Amrita School of Engineering, Chennai Campus..

19CSE101: Computer Systems Essentials

**LAB SHEET - Operating System**

**Unix Command Familiarization**

Section 1. man pages

This section will explain the use of man pages (also called manual pages) on your Unix or Linux computer. You will learn the man command together with related command whatis. Mostc10files and commands have. pretty good man pages to explain their use. Man pages also come in handy when you are using multiple flavours of Unix or several Linux distributions since options and parameters sometimes vary.

man commandp cd test

Type man followed by a command (for which you want help) and start reading. Press q to quit the man page. Some man pages contain examples (near the end).

whatis

To see just the description of a manual page, use whatis followed by a string.



Section 2: working with directories

This module is a brief overview of the most common commands to work with directories: pwd, cd, ls, mkdir and rmdir. These commands are available on any Linux (or Unix) system.

pwd

Pwd command gives the full path or address of the current working directory. Go ahead, try it: Open a command line interface (also called a terminal, console) and type pwd. The tool displays your current directory.



cd

You can change your current directory with the cd command (Change Directory).



cd ~

The cd is also a shortcut to get back into your home directory. Just typing cd without a target directory, will put you in your home directory. Typing cd ~ has the same effect.



cd ..

To go to the parent directory (the one just above your current directory in the directory tree), type cd .. .



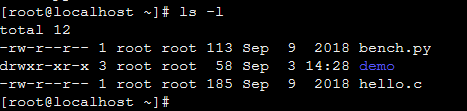
ls

You can list the contents of a directory with ls



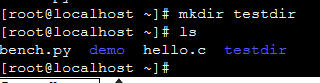
ls –l

Many times you will be using options with ls to display the contents of the directory in different formats or to display different parts of the directory. Typing just ls gives you a list of files in the directory. Typing ls -l (that is a letter L, not the number 1) gives you along listing. Go to the man page of ls and find out the different options of ls.



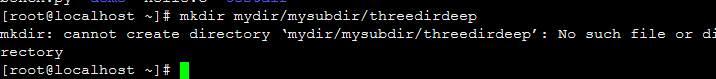
mkdir

Walking around the Unix file tree is fun, but it is even more fun to create your own directories with mkdir. You have to give at least one parameter to mkdir, the name of the new directory to be created.

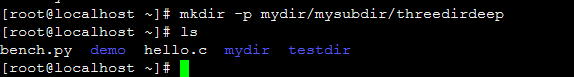


mkdir –p

The following command will fail, because the parent directory of three dirs. deep does not exist.



When given the option -p, mkdir will create parent directories as needed.



rmdir

When a directory is empty, you can use rmdir to remove the directory

rmdir –p

And similar to the mkdir -p option, you can use rmdir to recursively remove directories

Section 3: working with files

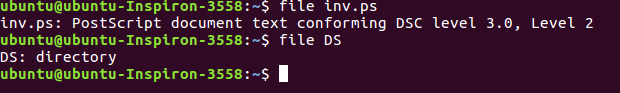
In this section we learn how to recognise, create, remove, copy and move files using commands like file, touch, rm, cp, mv and rename.

All files are case sensitive

Files on Linux (or any Unix) are case sensitive. This means that FILE1 is different from file1, and /etc/hosts is different from /etc/Hosts (the latter one does not exist on a typical Linux computer).

file

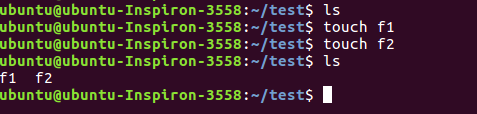
The file utility determines the file type. Linux does not use extensions to determine the file type. The command line does not care whether a file ends in .txt or .pdf. As a system administrator, you should use the file command to determine the file type.



touch create an empty file

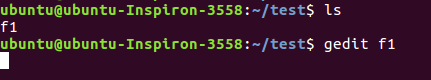
One easy way mkdir dawn to create an empty file is with touch. This screenshot starts with an empty directory, creates two files with touch and the lists

those files.



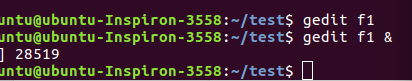
gedit

It is the general-purpose text editor. It is easy to use, with a clean and simple [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface). It includes tools for editing [source code](https://en.wikipedia.org/wiki/Source_code).



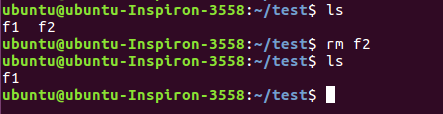
gedit can be used to edit an existing file or if the file with name f1 is not present then gedit will create a new file with name f1 and allows us to edit it. But gedit will not allow us to return to the terminal until its execution is completed.( Window closed)

gedit with & will allow the terminal and gedit window to execute simultaneously.



