UNICS and LINUX commands (7 Chapters)

-Shriram K Vasudevan

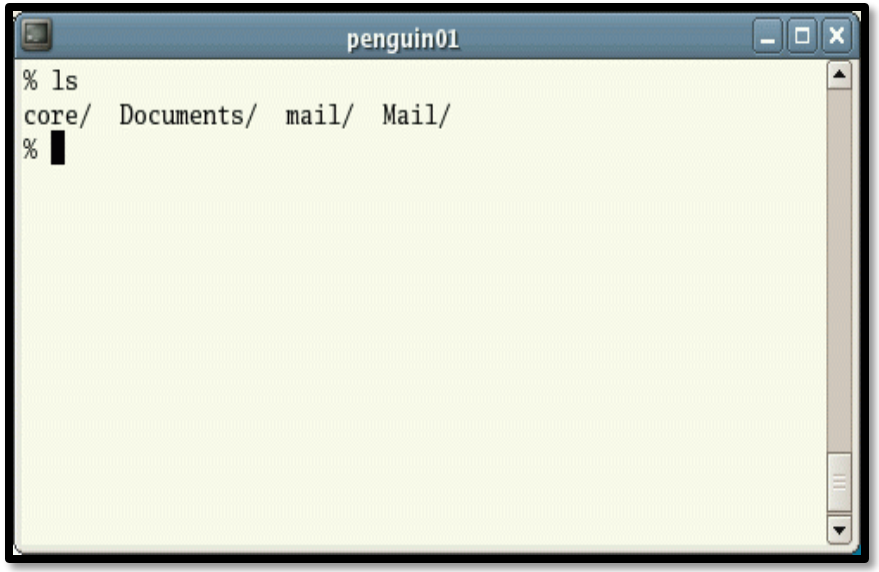
-SHRIRAMKV@GMAIL.COM

Chapter 1
Listing files and directories

- **ls (list)**
- When you first login, your current working directory is your home directory.
- Your home directory has the same name as your user-name, for example, **shriram**, and it is where your personal files and subdirectories are saved.
- To find out what is in your home directory, type

  % ls

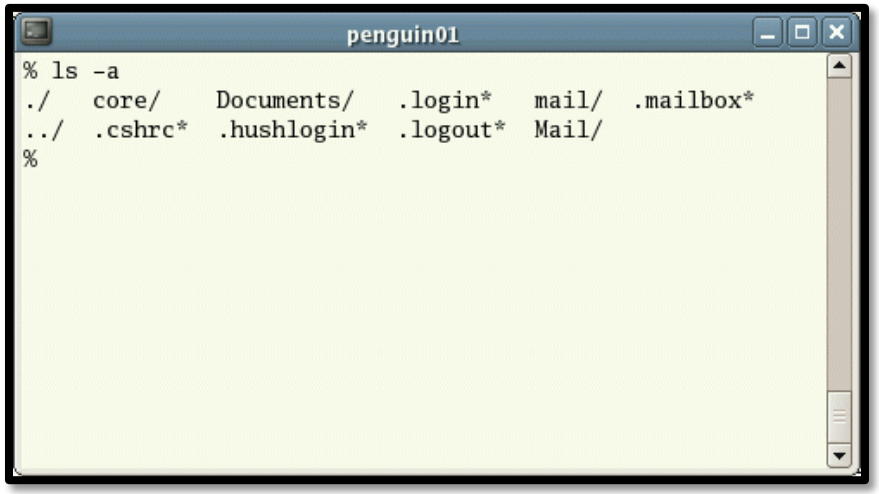
- The ls command (lowercase L and lowercase S) lists the contents of your current working directory.



Listing files and directories

- ls does not, in fact, cause all the files in your home directory to be listed, but only those ones whose name does not begin with a dot (.) Files beginning with a dot (.) are known as hidden files and usually contain important program configuration information.
- They are hidden because you should not change them unless you are very familiar with UNIX!!!
- To list all files in your home directory including those whose names begin with a dot, type

  % ls -a
- As you can see, ls -a lists files that are normally hidden.



