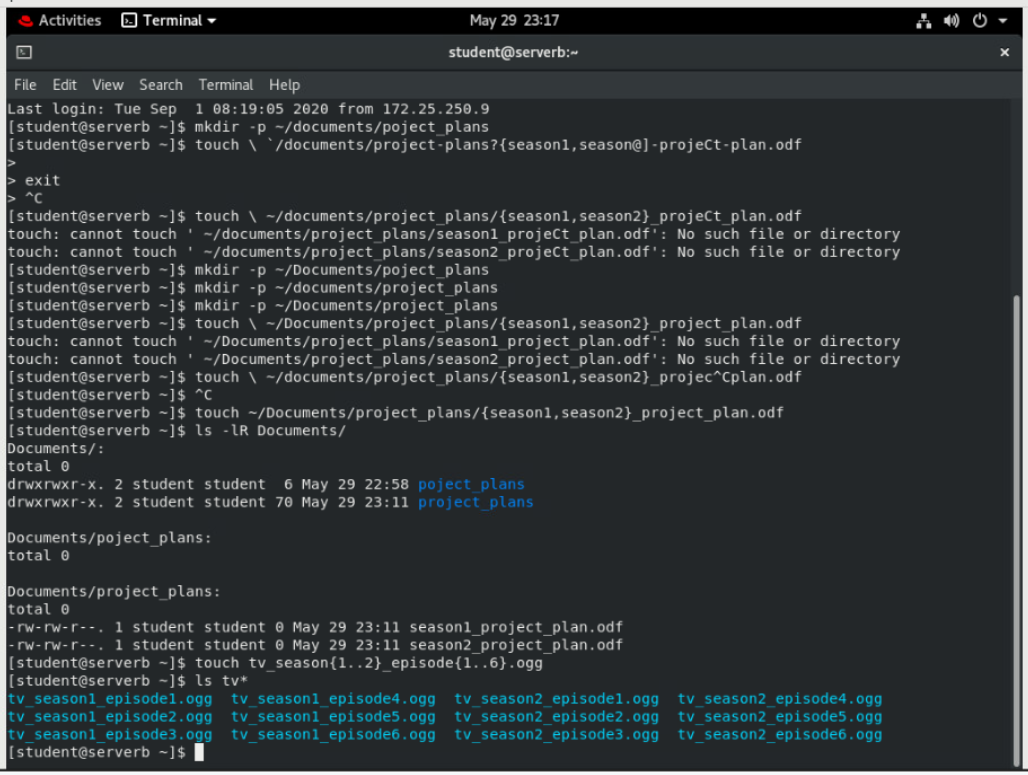
**[student@serverb ~]$ ls tv\***

tv\_season1\_episode1.ogg tv\_season1\_episode5.ogg tv\_season2\_episode3.ogg

tv\_season1\_episode2.ogg tv\_season1\_episode6.ogg tv\_season2\_episode4.ogg

tv\_season1\_episode3.ogg tv\_season2\_episode1.ogg tv\_season2\_episode5.ogg

tv\_season1\_episode4.ogg tv\_season2\_episode2.ogg tv\_season2\_episode6.ogg



1. As the author of a successful series of mystery novels, your next bestseller's chapters are being edited for publishing. Create a total of eight files with names mystery\_chapterX.odf. Replace *X* with the numbers 1 through 8.

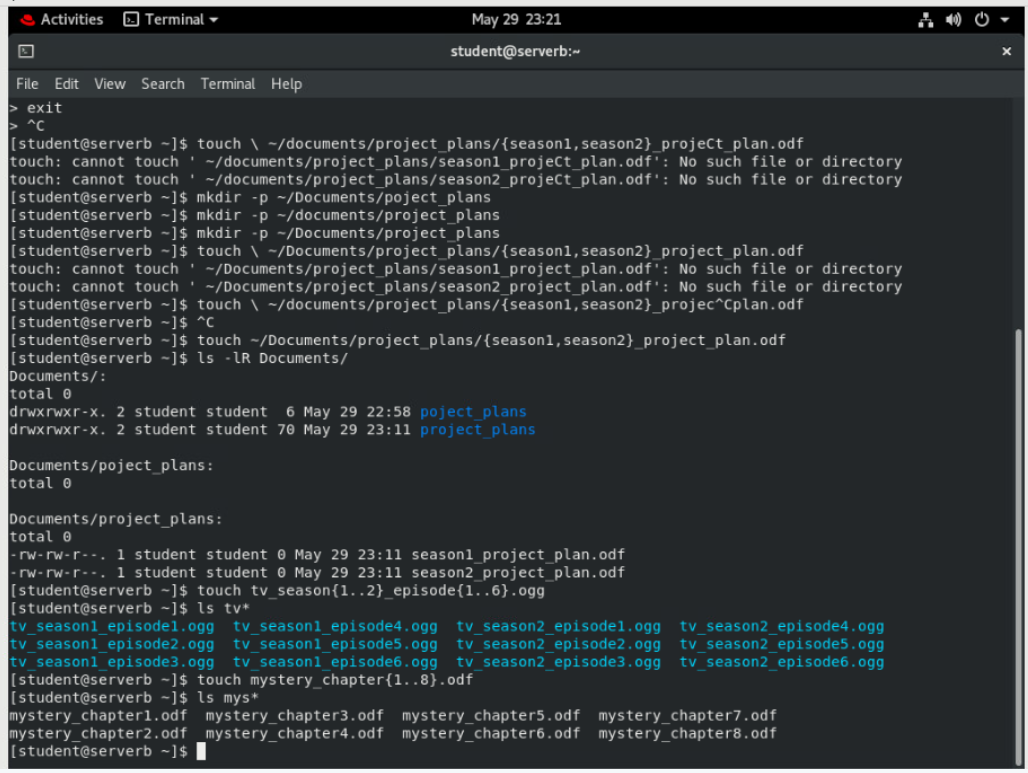
**[student@serverb ~]$ touch mystery\_chapter{1..8}.odf**

