Red Hat System Administration I

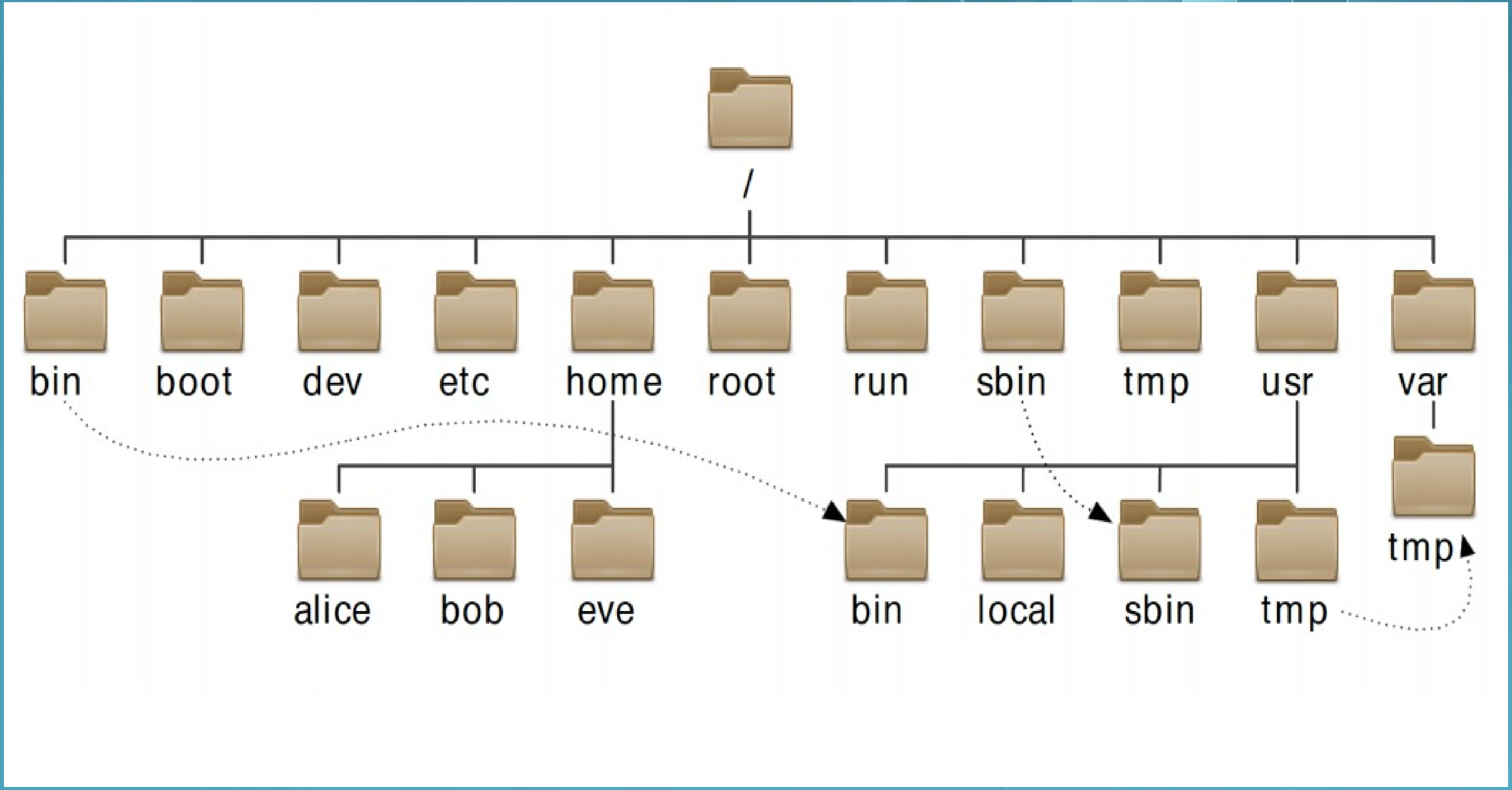
UNIT 2

Managing Files Form The Command Line

Objectives

- identify the purpose for important directorys on a linux system
- Specify files using absolute and reative path names
- Create copy,move and remove files and directories using command-line utilties
- Match one or more file names using shell expansion as argument to shell commands

The file system hierarchy



Important Red Hat Enterprise Linux directories

- /usr
  - Installed software, shared libraries, include files, and static read-only program data. Important subdirectories include:

    [/usr/bin]:

    User commands.