Making Directories

- **mkdir (make directory)**
- We will now make a subdirectory in your home directory to hold the files you will be creating and using in the course of this tutorial. To make a subdirectory called unixstuff in your current working directory type

  % mkdir unixstuff

- To see the directory you have just created, type

  % ls

```
Creating a Single Directory
ubuntu@ubuntu:~$ mkdir testdir
ubuntu@ubuntu:~$ ls -lrt
total 8
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Videos
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Public
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Music
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Documents
drwxr-xr-x 2 ubuntu ubuntu 80 2009-09-28 12:16 Desktop
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file1.txt
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file2.txt
-rw-r--r-- 1 ubuntu ubuntu  0 2009-09-28 12:48 file3.txt
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:49 testdir
ubuntu@ubuntu:~$
```

Making Directories

```
to create multiple directory
mkdir d1 d2 d3 d4 d5
ubuntu@ubuntu:~$ mkdir d1 d2 d3 d4 d5
ubuntu@ubuntu:~$ ls -lrt
total 8
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Videos
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Public
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Music
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Documents
drwxr-xr-x 2 ubuntu ubuntu 80 2009-09-28 12:16 Desktop
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file1.txt
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file2.txt
-rw-r--r-- 1 ubuntu ubuntu  0 2009-09-28 12:48 file3.txt
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:49 testdir
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:51 d5
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:51 d4
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:51 d3
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:51 d2
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:51 d1
```

Making Directories – Little Play

To create nested directories

```
ubuntu@ubuntu:~$ mkdir -p l1/l2/l3/l4
ubuntu@ubuntu:~$ ls -lrt
total 0
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:25 Videos
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:25 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:25 Public
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:25 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:25 Music
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:25 Documents
drwxr-xr-x 2 ubuntu ubuntu 80 2009-09-28 17:26 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:30 d5
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:30 d4
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:30 d3
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 17:30 d2
drwxr-xr-x 3 ubuntu ubuntu 60 2009-09-28 17:31 d1
drwxr-xr-x 3 ubuntu ubuntu 60 2009-09-28 17:32 l1
ubuntu@ubuntu:~$
```

To see the tree structure

```
ubuntu@ubuntu:~$ ls -R d1
d1:
d2

d1/d2:
d3

d1/d2/d3:
d4

d1/d2/d3/d4:
ubuntu@ubuntu:~$
```

Change Directory (cd)

To change the current working directory

```
ubuntu@ubuntu:~$ cd l1
ubuntu@ubuntu:~/l1$ pwd
/home/ubuntu/l1
```

To reach the Home directory again.

```
ubuntu@ubuntu:~$ cd l1
ubuntu@ubuntu:~/l1$ pwd
/home/ubuntu/l1
ubuntu@ubuntu:~/l1$ cd
ubuntu@ubuntu:~$ pwd
/home/ubuntu
ubuntu@ubuntu:~$
```

Cd ~ will also take you to your home directory ☺

The parent directory (..)

- (**..**) means the parent of the current directory, so typing

  % cd ..

- will take you one directory up the hierarchy (back to your home directory). Try it now.

pwd (print working directory)