**[student@serverb ~]$ ls mys\***

mystery\_chapter1.odf mystery\_chapter4.odf mystery\_chapter7.odf

mystery\_chapter2.odf mystery\_chapter5.odf mystery\_chapter8.odf

mystery\_chapter3.odf mystery\_chapter6.odf

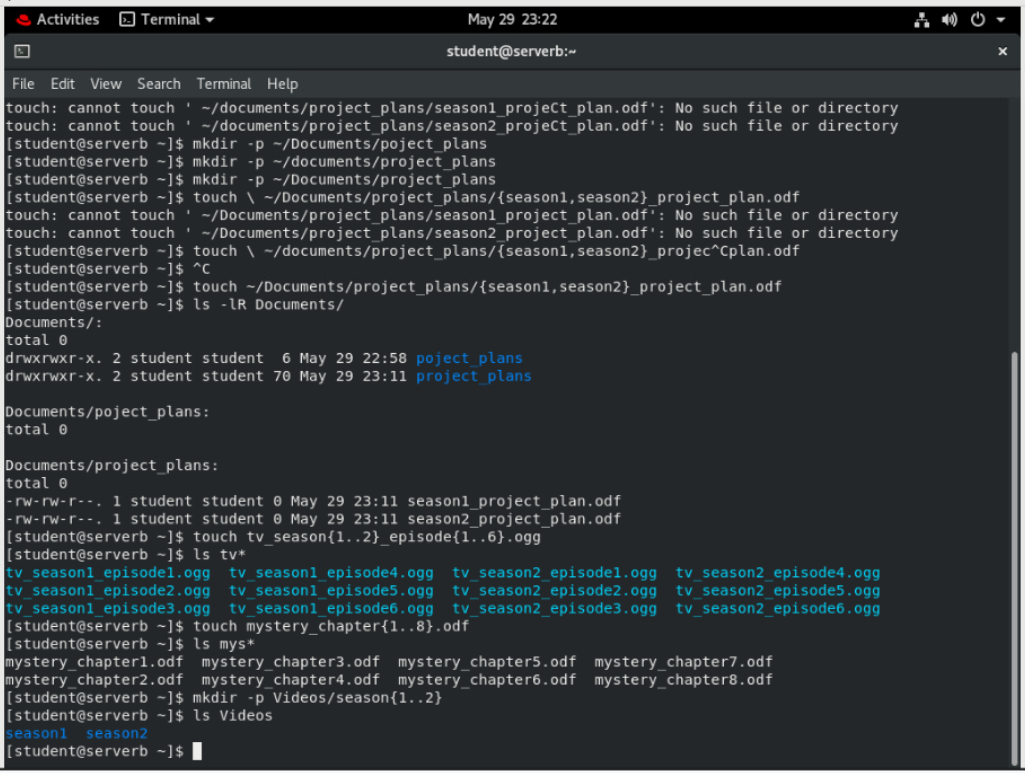


1. Use a single command to create two subdirectories named season1 and season2 under the Videos directory, to organize the TV episodes.

**[student@serverb ~]$ mkdir -p Videos/season{1..2}**

**[student@serverb ~]$ ls Videos**

season1 season2



1. Move the appropriate TV episodes into the season subdirectories. Use only two commands, specifying destinations using relative syntax.

**[student@serverb ~]$ mv tv\_season1\* Videos/season1**

**[student@serverb ~]$ mv tv\_season2\* Videos/season2**

**[student@serverb ~]$ ls -R Videos**

Videos:

season1 season2

Videos/season1:

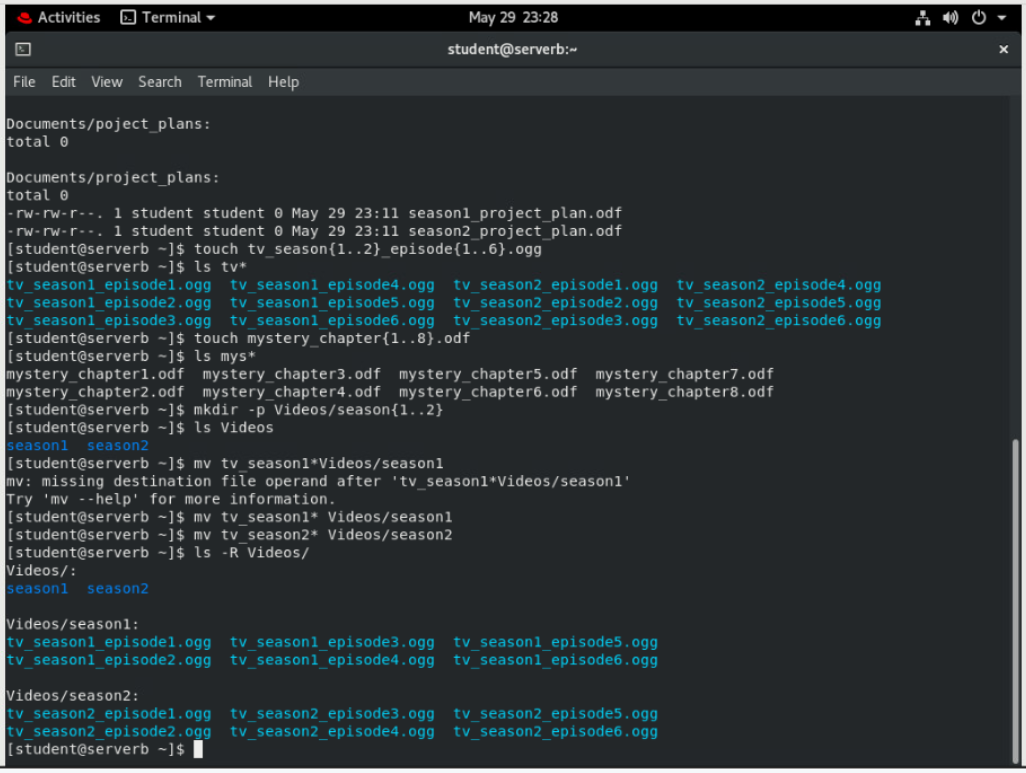
tv\_season1\_episode1.ogg tv\_season1\_episode3.ogg tv\_season1\_episode5.ogg

tv\_season1\_episode2.ogg tv\_season1\_episode4.ogg tv\_season1\_episode6.ogg

Videos/season2:

tv\_season2\_episode1.ogg tv\_season2\_episode3.ogg tv\_season2\_episode5.ogg

tv\_season2\_episode2.ogg tv\_season2\_episode4.ogg tv\_season2\_episode6.ogg



1. Create a 2-level directory hierarchy with a single command to organize the mystery book chapters. Create my\_bestseller under the Documents directory, and chapters under the new my\_bestseller directory.

**[student@serverb ~]$ mkdir -p Documents/my\_bestseller/chapters**

**[student@serverb ~]$ ls -R Documents**

Documents:

my\_bestseller project\_plans

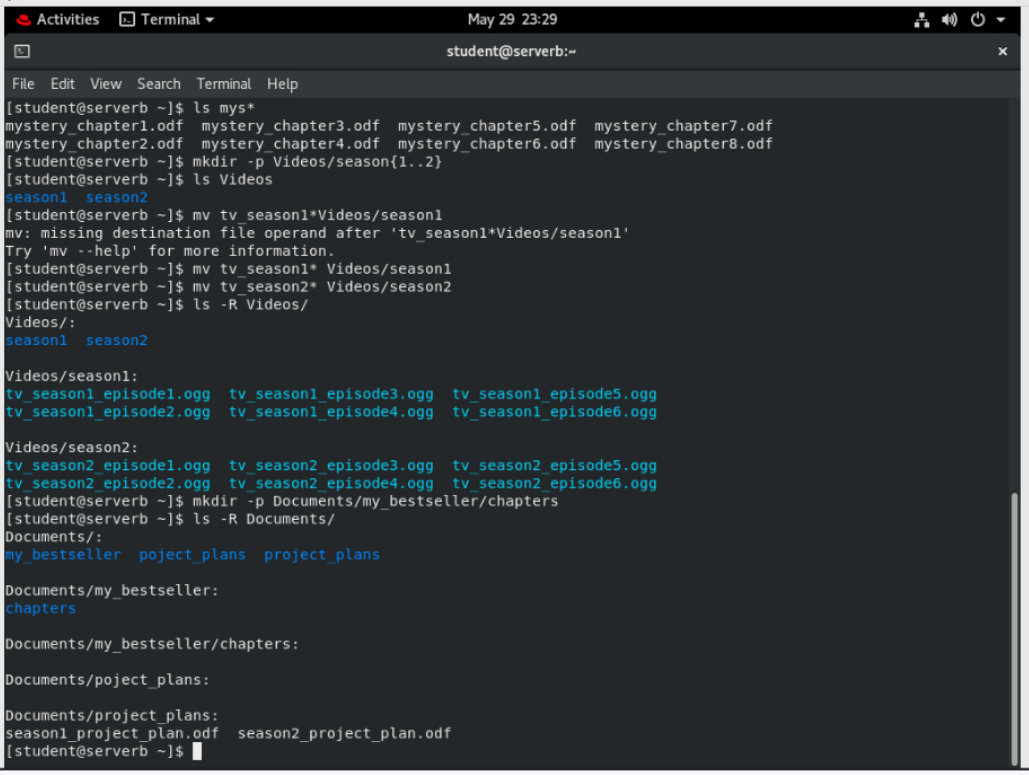
Documents/my\_bestseller:

chapters

Documents/my\_bestseller/chapters:

Documents/project\_plans:

season1\_project\_plan.odf season2\_project\_plan.odf



1. Create three more subdirectories directly under the my\_bestseller directory using a single command. Name these subdirectories editor, changes, and vacation. The -p option (create parents) is not needed because the my\_bestseller parent directory already exists.