    [/usr/sbin]:

    Systemadministration commands.

    [/usr/local]:

    Locally customized software
- /etc
  - Configuration files specific to this system
- /var
  - system data directory
- /run
  - Runtime data for processes started since the last boot. This includes process ID files and lock files,among other things. The contents of this directory are recreated on reboot.
- /mnt
  - divice temporary mount point
- /home
  - Home directories where regular users store their personal data and configuration files
- /root
  - Home directory for the administrative superuser, root
- /tmp
  - A world-writable space for temporary files. Files which are more than 10 days old are deleted from this directory automatically. Another temporary directory exists, /var/tmp, in which files that have not

    been accessed, changed, or modified in more than 30 days are deleted automatically
- /boot
  - Files needed in order to start the boot process
- /dev
  - Contains special device files which are used by the system to access hardware

Locating Files by Name

- Absolute paths and relative paths
  - Absolute paths:

    The true path in the system file
  - relative paths:

    the name about relative to the current directory
- Navigating paths
  - Notes: please look at the example on next page

Example 1

```
[student@desktopX ~]$ pwd
/home/student
[student@desktopX ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[student@desktopX ~]$ cd Videos
[student@desktopX Videos]$ pwd
/home/student/Videos
[student@desktopX Videos]$ cd /home/student/Documents
[student@desktopX Documents]$ pwd
/home/student/Documents
[student@desktopX Documents]$ cd
[student@desktopX ~]$ pwd
/home/student
[student@desktopX ~]$ touch Videos/blockbuster1.ogg
[student@desktopX ~]$ touch Videos/blockbuster2.ogg
[student@desktopX ~]$ touch Documents/thesis_chapter1.odf
[student@desktopX ~]$ touch Documents/thesis_chapter2.odf
```

Example 2

```
[student@desktopX ~]$ ls -l
total 15
drwxr-xr-x. 2 student student 4096 Feb 7 14:02 Desktop
drwxr-xr-x. 2 student student 4096 Jan 9 15:00 Documents
... ...
drwxr-xr-x. 2 student student 4096 Jan 9 15:00 Videos
[student@desktopX ~]$ ls -a
total 15
drwx------. 16 student student 4096 Feb 8 16:15 . drwxr-xr-x. 6 root root 4096 Feb 8
16:13 .. -rw-------. 1 student student 22664 Feb 8 00:37 .bash_history
-rw-r--r--. 1 student student 18 Jul 9 2013 .bash_logout
... ...
drwxr-xr-x. 11 student student 4096 Feb 6 13:07 .gnome2
drwxr-xr-x. 2 student student 4096 Jan 9 15:00 Videos
[student@desktopX ~]$ ls -R
.:
Desktop Documents Downloads Music Pictures Public Templates Videos
./Desktop:
./Documents:
thesis_chapter1.odf thesis_chapter2.odf
... ...
```