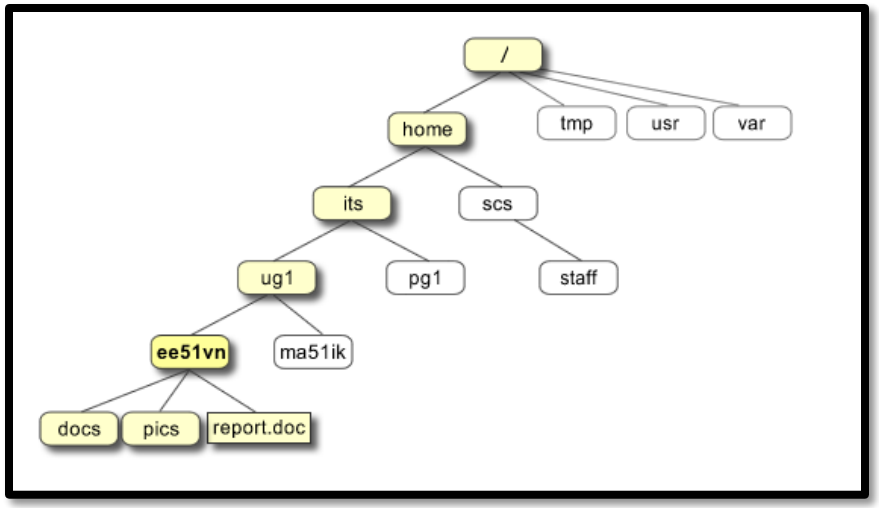
- Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory, type cd to get back to your home-directory and then type

  % pwd

- The full pathname will look something like this -

  /home/its/ug1/ee51vn

- which means that **ee51vn** (your home directory) is in the sub-directory **ug1** (the group directory),which in turn is located in the **its** sub-directory, which is in the **home** sub-directory, which is in the top-level root directory called " / " .



- NOTE: “/” is called as mount point ☺

Here you →

Command	Meaning
ls	list files and directories
ls -a	list all files and directories
mkdir	make a directory
cd *directory*	change to named directory
cd	change to home-directory
cd ~	change to home-directory
cd ..	change to parent directory
pwd	display the path of the current directory

Chapter 2
Copying Files

To copy files

cp file1.txt file2.txt

To copy folder
ubuntu@ubuntu:~$ cp -r d2 /home/ubuntu/d3
(This will copy a directory from Source to destination)

To rename the files
ubuntu@ubuntu:~$ mv file1.txt file2.txt
(File 1 will be moved as file2)

Removing Files and Directories

```
To remove an empty directory
(Directory with files cant be removed without removing the files in the dir)
ubuntu@ubuntu:~$ rmdir dd

To remove the directory which is not empty
ubuntu@ubuntu:~$ rm -rf l1
```

- To delete (remove) a file, use the rm command.
  % rm tempfile.txt

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.

Let me catch the CAT's HEAD and TAIL

cat (concatenate)

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

  % cat science.txt

- If the file is longer 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 science.txt

- Press the [**space-bar**] if you want to see another page, and 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 science.txt

  Then type

  % head -5 science.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 science.txt

ANSWER THIS NOW:

- Q. How can you view the last 15 lines of the file?

The CAT has grown bigger

```
To see a file content
cat filename

To append to a file
ubuntu@ubuntu:~$ cat >>file1.txt
THIS IS SHRIRAM
```

```
Copy onefile's content to another file.
ubuntu@ubuntu:~$  cat file1.txt >> file2.txt
ubuntu@ubuntu:~$ ls -lrt
total 8
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Videos
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Public
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Music
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Documents
drwxr-xr-x 2 ubuntu ubuntu 80 2009-09-28 12:16 Desktop
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file1.txt
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file2.txt
```

```
Creation of file using touch
ubuntu@ubuntu:~$ touch file3.txt
ubuntu@ubuntu:~$ ls -lrt
total 8
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Videos
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Public
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Music
drwxr-xr-x 2 ubuntu ubuntu 40 2009-09-28 12:16 Documents
drwxr-xr-x 2 ubuntu ubuntu 80 2009-09-28 12:16 Desktop
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file1.txt
-rw-r--r-- 1 ubuntu ubuntu 17 2009-09-28 12:45 file2.txt
-rw-r--r-- 1 ubuntu ubuntu  0 2009-09-28 12:48 file3.txt
```

Lets Count - WC

wc (word count)

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

  % wc -w science.txt

- To find out how many lines the file has, type

  % wc -l science.txt

- grep () - "global regular expression printer"
- grep is one of many standard UNIX utilities. It searches files for specified words or patterns. First clear the screen, then type

  % grep science science.txt

Here you →

Command	Meaning
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
less ***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

Chapter – 3
Re direction

- Most processes initiated by UNIX commands write to the standard output (that is, they write to the terminal screen), and many take their input from the standard input (that is, they read it from the keyboard). There is also the standard error, where processes write their error messages, by default, to the terminal screen.
- We have already seen one use of the cat command to write the contents of a file to the screen.