**[student@serverb ~]$ mkdir Documents/my\_bestseller/{editor,changes,vacation}**

**[student@serverb ~]$ ls -R Documents**

Documents:

my\_bestseller project\_plans

Documents/my\_bestseller:

changes chapters editor vacation

Documents/my\_bestseller/changes:

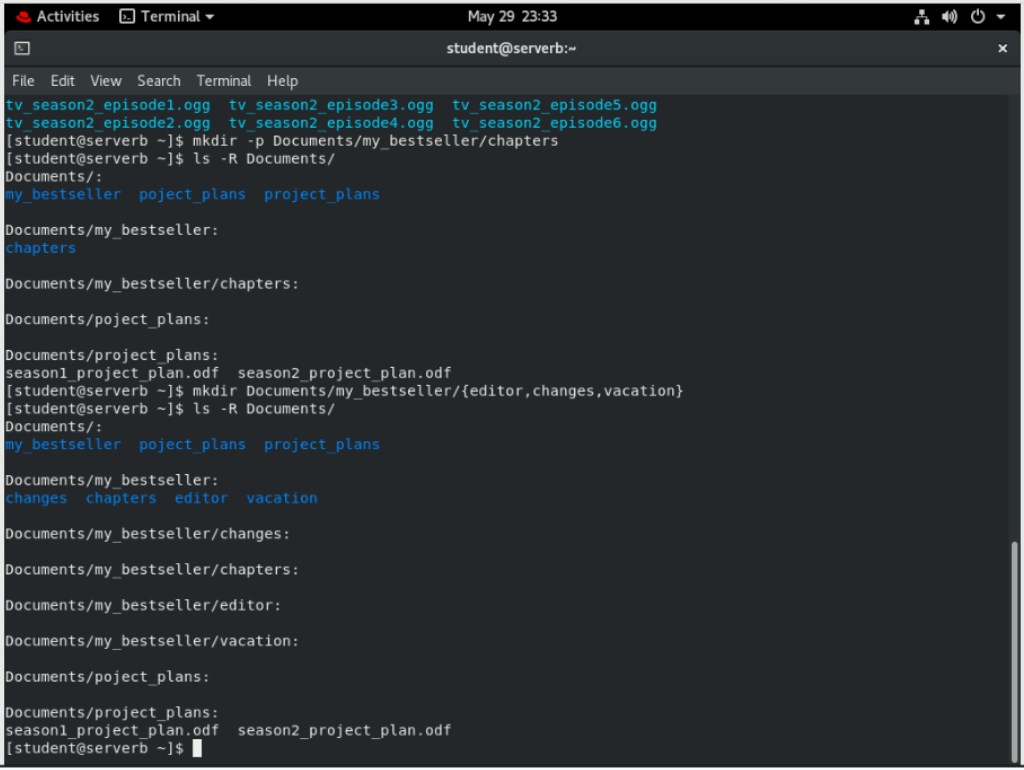
Documents/my\_bestseller/chapters:

Documents/my\_bestseller/editor:

Documents/my\_bestseller/vacation:

Documents/project\_plans:

season1\_project\_plan.odf season2\_project\_plan.odf



1. Change to the chapters directory. Using the tilde (~) home directory shortcut to specify the source files, move all book chapters to the chapters directory, which is now your current directory. What is the simplest syntax to specify the destination directory?

**[student@serverb ~]$ cd Documents/my\_bestseller/chapters**

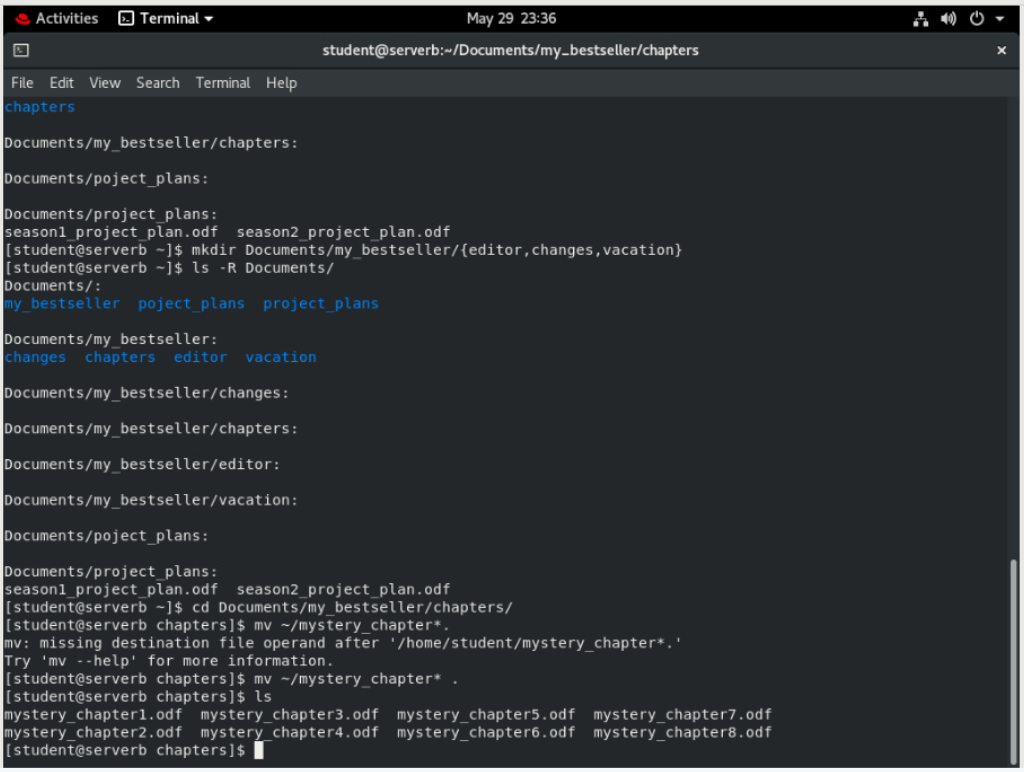
**[student@serverb chapters]$ mv ~/mystery\_chapter\* .**

**[student@serverb chapters]$ ls**

mystery\_chapter1.odf mystery\_chapter4.odf mystery\_chapter7.odf

mystery\_chapter2.odf mystery\_chapter5.odf mystery\_chapter8.odf

mystery\_chapter3.odf mystery\_chapter6.odf



1. You sent the first two chapters to the editor for review. Move only those two chapters to the editor directory to avoid modifying them during the review. Starting from the chapters subdirectory, use brace expansion with a range to specify the chapter file names to move and a relative path for the destination directory.