Example 3-1

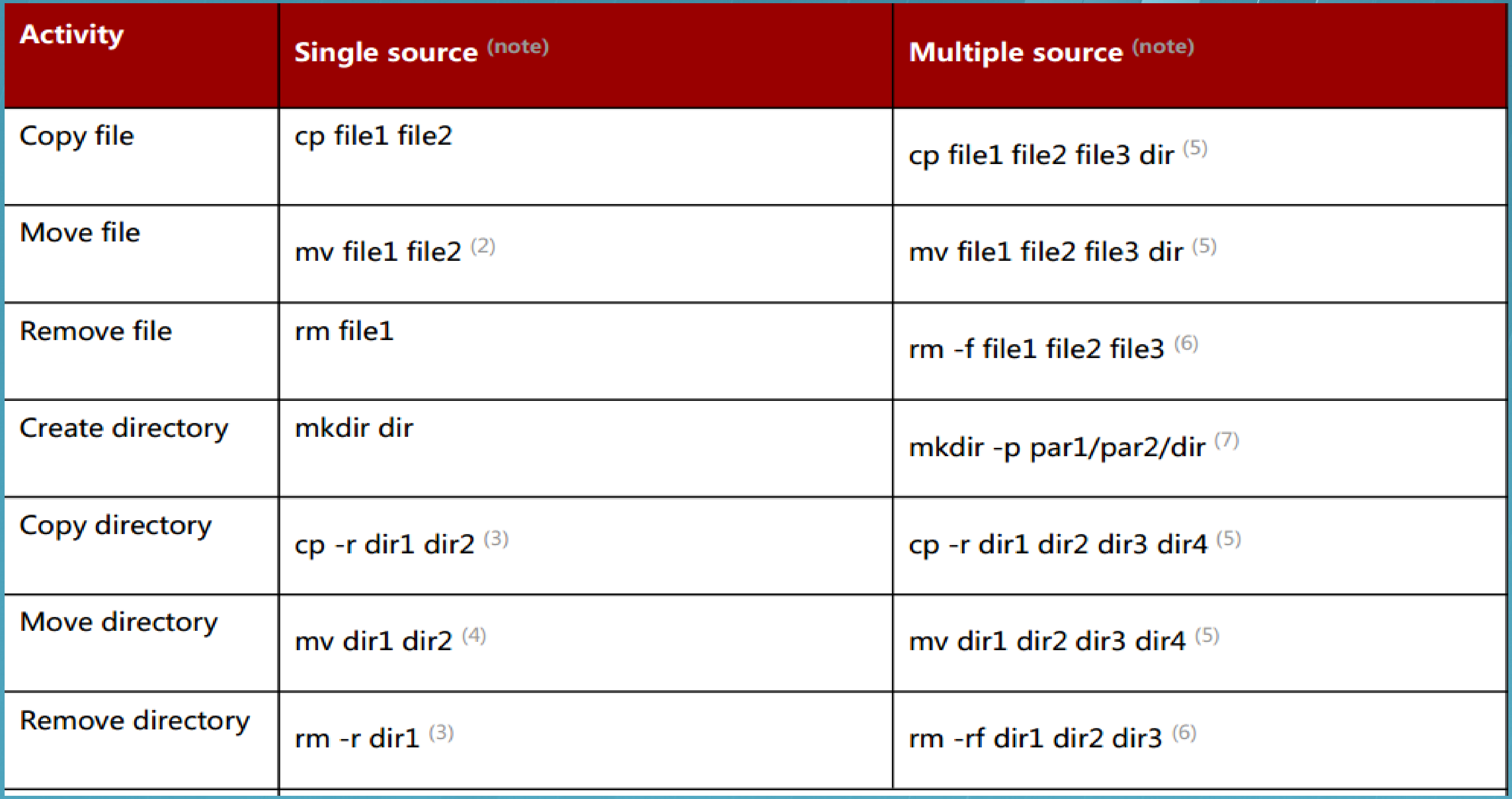
```
[student@desktopX ~]$ cd Videos
[student@desktopX Videos]$ pwd
/home/student/Videos
[student@desktopX Videos]$ cd /home/student/Documents
[student@desktopX Documents]$ pwd
/home/student/Documents
[student@desktopX Documents]$ cd -
[student@desktopX Videos]$ pwd
/home/student/Videos
[student@desktopX Videos]$ cd -
[student@desktopX Documents]$ pwd
[student@desktopX Documents]$ cd -
[student@desktopX Videos]$ pwd
/home/student/Videos
[student@desktopX Videos]$ cd
```

Example 3-2

```
[student@desktopX Videos]$ pwd
/home/student/Videos
[student@desktopX Videos]$ cd .
[student@desktopX Videos]$ pwd
/home/student/Videos
[student@desktopX Videos]$ cd ..
[student@desktopX ~]$ pwd
/home/student
[student@desktopX ~]$ cd ..
[student@desktopX home]$ pwd
/home
[student@desktopX home]$ cd ..
[student@desktopX /]$ pwd
/
[student@desktopX /]$ cd
[student@desktopX ~]$ pwd
/home/student
```