- Now type cat without specifying a file to read

  % cat
- Then type a few words on the keyboard and press the [**Return**] key.
- Finally hold the [**Ctrl**] key down and press [**d**] (written as **^D** for short) to end the input.
- What has happened?
- If you run the cat command without specifying a file to read, it reads the standard input (the keyboard), and on receiving the 'end of file' (**^D**), copies it to the standard output (the screen).
- In UNIX, we can redirect both the input and the output of commands.

Re - direction

- We will now use the cat command to join (concatenate) **list1** and **list2** into a new file called **biglist**. Type

  % cat list1 list2 > biglist

- What this is doing is reading the contents of **list1** and **list2** in turn, then outputing the text to the file **biglist**
- To read the contents of the new file, type % cat biglist

Let the water flow - PIPES

- To see who is on the system with you, type

  % who

- One method to get a sorted list of names is to type,

  % who > names.txt
  % sort < names.txt

- What you really want to do is connect the output of the who command directly to the input of the sort command. This is exactly what pipes do. The symbol for a pipe is the vertical bar |
- For example, typing

  % who | sort

  will give the same result as above, but quicker and cleaner.
- To find out how many users are logged on, type

  % who | wc -l

Contd.,

Syntax: *command1 | command2*

Command using Pips	Meaning or Use of Pipes
$ ls | more	Here the output of ls command is given as input to more command So that output is printed one screen full page at a time
$ who | sort	Here output of who command is given as input to sort command So that it will print sorted list of users
$ who | wc -l	Here output of who command is given as input to wc command So that it will number of user who logon to system
$ ls -l | wc -l	Here output of ls command is given as input to wc command So that it will print number of files in current directory.
$ who | grep raju	Here output of who command is given as input to grep command So that it will print if particular user name if he is logon or nothing is printed (To see for particular user logon)

Here you →

Command	Meaning
command > ***file***	redirect standard output to a file
command >> ***file***	append standard output to a file
command < ***file***	redirect standard input from a file
command1 | ***command2***	pipe the output of command1 to the input of command2
cat ***file1 file2*** > ***file0***	concatenate file1 and file2 to file0
sort	sort data
who	list users currently logged in

Chapter 4
Few New Intros

To see the file types

```
ubuntu@ubuntu:~$ file *
d2:       directory
d3:       directory
d4:       directory
d5:       directory
Desktop:    directory
Documents:  directory
file.txt:   ASCII text
first.sh:   ASCII English text, with CRLF, LF line terminators
first.sh~:  ASCII English text, with CRLF, LF line terminators
h.sh:       ASCII text
h.sh~:      ASCII text
karthi.txt: ASCII text
Music:      directory
Pictures:   directory
Public:     directory
secon.sh:   ASCII English text
secon.sh~:  ASCII English text
Templates:  directory
Videos:     directory
```

To see the date

```
ubuntu@ubuntu:~$ date
Mon Sep 28 18:43:56 UTC 2009
ubuntu@ubuntu:~$
```

To display the calendarubuntu@ubuntu:~$ cal

```
   September 2009
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

To display all available groups

```
ubuntu@ubuntu:~$ groups
ubuntu adm dialout cdrom plugdev lpadmin admin sambashare
```

To view the location of a command

```
ubuntu@ubuntu:~$  which whoami
/usr/bin/whoami
ubuntu@ubuntu:~$ which who
/usr/bin/who
ubuntu@ubuntu:~$ which cat
/bin/cat
```

To get the calendar displayed for particular month

```
ubuntu@ubuntu:~$ cal 9 2009
   September 2009
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