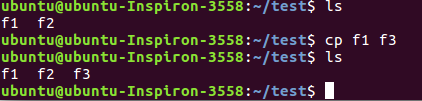
rm remove forever

When you no longer need a file, use rm to remove it. Unlike some graphical user interfaces, the command line in general does not have a waste bin or trash can to recover files. When you use rm to remove a file, the file is gone. Therefore, be careful when removing files!



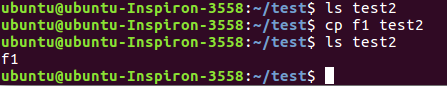
cp

copy one file. To copy a file, use cp with a source and a target argument.



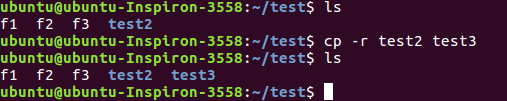
copy to another directory

If the target is a directory, then the source files are copied to that target directory.



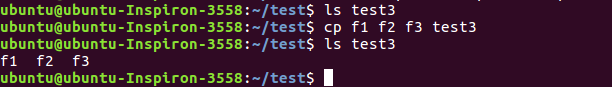
cp -r

To copy complete directories, use cp -r (the -r option forces recursive copying of all files in all subdirectories).



copy multiple files to directory

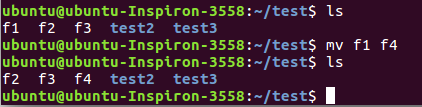
You can also use cp to copy multiple files into a directory. In this case, the last argument must be a directory.

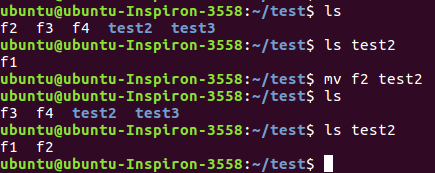


mv

rename files with mv

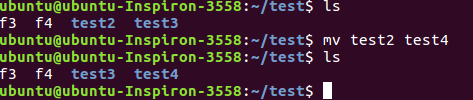
Use mv to rename a file or to move the file to another directory.





rename directories with mv

The same mv command can be used to rename directories.

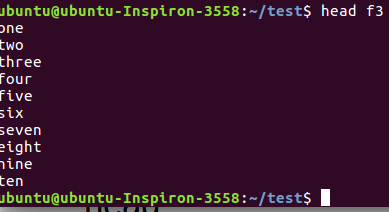


Section 4 working with file contents

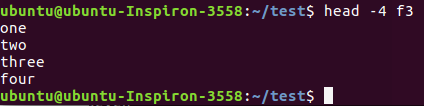
In this section we will look at the contents of text files with head, tail, cat, tac, more, less and strings. We will also get a glimpse of the possibilities of tools like cat on the command line.

head

You can use head to display the first ten lines of a file.

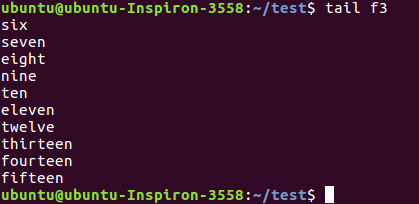


The head command can also display the first n lines of a file.



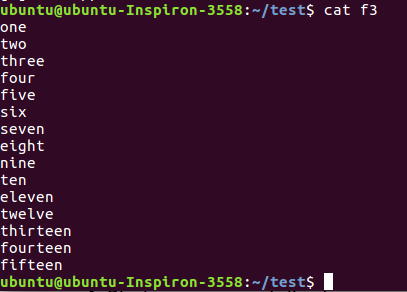
tail

Similar to head, the tail command will display the last ten lines of a file.



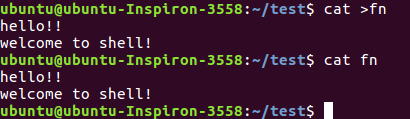
cat

The cat command is one of the most universal tools, yet all it does is copy standard input to standard output. In combination with the shell this can be very powerful and diverse. Some examples will give a glimpse into the possibilities. The first example is simple, you can use cat to display a file on the screen or terminal. If the file is longer than the screen, it will scroll to the end.



create files

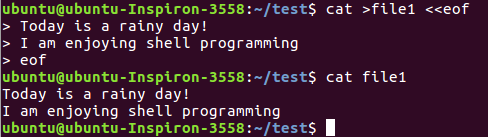
You can use cat to create flat text files. Type the cat > fn command as shown in the screenshot below. Then type one or more lines, finishing each line with the enter key. After the last line, type and hold the Control (Ctrl) key and press d.



