Note the use of the --local option that makes check50 a little faster. Also note the path to the checks ends with lab2, not lab1.

Start Recording Your Commands

Run **script -a**to record the commands that are typed in to the prompt. Run the following

% script -a

The "**-a**" at the end of the command above is an option which tells script to append new commands to the current typescript file. ***If you do not run script with the -a option, the typescript file will be overwritten! This means that you will have to redo the lab, so make sure to use the -a option with the script command. It doesn't hurt to simply always use -a.***

2.1 Copying Files

**cp file1 file2** is the command which makes a copy of **file1** in the current working directory and calls it **file2**

First, cd to your unixstuff directory.

% cd ~/unixstuff

We will see some file commands in this lab, so we need a file. Use the following command to download a file that contains a bunch of cheesy jokes.

% wget https://raw.githubusercontent.com/uwrf-csis/csis248/main/lab2/jokes.txt

**wget** is a very powerful tool that lets you download download files from command line amongst other things.

Use the ls command to make sure your commands worked. You should see a jokes.txt file in your unixstuff directory.

Create a backup of your **jokes.txt** file by copying it to a file called **jokes.bak**

% cp jokes.txt jokes.bak

2.2 Moving files

**mv file1 file2** moves (or renames) **file1** to **file2**

To move a file from one place to another, use the mv command. This has the effect of moving rather than copying the file, so you end up with only one file rather than two.

It can also be used to rename a file, by moving the file to the same directory, but giving it a different name.

We are now going to move the file jokes.bak to your backup directory. If you do not have a directory called "backups", create one.

Then, inside the **unixstuff** directory, type

% mv jokes.bak backups/.

Type ls and ls backups to see if it has worked.

2.3 Removing files and directories

rm (remove), rmdir (remove directory)

To delete (remove) a file, use the rm command. As an example, we are going to create a copy of the **jokes.txt** file then delete it.

Inside your **unixstuff** directory, type

% cp jokes.txt tempfile.txt  
% ls (to check if it has created the file)  
% rm tempfile.txt  
% ls (to check if it has deleted the file)

You can use the rmdir command to remove a directory (make sure it is empty first). Try to remove the **backups** directory. You will not be able to since UNIX will not let you remove a non-empty directory.

Exercise 2b

Create a directory called **tempstuff** using mkdir , then remove it using the rmdir command.

2.4 Displaying the contents of a file on the screen

clear (clear screen)

Before you start the next section, you may like to clear the terminal window of the previous commands so the output of the following commands can be clearly understood.

At the prompt, type

% clear

This will clear all text and leave you with the % prompt at the top of the window. You can also use the keys [Ctrl-L] to clear the screen.

cat (concatenate)

The command cat can be used to display the contents of a file on the screen. Type:

% cat jokes.txt

As you can see, the file is longer than than the size of the window, so it scrolls past making it unreadable.

less

The command less writes the contents of a file onto the screen a page at a time. Type

% less jokes.txt

Press the [space-bar] if you want to see another page, type [q] if you want to quit reading. As you can see, less is used in preference to cat for long files.

head

The head command writes the first ten lines of a file to the screen.

First clear the screen then type

% head jokes.txt

Then type

% head -5 jokes.txt

What difference did the -5 do to the head command?

tail

The tail command writes the last ten lines of a file to the screen.

Clear the screen and type

% tail jokes.txt

Tail shows the last 10 lines by default. How can you view the last 15 lines of the file? The man pages might help.

2.5 Searching the contents of a file

Simple searching using less

Using less, you can search though a text file for a keyword (pattern). For example, to search through **jokes.txt** for the word 'science', type

% less jokes.txt

then, still in less (i.e. don't press [q] to quit), type a forward slash [/] followed by the word to search

/walk

As you can see, less finds and highlights the keyword. Type [n] to search for the next occurrence of the word.

grep

grep is one of many standard UNIX utilities. It searches files for specified words or patterns. First clear the screen, then type

% grep Walk jokes.txt

As you can see, grep has printed out each line containing the word science.

Or has it????

Try typing

% grep walk jokes.txt

The grep command is case sensitive; it distinguishes between Walk and walk.

To ignore upper/lower case distinctions, use the -i option, i.e. type

% grep -i Walk jokes.txt

To search for a phrase or pattern, you must enclose it in single quotes (the apostrophe symbol). For example to search for spinning top, type

% grep -i 'walks into' jokes.txt

Some of the other options of grep are:

-v display those lines that do NOT match  
-n precede each maching line with the line number  
-c print only the total count of matched lines

Try some of them and see the different results. Don't forget, you can use more than one option at a time, for example, the number of lines without the words walk or Walk is

% grep -ivc walk jokes.txt

wc (word count)

A handy little utility is the wc command, short for word count. To do a word count on **jokes.txt**, type

% wc -w jokes.txt

To find out how many lines the file has, type

% wc -l jokes.txt

2.6 Getting Help

On-line Manuals

Examples for how to use just about any command can be found online using a search engine. StackOverflow usually has good examples for how to use specific commands.

There are on-line manuals which gives information about most commands. The manual pages tell you which options a particular command can take, and how each option modifies the behavior of the command. Type man command to read the manual page for a particular command.

For example, to find out more about the wc (word count) command, type

% man wc

Alternatively

% whatis wc

gives a one-line description of the command, but omits any information about options etc.

Apropos

When you are not sure of the exact name of a command,

% apropos keyword

will give you the commands with keyword in their manual page header. For example, try typing

% apropos copy

2.7 Submit Your Work

Stop the script command by typing

% exit

Then type

% check50 uwrf-csis/csis248/main/lab2

Make sure all checks pass. If they do not, then you probably skipped a step in the lab. The checks show which commands are missing. Rerun the **script -a** command and finish the lab and verify that all checks pass. Don't forget to use the **exit** command to stop the script command.

Finally, submit your work by typing the following command and answering the question that follows.

% submit50 uwrf-csis/csis248/main/lab2

Summary

|  |  |
| --- | --- |
| cp file1 file2 | copy file1 and call it file2 |
| mv file1 file2 | move or rename file1 to file2 |
| rm file | remove a file |
| rmdir directory | remove a directory |
| cat file | display a file |
| more file | display a file a page at a time |
| head file | display the first few lines of a file |
| tail file | display the last few lines of a file |
| grep 'keyword' file | search a file for keywords |
| wc file | count number of lines/words/characters in file |
| man command | read the online manual page for a command |
| whatis command | brief description of a command |
| apropos keyword | match commands with keyword in their man pages |