**[student@serverb chapters]$ mv mystery\_chapter{1..2}.odf ../editor**

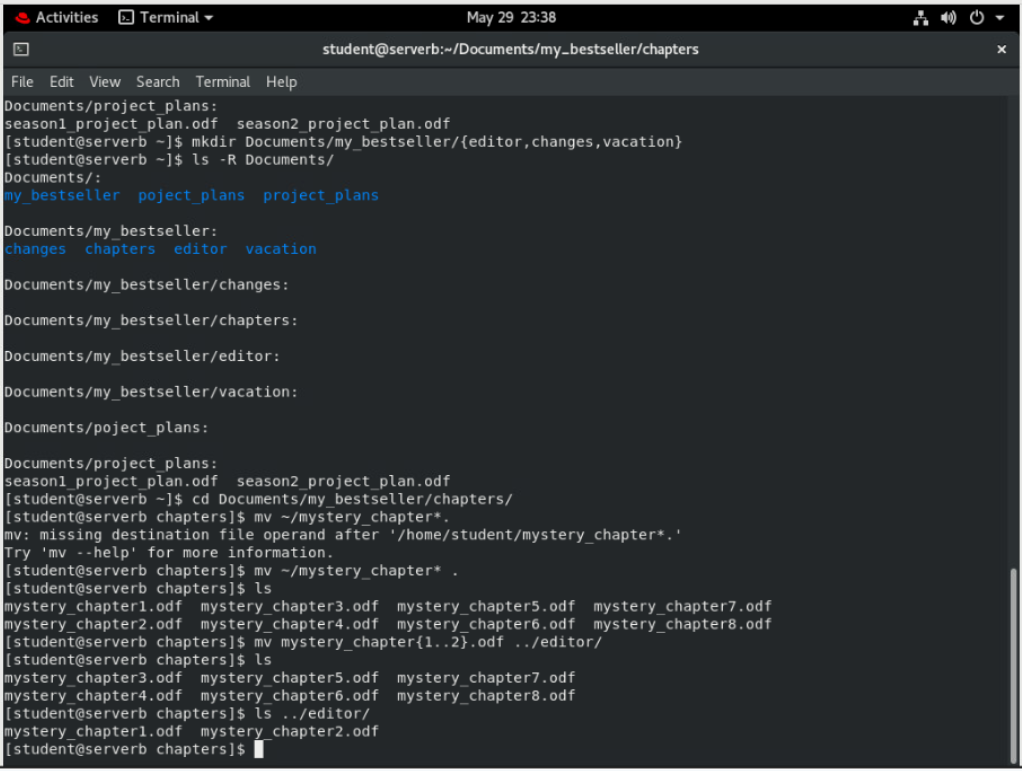
**[student@serverb chapters]$ ls**

mystery\_chapter3.odf mystery\_chapter5.odf mystery\_chapter7.odf

mystery\_chapter4.odf mystery\_chapter6.odf mystery\_chapter8.odf

**[student@serverb chapters]$ ls ../editor**

mystery\_chapter1.odf mystery\_chapter2.odf



1. While on vacation you intend to write chapters 7 and 8. Use a single command to move the files from the chapters directory to the vacation directory. Specify the chapter file names using brace expansion with a list of strings and without using wildcard characters.

**[student@serverb chapters]$ mv mystery\_chapter{7,8}.odf ../vacation**

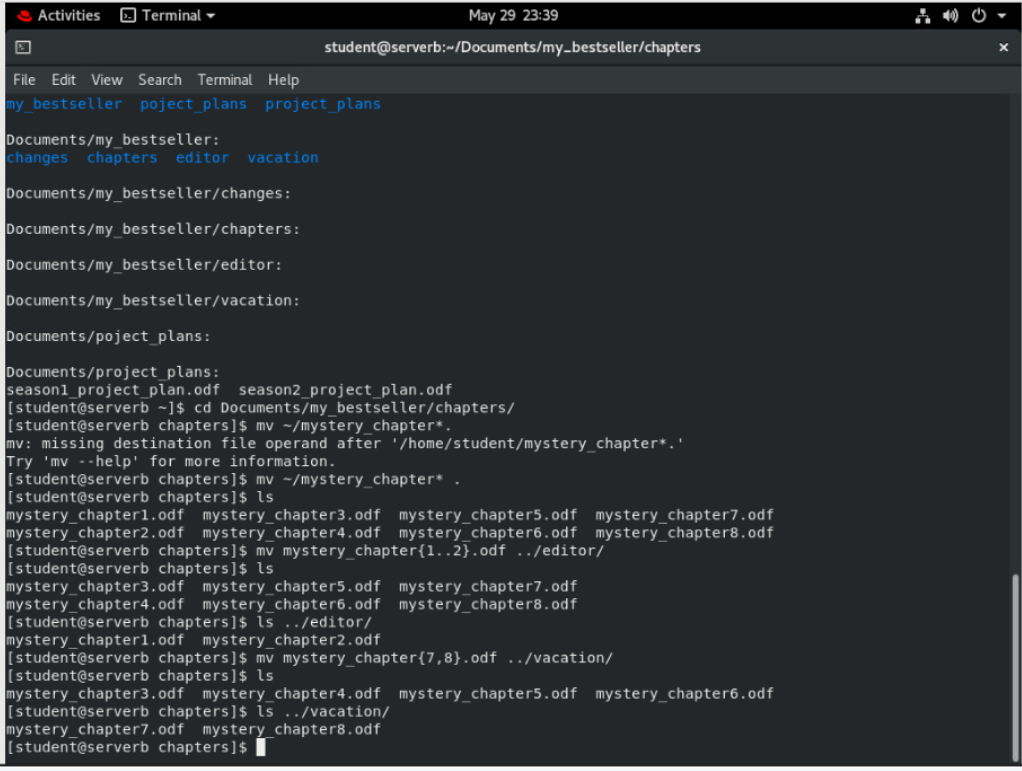
**[student@serverb chapters]$ ls**

mystery\_chapter3.odf mystery\_chapter5.odf

mystery\_chapter4.odf mystery\_chapter6.odf

**[student@serverb chapters]$ ls ../vacation**

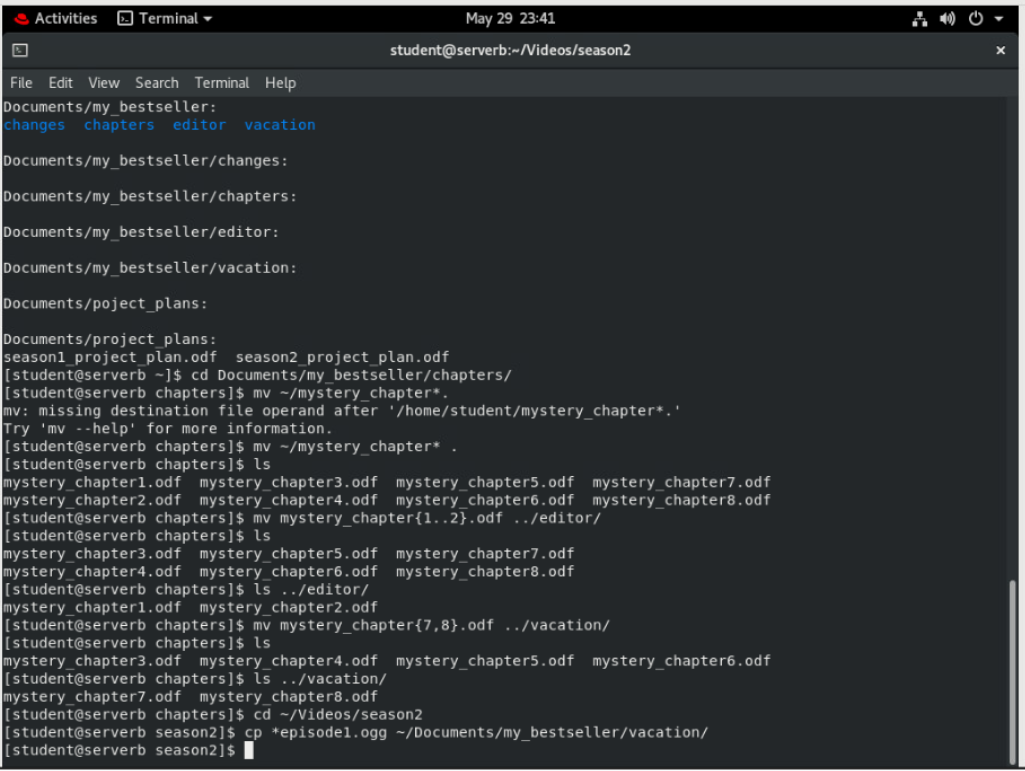
mystery\_chapter7.odf mystery\_chapter8.odf



1. Change your working directory to ~/Videos/season2, and then copy the first episode of the season to the vacation directory.