Managing Files Using Command-Line Tools

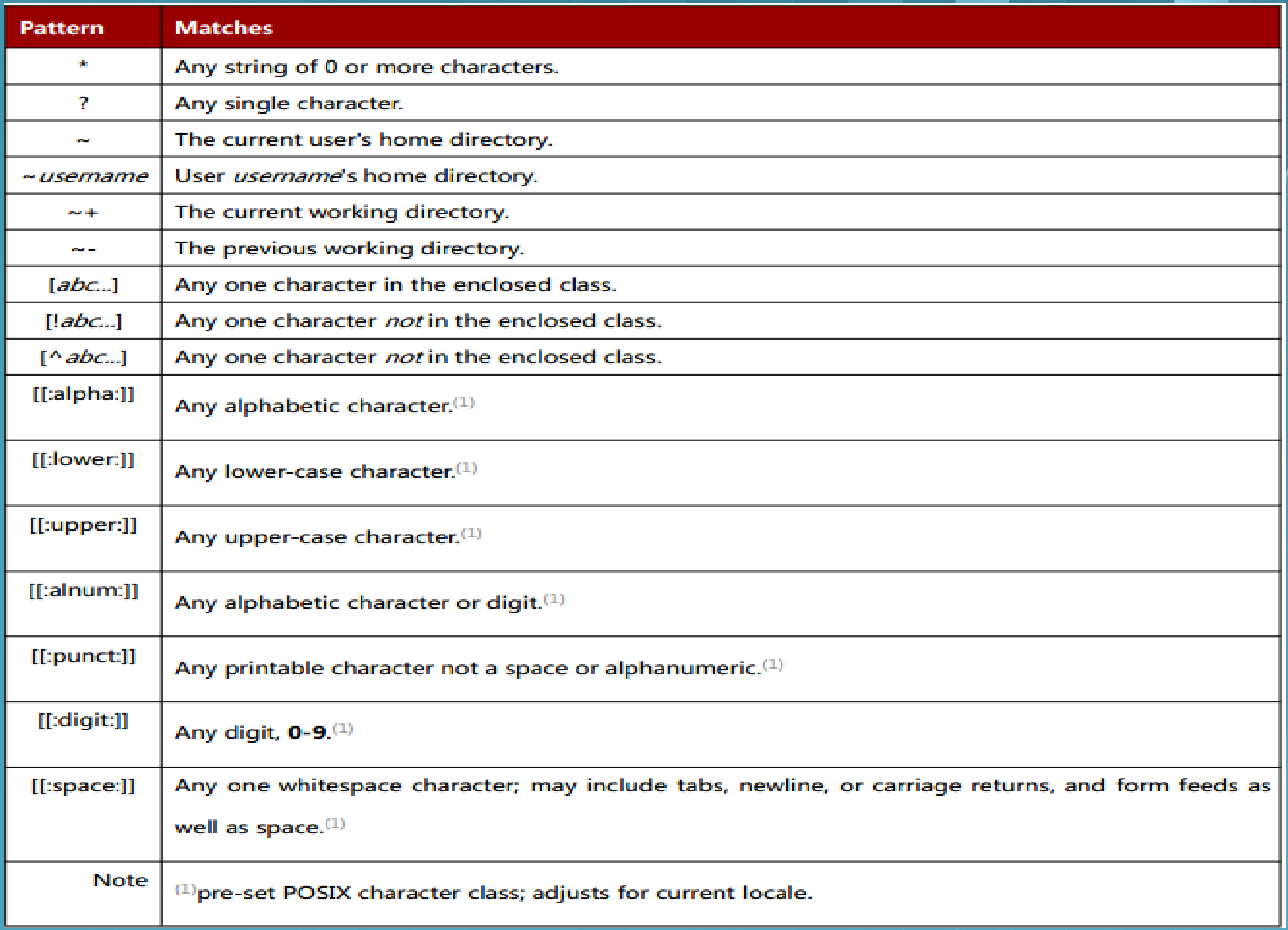
- Command-line file management



Activity	Single source (note)	Multiple source (note)
Copy file	cp file1 file2	cp file1 file2 file3 dir (5)
Move file	mv file1 file2 (2)	mv file1 file2 file3 dir (5)
Remove file	rm file1	rm -f file1 file2 file3 (6)
Create directory	mkdir dir	mkdir -p par1/par2/dir (7)
Copy directory	cp -r dir1 dir2 (3)	cp -r dir1 dir2 dir3 dir4 (5)
Move directory	mv dir1 dir2 (4)	mv dir1 dir2 dir3 dir4 (5)
Remove directory	rm -r dir1 (3)	rm -rf dir1 dir2 dir3 (6)

Matching File Names Using Path Name Expansion

Pattern matching



Pattern	Matches
*	Any string of 0 or more characters.
?	Any single character.
~	The current user's home directory.
~*username*	User *username*'s home directory.
~+	The current working directory.
~-	The previous working directory.
[*abc*...]	Any one character in the enclosed class.
[!*abc*...]	Any one character *not* in the enclosed class.
[^*abc*...]	Any one character *not* in the enclosed class.
[[:alpha:]]	Any alphabetic character.(1)
[[:lower:]]	Any lower-case character.(1)
[[:upper:]]	Any upper-case character.(1)
[[:alnum:]]	Any alphabetic character or digit.(1)
[[:punct:]]	Any printable character not a space or alphanumeric.(1)
[[:digit:]]	Any digit, **0-9**.(1)
[[:space:]]	Any one whitespace character; may include tabs, newline, or carriage returns, and form feeds as well as space.(1)
Note	(1)pre-set POSIX character class; adjusts for current locale.