Wildcards

- **The * wildcard**
- The character * is called a wildcard, and will match against none or more character(s) in a file (or directory) name. For example, in your directory, type

  % ls list*

- This will list all files in the current directory starting with **list....**
- Try typing

  % ls *list

- This will list all files in the current directory ending with **....list**

Wildcards

- **The ? wildcard**
- The character ? will match exactly one character.
  So **?ouse** will match files like **house** and **mouse**, but not **grouse**.
  Try typing

  % ls ?list

Filename conventions

Good filenames	Bad filenames
project.txt	project
my_big_program.c	my big program.c
fred_dave.doc	fred & dave.doc

MAN can HELP you to know WHATIS this! - **On-line Manuals**

- 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 behaviour 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.
- And u can do man man ☺

Are you appropriate?

- ***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 date

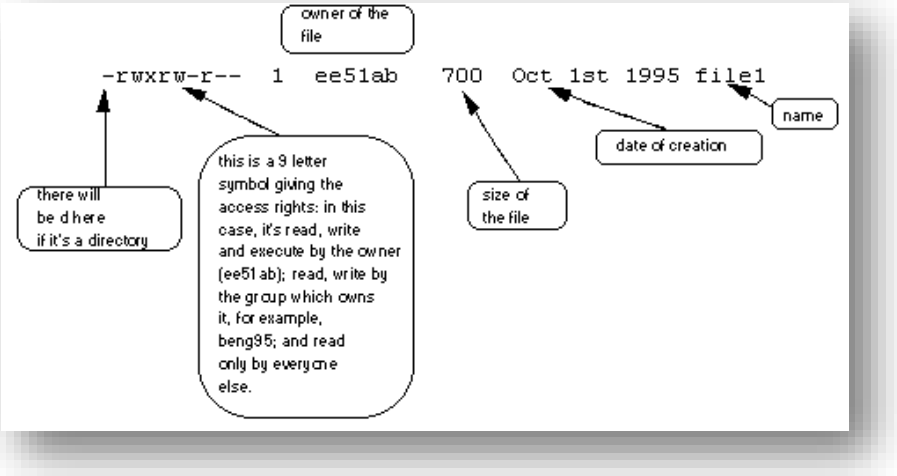
Here you →

Command	Meaning
*	match any number of characters
?	match one character
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

Chapter 5
File system security (access rights)

- In your directory, type

  % ls -l (l for long listing!)



File system security (access rights)

Access rights on files.

- r (or -), indicates read permission (or otherwise), that is, the presence or absence of permission to read and copy the file
- w (or -), indicates write permission (or otherwise), that is, the permission (or otherwise) to change a file
- x (or -), indicates execution permission (or otherwise), that is, the permission to execute a file, where appropriate

Access rights on directories.

- r allows users to list files in the directory;
- w means that users may delete files from the directory or move files into it;
- x means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files.

Changing access rights – Lets Change

- **chmod (changing a file mode)**
- Only the owner of a file can use chmod to change the permissions of a file. The options of chmod are as follows

Symbol	Meaning
u	user
g	group
o	other
a	all
r	read
w	write (and delete)
x	execute (and access directory)
+	add permission
-	take away permission

Changing access rights – Lets Change

- For example, to remove read write and execute permissions on the file **biglist** for the group and others, type

  % chmod go-rwx biglist

- This will leave the other permissions unaffected.
- To give read and write permissions on the file **biglist** to all,

  % chmod a+rw biglist

Processes and Jobs

- A process is an executing program identified by a unique PID (process identifier). To see information about your processes, with their associated PID and status, type

  % ps

- A process may be in the foreground, in the background, or be suspended. In general the shell does not return the UNIX prompt until the current process has finished executing.
- Some processes take a long time to run and hold up the terminal.
- Backgrounding a long process has the effect that the UNIX prompt is returned immediately, and other tasks can be carried out while the original process continues executing.

