

Lab: Managing Files with Shell Expansion

Performance checklist

In this lab, you will create, move, and remove files and folders using a variety of file name matching shortcuts.

Outcomes:

Familiarity and practice with many forms of wildcards for locating and using files.

Before you begin...

Perform the following steps on serverX unless directed otherwise. Log in as **student** and begin the lab in the home directory.

1. To begin, create sets of empty practice files to use in this lab. If an intended shell expansion shortcut is not immediately recognized, students are expected to 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.
2. 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.
3. To organize the TV episodes, create two subdirectories named **season1** and **season2** under the existing **Videos** directory. Use one command.
4. Move the appropriate TV episodes into the season subdirectories. Use only two commands, specifying destinations using relative syntax.
5. To organize the mystery book chapters, create a two-level directory hierarchy with one command. Create **my_bestseller** under the existing **Documents** directory, and **chapters** beneath the new **my_bestseller** directory.
6. Using one command, create three more subdirectories directly under the **my_bestseller** directory. Name these subdirectories **editor**, **plot_change**, and **vacation**. The *create parent* option is not needed since the **my_bestseller** parent directory already exists.
7. Change to the **chapters** directory. Using the home directory shortcut to specify the source files, move all book chapters into the **chapters** directory, which is now your current directory. What is the simplest syntax to specify the destination directory?
8. The first two chapters are sent to the editor for review. To remember to not modify these chapters during the review, move those two chapters only to the **editor** directory. Use relative syntax starting from the **chapters** subdirectory.
9. Chapters 7 and 8 will be written while on vacation. Move the files from **chapters** to **vacation**. Use one command without wildcard characters.
10. With one command, change the working directory to the season 2 TV episodes location, then copy the first episode of the season to the **vacation** directory.

11. With one command, change the working directory to **vacation**, then list its files. Episode 2 is also needed. Return to the **season2** directory using the *previous working directory* shortcut. This will succeed if the last directory change was accomplished with one command. Copy the episode 2 file into **vacation**. Return to **vacation** using the shortcut again.
12. Chapters 5 and 6 may need a plot change. To prevent these changes from modifying original files, copy both files into **plot_change**. Move up one directory to **vacation**'s parent directory, then use one command from there.
13. To track changes, make three backups of chapter 5. Change to the **plot_change** directory. Copy **mystery_chapter5.odf** as a new file name to include the full date (Year-Mo-Da). Make another copy appending the current timestamp (as the number of seconds since the *epoch*) to ensure a unique file name. Also make a copy appending the current user (*\$USER*) to the file name. See the solution for the syntax of any you are unsure of (like what arguments to pass the **date**).

Note, we could also make the same backups of the chapter 6 files too.

14. The plot changes were not successful. Delete the **plot_change** directory. First, delete all of the files in the **plot_change** directory. Change directory up one level because the directory cannot be deleted while it is the working directory. Try to delete the directory using the **rm** command *without* the *recursive* option. This attempt should fail. Now use the **rmdir** command, which will succeed.
15. 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 home directory.