The Ctrl d key combination will send an EOF (End of File) to the running process ending the cat command.

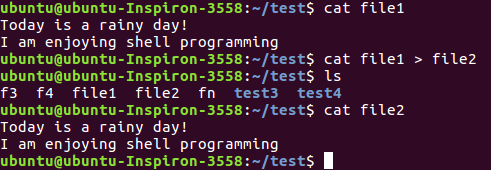
custom end marker

You can choose an end marker for cat with << as is shown in this screenshot. This construction is called a here directive and will end the cat command.



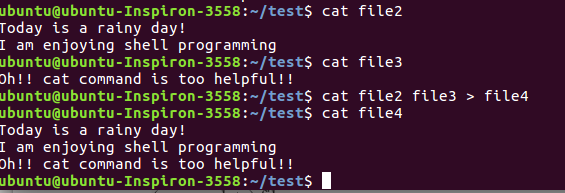
copy files

In the third example you will see that cat can be used to copy files.



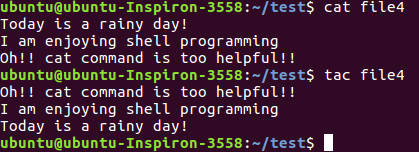
concatenate

cat is short for concatenate. One of the basic uses of cat is to concatenate files into a bigger (or complete) file.



tac

This example will show you the purpose of tac (cat backwards).



Lab Exercise

1. display the path of your current directory

Pwd

1. Make a new directory named main.

Mkdir main

1. Now change to the directory main.

Cd main

4. Make the directories in the following hierarchy using a single command.

Dir1 à Dir 2 àDir3

Mkdir -p Dir1/Dir2/Dir3/

Cd~/Dir1/Dir2/Dir3/

1. Print the path of the current directory.

pwd

1. Go to Dir3 using a single command.

Cd dir3

7. Create a new file **demo1**, type and save the following contents,

This is my first file in shell.

I can edit this file!!!

Gedit demo1 and this is my first file in the shell.I can edit this file!!  
  
8. Create a new file **demo2**, type and save the following contents,

Hi!!! This is the second file.

I am doing shell commands.

I can edit this file!!!cd

Gedit demo2 hi!!! This is the second file.I am doing shell commands.I can edit this file.

9. Display the contents of file **demo1** in terminal.

ls demo1

10. List the files and folders present in Dir3.

Ls Dir3

11. Go to Dir 2.

Cd Dir2

12. Go to your home directory.

pwd

13. Stay where you are, and list the contents of Dir3.

Ls Dir3

14. List all the files (including hidden files) in your home directory.

Ls -l

15. Create a new file **test1**, type and save the contents into your file.

I am working with linux shell.

Good bye

Touch test1 and I am working with linux shell .Good bye

16. Copy the contents of **test1** to **test2** in the same directory.

Cp test1 test2

17. Rename **test2** as **test3**.

mv test 2 test3

18. Determine the file type of **test3**.

File -b test3

19. Move the file **test3** to the directory Dir3.

20. Create a file **count**, with content one to twenty in words with one line having only one number using a single command.

Cat>count

One

Two

Three

Four

Five

Six

Seven

Eight

Nine

Ten

Eleven

Twelve

Thirteen

Fourteen

Fifteen

Sixteen

Seventeen

Eighteen

Nineteen

twenty

21. Copy the file **count** to **count2** using cat command.

Cat count>count2

22. Create another file **count3** with numbers twenty one to twenty five (in five lines).

Cat>count3

Twentyone

Twentytwo

Twentythree

Twentyfour

twentyfive

23. Concatenate the contents of files **count2** and **count3** and write it into the file **countfinal**.

Cat count2 count3>countfinal

24. Remove the files **demo1** and **demo2** in directory Dir3.

Rm main/Dir1/Dir2/Dir3/demo1

Rm main/Dir1/Dir2/Dir3/demo2

25.Go to Dir2 and remove the subdirectory Dir3.

Rm -rf dir3

26. Come back to your home folder and remove Dir2.

Cd

Rm-rf main/Dir1/Dir2/

27. Display first 10 lines of the file **countfinal** in terminal.

Head countfinal

28. Display last 10 lines of the file **countfinal** in terminal.

Tail countfinal

29. Display first 5 lines of the file **countfinal** in terminal.

Head – 5 countfinal

30. Display last 4 lines of the file **countfinal** in terminal.

Tail -4 countfinal

31. Display the contents of the file **countfinal** in the inverted form.(last line first and first line last)

Tac countfinal

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This is my first file in shell.