Processes and Jobs

- **Running background processes**
- To background a process, type an **&** at the end of the command line. For example, the command sleep waits a given number of seconds before continuing. Type

  % sleep 10
- This will wait 10 seconds before returning the command prompt %. Until the command prompt is returned, you can do nothing except wait.
- To run sleep in the background, type

  % sleep 10 &

  [1] 6259
- The **&** runs the job in the background and returns the prompt straight away, allowing you do run other programs while waiting for that one to finish.
- The first line in the above example is typed in by the user; the next line, indicating job number and PID, is returned by the machine.
- The user is be notified of a job number (numbered from 1) enclosed in square brackets, together with a PID and is notified when a background process is finished.
- Backgrounding is useful for jobs which will take a long time to complete.

Backgrounding a current foreground process

- You can suspend the process running in the foreground by typing **^Z**, i.e.hold down the [**Ctrl**] key and type [**z**]. Then to put it in the background, type

  % bg

- Note: do not background programs that require user interaction e.g. vi

listing suspended and background processes

- When a process is running, backgrounded or suspended, it will be entered onto a list along with a job number. To examine this list, type

  % jobs

- An example of a job list could be

  [1] Suspended sleep 1000
  [2] Running netscape
  [3] Running matlab

- To restart (foreground) a suspended processes, type

  % fg %*jobnumber*

- For example, to restart sleep 1000, type

  % fg %1

- Typing fg with no job number foregrounds the last suspended process.

Killing a process
kill (terminate or signal a process)

- It is sometimes necessary to kill a process (for example, when an executing program is in an infinite loop)
- To kill a job running in the foreground, type **^C** (control c). For example, run

```
% sleep 100
^C
```

ps (process status)

- Alternatively, processes can be killed by finding their process numbers (PIDs) and using kill *PID_number*

  % sleep 1000 &
  % ps

- PID TT S TIME COMMAND
  20077 pts/5 S 0:05 sleep 1000
  21563 pts/5 T 0:00 netscape
  21873 pts/5 S 0:25 nedit
- To kill off the process **sleep 1000**, type

  % kill 20077

  and then type ps again to see if it has been removed from the list.
- If a process refuses to be killed, uses the **-9** option, i.e. type

  % kill -9 20077
- Note: It is not possible to kill off other users' processes !!!

Here U →
<-- (Lets Recap here)

Command	Meaning
ls -lag	list access rights for all files
chmod [*options*] *file*	change access rights for named file
***command* &**	run command in background
^C	kill the job running in the foreground
^Z	suspend the job running in the foreground
bg	background the suspended job
jobs	list current jobs
fg %1	foreground job number 1
kill %1	kill job number 1
ps	list current processes
kill 26152	kill process number 26152

CHAPTER 6
FEW MORE USEFUL COMMANDS

- **quota**
- All students are allocated a certain amount of disk space on the file system for their personal files, usually about 100Mb. If you go over your quota, you are given 7 days to remove excess files.
- To check your current quota and how much of it you have used, type

  % quota -v

- Uname –a
- This will let u know which version of linux are u using!

- **du**
- The du command outputs the number of kilobyes used by each subdirectory. Useful if you have gone over quota and you want to find out which directory has the most files. In your home-directory, type

  % du -s *

- The **-s** flag will display only a summary (total size) and the * means all files and directories.

Contd.,

- **gzip**
- This reduces the size of a file, thus freeing valuable disk space. For example, type

  % ls -l science.txt

  and note the size of the file using ls -l . Then to compress science.txt, type

  % gzip science.txt
- This will compress the file and place it in a file called **science.txt.gz**
- To see the change in size, type ls -l again.
- To expand the file, use the gunzip command.

  % gunzip science.txt.gz

- **zcat**
- zcat will read gzipped files without needing to uncompress them first.

  % zcat science.txt.gz
- If the text scrolls too fast for you, pipe the output though less .