**[student@serverb chapters]$ cd ~/Videos/season2**

[student@serverb season2]$ cp \*episode1.ogg ~/Documents/my\_bestseller/vacation



1. Use a single **cd** command to change from your working directory to the ~/Documents/my\_bestseller/vacation directory. List its files. Use the *previous working directory* argument to return to the season2 directory. (This will succeed if the last directory change with the **cd** command was accomplished with one command rather than several **cd** commands.) From the season2 directory, copy the episode 2 file into the vacation directory. Use the shortcut again to return to the vacation directory.

**[student@serverb season2]$ cd ~/Documents/my\_bestseller/vacation**

**[student@serverb vacation]$ ls**

mystery\_chapter7.odf mystery\_chapter8.odf tv\_season2\_episode1.ogg

**[student@serverb vacation]$ cd -**

/home/ec2-user/Videos/season2

**[student@serverb season2]$ cp \*episode2.ogg ~/Documents/my\_bestseller/vacation**

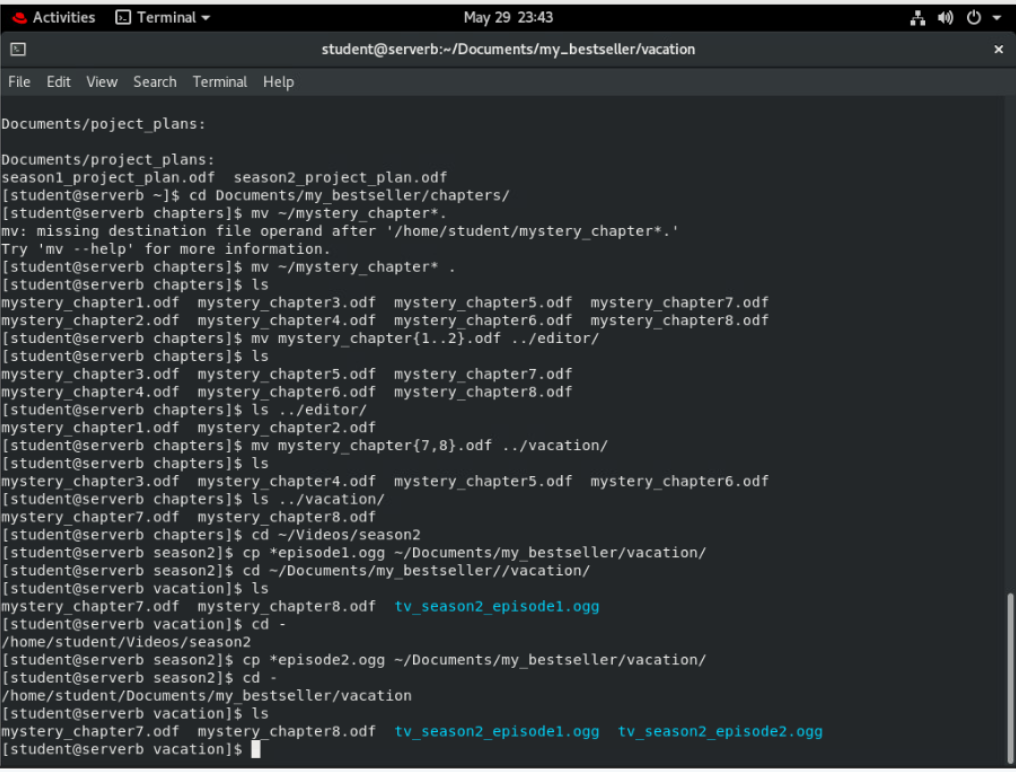
**[student@serverb vacation]$ cd -**

/home/ec2-user/Documents/my\_bestseller/vacation

**[student@serverb vacation]$ ls**

mystery\_chapter7.odf tv\_season2\_episode1.ogg

mystery\_chapter8.odf tv\_season2\_episode2.ogg



1. The authors of chapters 5 and 6 want to experiment with possible changes. Copy both files from the ~/Documents/my\_bestseller/chapters directory to the ~/Documents/my\_bestseller/changes directory to prevent these changes from modifying original files. Navigate to the ~/Documents/my\_bestseller directory. Use square-bracket pattern matching to specify which chapter numbers to match in the filename argument of the **cp** command.

**[student@serverb vacation]$ cd ~/Documents/my\_bestseller**

**[student@serverb my\_bestseller]$ cp chapters/mystery\_chapter[56].odf changes**

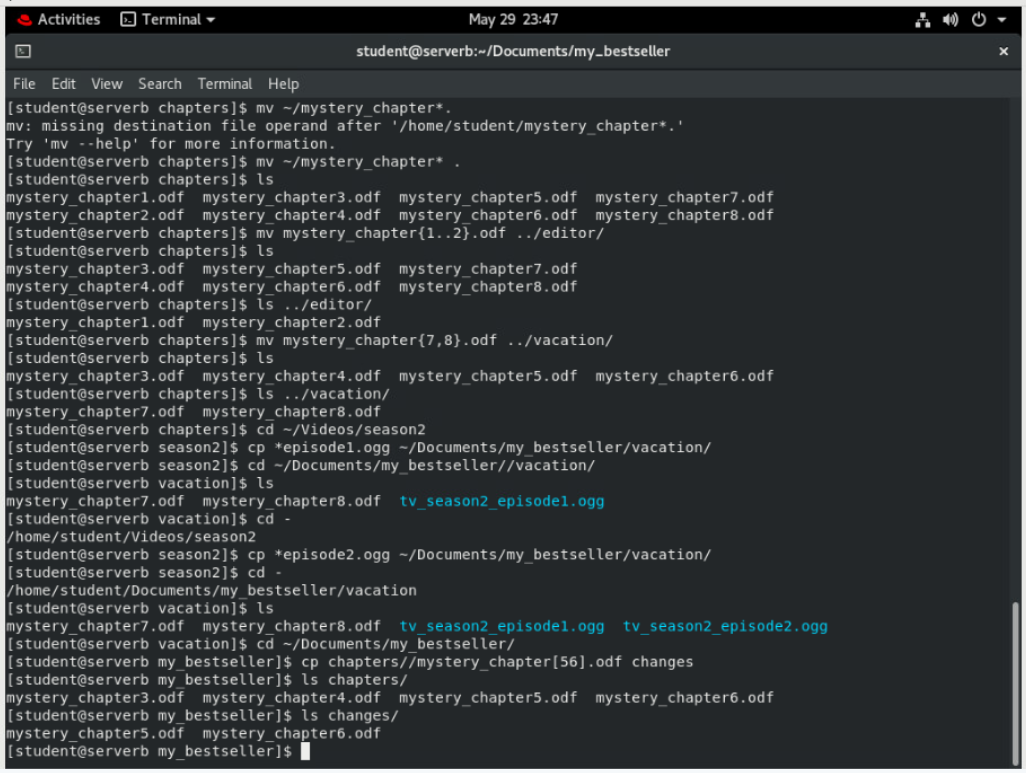
**[student@serverb my\_bestseller]$ ls chapters**

mystery\_chapter3.odf mystery\_chapter5.odf

mystery\_chapter4.odf mystery\_chapter6.odf

**[student@serverb my\_bestseller]$ ls changes**

mystery\_chapter5.odf mystery\_chapter6.odf



1. Change your current directory to the change’s directory.

Use the **date +%F** command with command substitution to copy mystery\_chapter5.odf to a new file which includes the full date. The name should have the form mystery\_chapter5\_YYYY-MM-DD.odf.

