Practice: Command-Line File Management

Guided exercise

In this lab, you will practice efficient techniques for creating and organizing files using directories, file copies, and links.

Outcomes:
Students will practice creating, rearranging, and deleting files.

Before you begin...
Log into your student account on serverX. Begin in your home directory.

□ 1. In your home directory, create sets of empty practice files to use for the remainder of this lab. If the intended command is not immediately recognized, students are expected to use the guided solution to see and practice how the task is accomplished. Use the shell tab completion to locate and complete path names more easily.

Create six files with names of the form **songX.mp3**.

Create six files with names of the form **snapX.jpg**.

Create six files with names of the form **filmX.avi**.

In each set, replace X with the numbers 1 through 6.

```
[student@serverX ~]$ touch song1.mp3 song2.mp3 song3.mp3 song4.mp3 song5.mp3
 song6.mp3
[student@serverX ~]$ touch snap1.jpg snap2.jpg snap3.jpg snap4.jpg snap5.jpg
 snap6.jpg
[student@serverX ~]$ touch film1.avi film2.avi film3.avi film4.avi film5.avi
 film6.avi
[student@serverX ~]$ ls -l
```

□ 2. From your home directory, move the song files into your **Music** subdirectory, the snapshot files into your **Pictures** subdirectory, and the movie files into your **Videos** subdirectory.

When distributing files from one location to many locations, first change to the directory containing the *source* files. Use the simplest path syntax, absolute or relative, to reach the destination for each file management task.

```
[student@serverX ~]$ mv song1.mp3 song2.mp3 song3.mp3 song4.mp3 song5.mp3
 song6.mp3 Music
[student@serverX ~]$ mv snap1.jpg snap2.jpg snap3.jpg snap4.jpg snap5.jpg
 snap6.jpg Pictures
[student@serverX ~]$ mv film1.avi film2.avi film3.avi film4.avi film5.avi
 film6.avi Videos
[student@serverX ~]$ ls -l Music Pictures Videos
```

□ 3. In your home directory, create three subdirectories for organizing your files into projects. Call these directories **friends**, **family**, and **work**. Create all three with one command.

You will use these directories to rearrange your files into projects.

```
[student@serverX ~]$ mkdir friends family work
[student@serverX ~]$ ls -l
```

□ 4. You will collect some of the new files into the project directories for family and friends. Use as many commands as needed. You do not have to use only one command as in the example. For each project, first change to the project directory, then copy the source files into this directory. You are making copies, since you will keep the originals after giving these projects to family and friends.

Copy files (all types) containing numbers 1 and 2 to the friends folder.

Copy files (all types) containing numbers 3 and 4 to the family folder.

When collecting files from multiple locations into one location, change to the directory that will contain the *destination* files. Use the simplest path syntax, absolute or relative, to reach the source for each file management task.

```
[student@serverX ~]$ cd friends
[student@serverX friends]$ cp ~/Music/song1.mp3 ~/Music/song2.mp3 ~/Pictures/
snap1.jpg ~/Pictures/snap2.jpg ~/Videos/film1.avi ~/Videos/film2.avi .
[student@serverX friends]$ ls -l
[student@serverX friends]$ cd ../family
[student@serverX family]$ cp ~/Music/song3.mp3 ~/Music/song4.mp3 ~/Pictures/
snap3.jpg ~/Pictures/snap4.jpg ~/Videos/film3.avi ~/Videos/film4.avi .
[student@serverX family]$ ls -l
```

□ 5. For your work project, you will create additional copies.

```
[student@serverX family]$ cd ../work
[student@serverX work]$ cp ~/Music/song5.mp3 ~/Music/song6.mp3 ~/Pictures/
snap5.jpg ~/Pictures/snap6.jpg ~/Videos/film5.avi ~/Videos/film6.avi .
[student@serverX work]$ ls -l
```

□ 6. Your projects are now done. Time to clean up the projects.

Change to your home directory. Attempt to delete both the family and friends projects with a single **rmdir** command.

```
[student@serverX work]$ cd
[student@serverX ~]$ rmdir family friends
rmdir: failed to remove `family': Directory not empty
rmdir: failed to remove `friends': Directory not empty
```

Using the **rmdir** command should fail since both directories are non-empty.

□ 7. Use another command that will succeed in deleting both the family and friends folders.

```
[student@serverX ~]$ rm -r family friends
[student@serverX ~]$ ls -l
```

□ 8. Delete all the files in the work project, but do not delete the work directory.

```
[student@serverX ~]$ cd work
[student@serverX work]$ rm song5.mp3 song6.mp3 snap5.jpg snap6.jpg film5.avi
 film6.avi
[student@serverX work]$ ls -l
```

□ 9. Finally, from your home directory, use the **rmdir** command to delete the work directory. The command should succeed now that it is empty.

```
[student@serverX work]$ cd
[student@serverX ~]$ rmdir work
[student@serverX ~]$ ls -l
```