Example 1

```
[student@desktopX ~]$ mkdir glob; cd glob
[student@desktopX glob]$ touch alfa bravo charlie delta echo able baker cast
dog easy
[student@desktopX glob]$ ls
able alfa baker bravo cast charlie delta dog easy echo
[student@desktopX glob]$ ls a*
able alfa
[student@desktopX glob]$ ls *a*
able alfa baker bravo cast charlie delta easy
[student@desktopX glob]$ ls [ac]*
able alfa cast charlie
[student@desktopX glob]$ ls ????
able alfa cast easy echo
[student@desktopX glob]$ ls ?????
baker bravo delta
```

Example 2

```
[student@desktopX glob]$ ls ~/glob
able alfa baker bravo cast charlie delta dog easy echo
[student@desktopX glob]$ echo ~/glob
/home/student/glob
[student@desktopX glob]$ echo {Sunday,Monday,Tuesday,Wednesday}.log
Sunday.log Monday.log Tuesday.log Wednesday.log
[student@desktopX glob]$ echo file{1..3}.txt
file1.txt file2.txt file3.txt
[student@desktopX glob]$ echo file{a..c}.txt
filea.txt fileb.txt filec.txt
[student@desktopX glob]$ echo file{a,b}{1,2}.txt
filea1.txt filea2.txt fileb1.txt fileb2.txt
[student@desktopX glob]$ echo file{a{1,2},b,c}.txt
filea1.txt filea2.txt fileb.txt filec.txt
```

Example 3

```
[student@desktopX glob]$ echo Today is `date +%A`.
Today is Wednesday.
[student@desktopX glob]$ echo The time is $(date +%M) minutes past $(date +%l%p).
The time is 26 minutes past 11AM.
[student@desktopX glob]$ host=$(hostname); echo $host
desktopX
[student@desktopX glob]$ echo "***** hostname is ${host} *****"
***** hostname is desktopX *****
[student@desktopX glob]$ echo Your username variable is \$USER.
Your username variable is $USER.
[student@desktopX glob]$ echo "Will variable $host evaluate to $
(hostname)?"
Will variable desktopX evaluate to desktopX?
[student@desktopX glob]$ echo 'Will variable $host evaluate to $(hostname)?'
Will variable $host evaluate to $(hostname)?
```

Lab-1

<Lab1>
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.
<Lab2>
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
<Lab3>
To organize the TV episodes, create two subdirectories named "season1" and "season2" under the existing "Videos" directory. Use one command.

Lab-2

<Lab4>
Move the appropriate TV episodes into the season subdirectories. Use only two commands,specifying destinations using relative syntax.
<Lab5>
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
<Lab6>
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
<Lab7>
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?

Lab-3

<Lab8>
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
<Lab9>
Chapters 7 and 8 will be written while on vacation. Move the files form "chapters" to "vacation". Use one command without wildcard characters.
<Lab10>
With one command, change the working directory to the season 2 TV episodes location, then link (not copy) the first episode of the season to the "vacation" directory.
<Lab11>
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. Link the episode 2 file into "vacation". Return to "vacation" using the shortcut again

Lab-4

<Lab12>
Chapters 5 and 6 may need a plot change. To prevent these changes from modifying original files, copy, not link, both files into "plot_change". Move up one directory to "vacation's" parent directory, then use one command from there.
<Lab13>
To track changes, save each version as a new file name to include the date. Change to the "plot_change" directory. Copy "mystery_chapter5.odf", appending the current timestamp (as the number of seconds since the epoch) to ensure a unique file name. See the solution for syntax. Also make a copy appending the current user ($USER) to the file name.
<Lab14>
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.

Lab-5

When the "vacation" is over, the vacation directory is no longer needed. Delete it using the rm command with the recursive option. Confirm that the original files still remain after their linked versions have been removed.

When finished, return to the home directory

That's all