Make another copy of mystery\_chapter5.odf, appending the current time stamp (as the number of seconds since the epoch, 1970-01-01 00:00 UTC) to ensure a unique file name. Use command substitution with the **date +%s** command to accomplish this.

**[student@serverb my\_bestseller]$ cd changes**

**[student@serverb changes]$ cp mystery\_chapter5.odf \**

**mystery\_chapter5\_$(date +%F).odf**

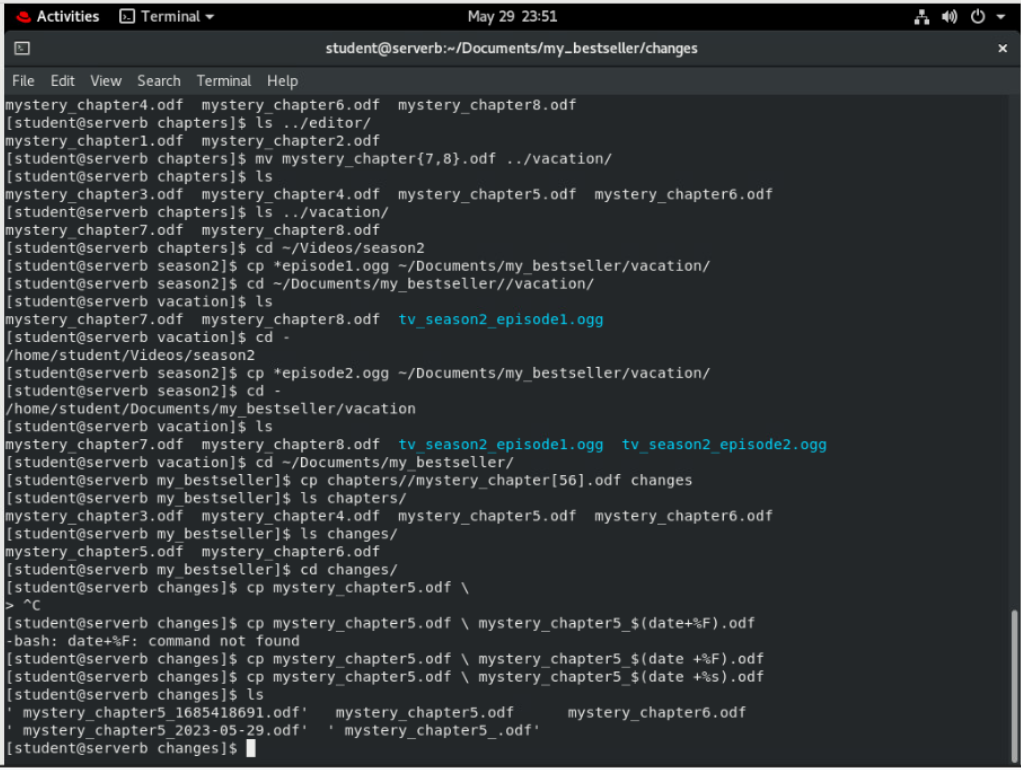
**[student@serverb changes]$ cp mystery\_chapter5.odf \**

**mystery\_chapter5\_$(date +%s).odf**

**[student@serverb changes]$ ls**

mystery\_chapter5\_1492545076.odf mystery\_chapter5.odf

mystery\_chapter5\_2017-04-18.odf mystery\_chapter6.odf



1. After further review, you decide that the plot changes are not necessary. Delete the changes directory.

If necessary, navigate to the changes directory and delete all the files within the directory. You cannot delete a directory while it is the current working directory. Change to the parent directory of the change’s directory. Try to delete the empty directory using the **rm** command without the -r recursive option. This attempt should fail. Finally, use the **rmdir** command to delete the empty directory, which will succeed.

**[student@serverb changes]$ rm mystery\***

**[student@serverb changes]$ cd ..**

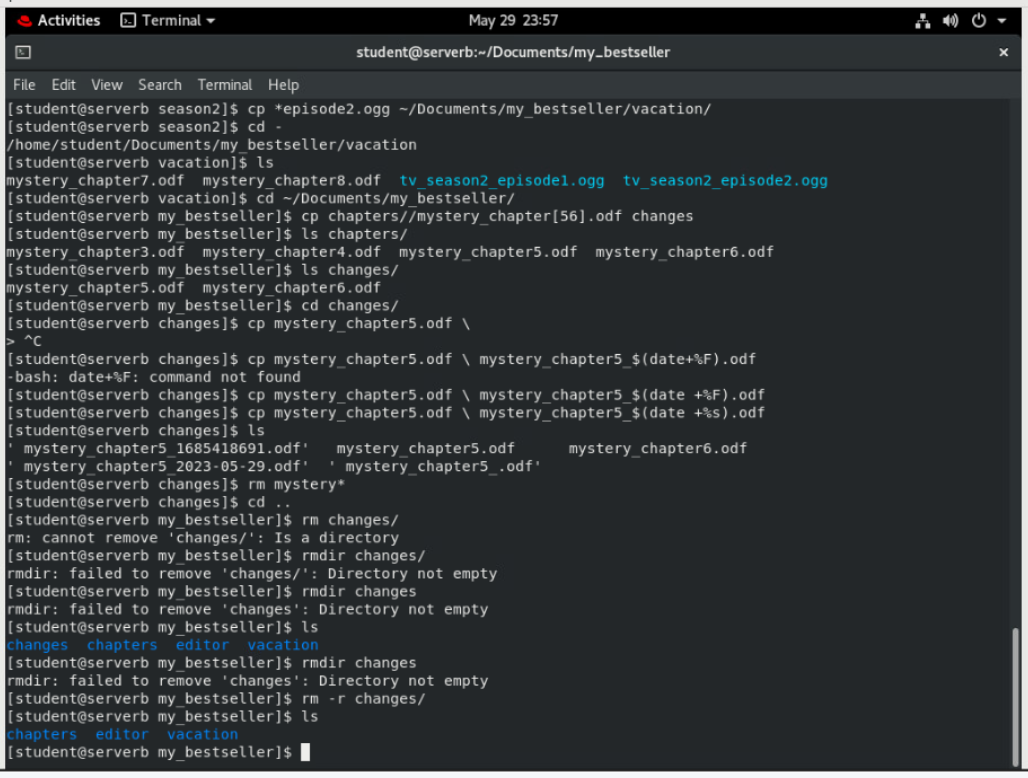
**[student@serverb my\_bestseller]$ rm changes**

rm: cannot remove 'changes': Is a directory

**[student@serverb my\_bestseller]$ rmdir changes**

**[student@serverb my\_bestseller]$ ls**

chapters editor vacation



1. When the vacation is over, the vacation directory is no longer needed. Delete it using the **rm** command with the *recursive* option.

When finished, return to the student user's home directory.

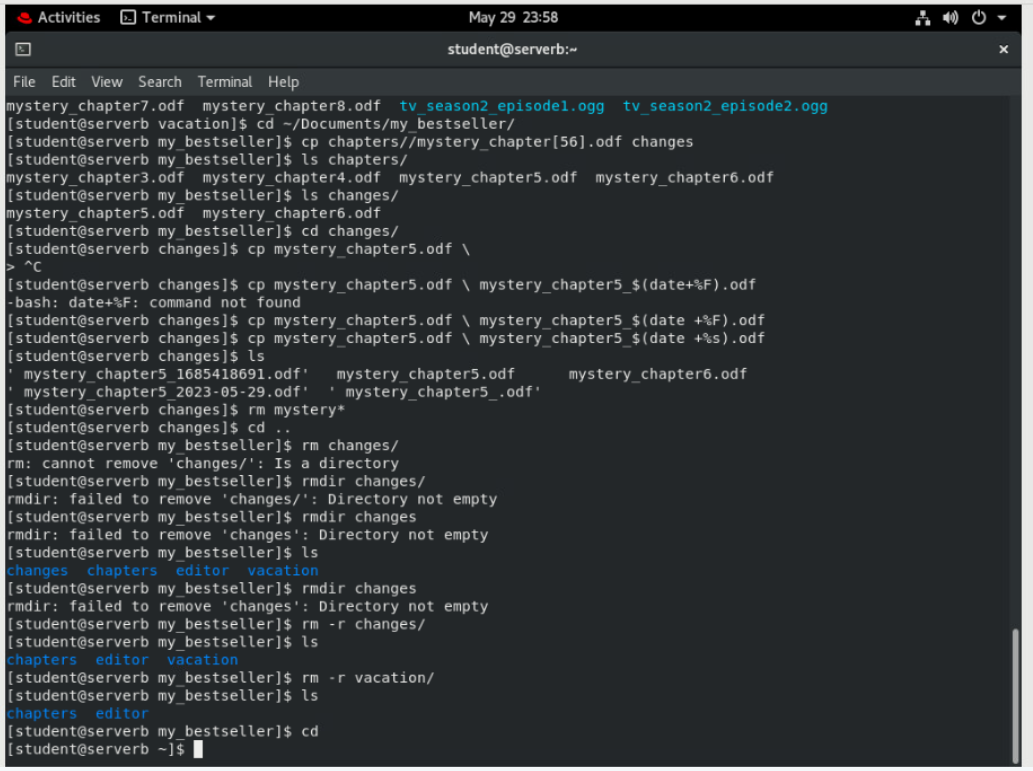
**[student@serverb my\_bestseller]$ rm -r vacation**

**[student@serverb my\_bestseller]$ ls**

chapters editor

**[student@serverb my\_bestseller]$ cd**

**[student@serverb ~]$**



1. Create a hard link to the ~/Documents/project\_plans/season2\_project\_plan.odf file named ~/Documents/backups/season2\_project\_plan.odf.back. A hard link will protect against accidental deletion of the original file and will keep the backup file updated as changes are made to the original.

Notice that the link count is 2 for both season2\_project\_plan.odf.back and season2\_project\_plan.odf files.

**[student@serverb ~]$ mkdir ~/Documents/backups**

**[student@serverb ~]$ ln ~/Documents/project\_plans/season2\_project\_plan.odf \**

**~/Documents/backups/season2\_project\_plan.odf.back**

**[student@serverb ~]$ ls -lR ~/Documents/**

/home/student/Documents/:

total 0

drwxrwxr-x. 2 student student 43 Jan 31 18:59 backups

drwxrwxr-x. 4 student student 36 Jan 31 19:42 my\_bestseller

drwxrwxr-x. 2 student student 70 Jan 31 18:20 project\_plans

/home/student/Documents/backups:

total 4

**-rw-rw-r--. 2 student student 0 Jan 31 19:05 season2\_project\_plan.odf.back**

/home/student/Documents/my\_bestseller:

total 0

drwxrwxr-x. 2 student student 118 Jan 31 19:39 chapters

drwxrwxr-x. 2 student student 62 Jan 31 19:38 editor

/home/student/Documents/my\_bestseller/chapters:

total 0

-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery\_chapter3.odf

-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery\_chapter4.odf

-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery\_chapter5.odf

-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery\_chapter6.odf

/home/student/Documents/my\_bestseller/editor:

total 0

-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery\_chapter1.odf

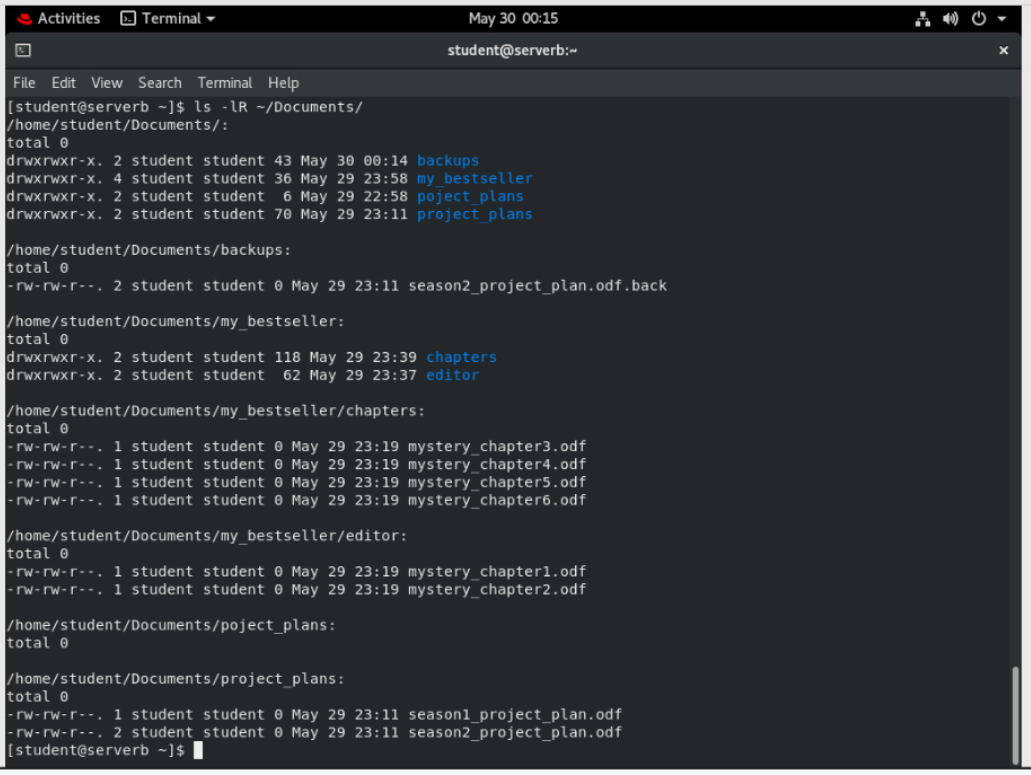
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery\_chapter2.odf

/home/student/Documents/project\_plans:

total 4

-rw-rw-r--. 1 student student 0 Jan 31 18:20 season1\_project\_plan.odf

**-rw-rw-r--. 2 student student 0 Jan 31 19:05 season2\_project\_plan.odf**



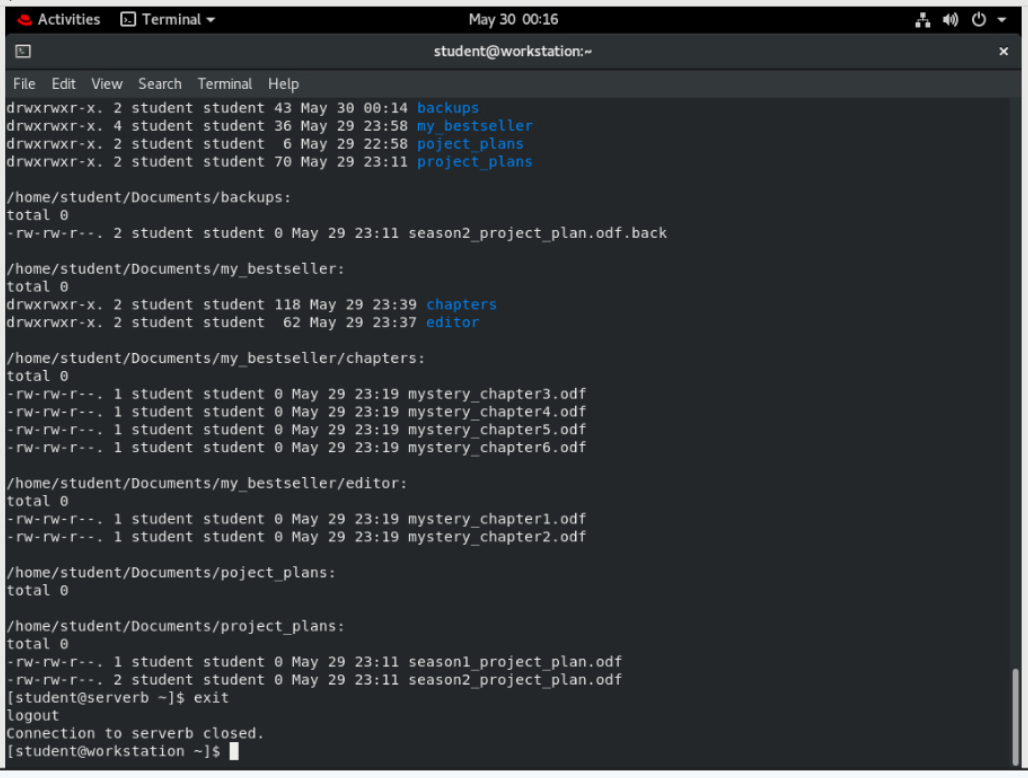
1. Exit from serverb.

**[student@serverb ~]$ exit**

logout

Connection to serverb closed.

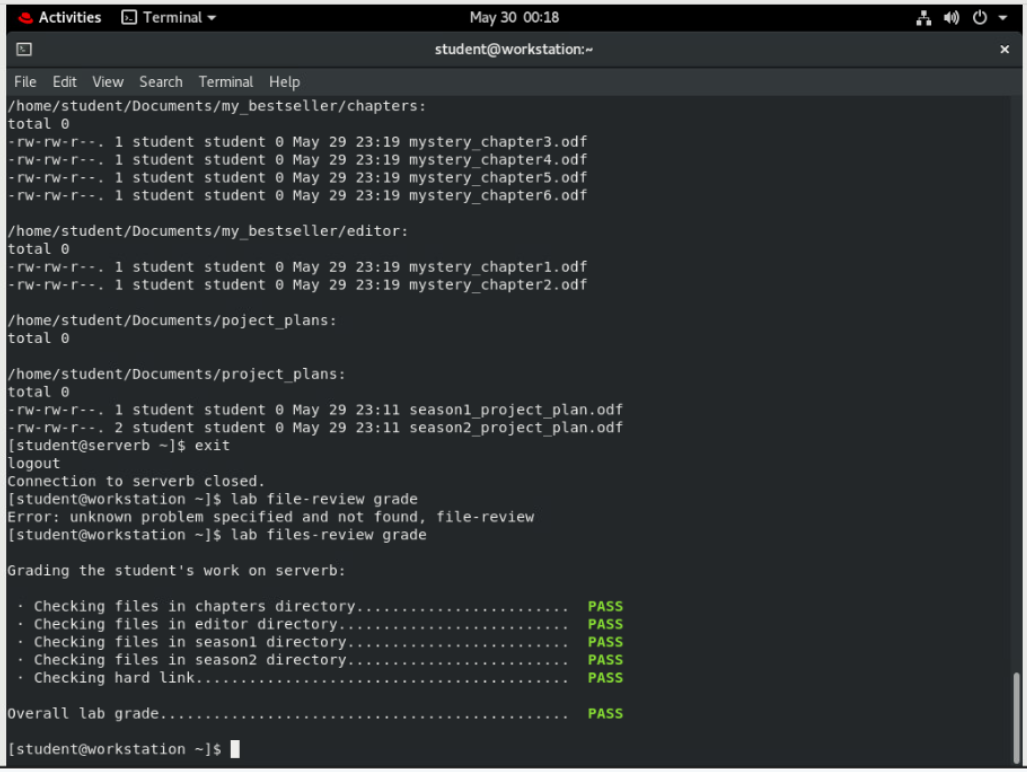
[student@workstation ~]$

****

**Evaluation:**

On workstation, run the **lab files-review grade** script to confirm success on this lab.

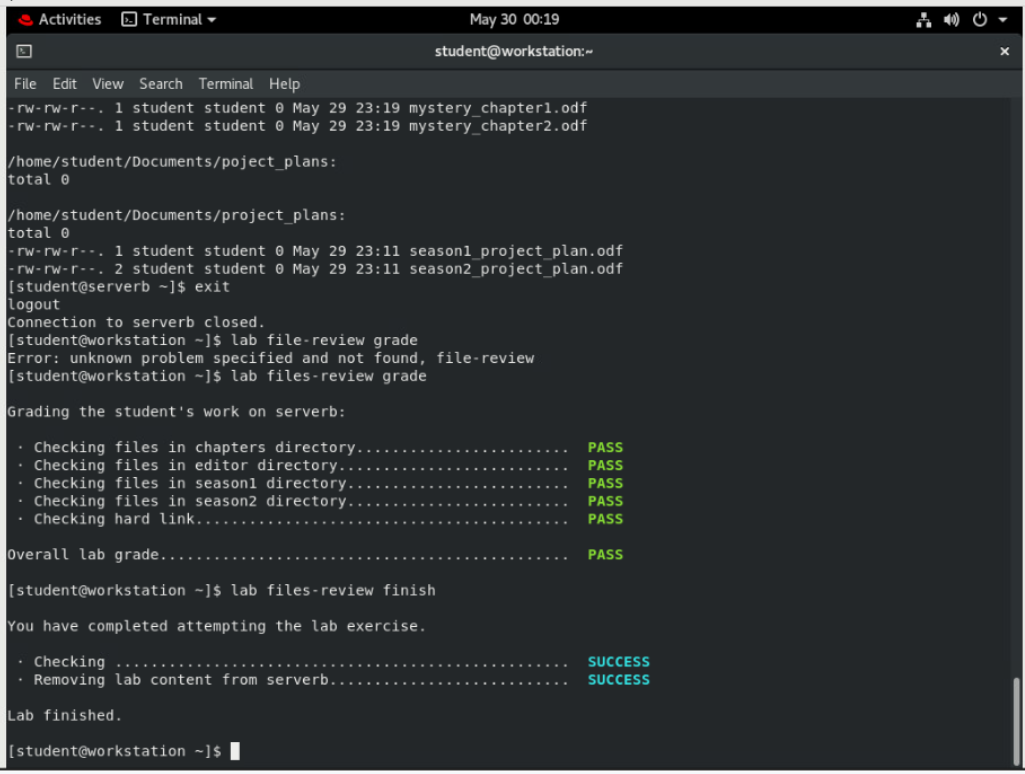
**[student@workstation ~]$ lab files-review grade**

****

**Finish:**

On workstation, run the **lab files-review finish** script to finish this lab. This script removes all files and directories created on serverb during the lab exercise.

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



This concludes the lab.