  % zcat science.txt.gz | less

- **diff**
- This command compares the contents of two files and displays the differences. Suppose you have a file called **file1** and you edit some part of it and save it as **file2**. To see the differences type

  % diff file1 file2

History is important ☺

% history (show command history list)

% !! (recall last command)

% !-3 (recall third most recent command)

% !5 (recall 5th command in list)

% !grep (recall last command starting with grep)

- You can increase the size of the history buffer by typing

  % set history=100

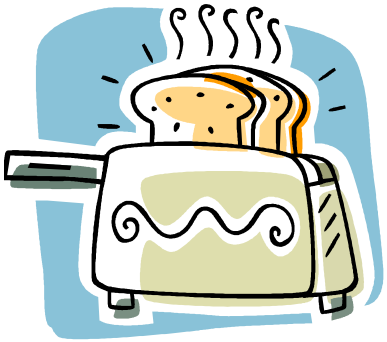
Chapter 7
We are nearing to end☺

- **environment Variables**
- An example of an environment variable is the OSTYPE variable. The value of this is the current operating system you are using. Type
  - % echo $OSTYPE
- More examples of environment variables are
  - USER (your login name)
  - HOME (the path name of your home directory)
  - HOST (the name of the computer you are using)
  - ARCH (the architecture of the computers processor)
  - DISPLAY (the name of the computer screen to display X windows)
  - PRINTER (the default printer to send print jobs)
  - PATH (the directories the shell should search to find a command)

Finding out the current values of these variables.

- ENVIRONMENT variables are set using the setenv command, displayed using the printenv or env commands, and unset using the unsetenv command.
- To show all values of these variables, type
  - % printenv | less

What is Process??



Let`s have some food for brain ☺

What is Process?? - Most important thing to know

- Process is any kind of program or task carried out by your PC. For e.g. **$ ls -lR , is** command or a request to list files in a directory and all subdirectory in your current directory. It is a process.
- A process is program (command given by user) to perform some Job.
- In Linux when you start process, it gives a number (called PID or process-id), PID starts from 0 to 65535.

Contd.,

- **$ ls / -R | wc -l**

- This command will take lot of time to search all files on your system. So you can run such command in Background or simultaneously by giving command like

  $ ls / -R | wc -l &

- The ampersand (&) at the end of command tells shells start command (ls / -R | wc -l) and run it in background takes next command immediately.
- An instance of running command is called process and the number printed by shell is called process-id (PID), this PID can be use to refer specific running process.

Process Related Commands

For this purpose	Use this Command	Example
To see currently running process	ps	**$ ps**
To stop any process i.e. to kill process	kill {PID}	**$ kill 1012**
To get information about all running process	ps -ag	**$ ps -ag**
To stop all process except your shell	kill 0	**$ kill 0**
For background processing (With &, use to put particular command and program in background)	linux-command &	**$ ls / -R | wc -l &**

NOTE that you can only kill process which are created by yourself. A Administrator can almost kill 95-98% process. But some process can not be killed, such as VDU Process.

Advanced Commands in Alphabetical Order

- 1. Alias

```
shri@ubuntu:~$ alias d='date'
shri@ubuntu:~$ d
Tue Mar 30 02:01:20 PDT 2010
shri@ubuntu:~$
shri@ubuntu:~$ alias w='who am i'
shri@ubuntu:~$ w
shri     pts/0        2010-03-26 21:19 (:0.0)
```

AWK

- awk command is used to manipulate the text. This command checks each line of a file, looking for patterns that match those given on the command line.

```
shri@ubuntu:~$ cat file1.txt
14 15 16
17 18 19
20 21 22
23 24 25
shri@ubuntu:~$ awk '{print $2}' file1.txt
15
18
21
24
shri@ubuntu:~$
```

AWK – Slight Improvement

To multiply the column-1 and column-2 and redirect the output to file2.txt:

```
awk '{print $1,$2,$1*$2}' file1.txt > file2.txt
```

Command Explanation:
$1 : Prints 1st column
$2 : Prints 2ndcolumn
$1*$2 : Prints Result of $1 x $2
file1.txt : input file
> : redirection symbol
file2.txt : output file

```
shri@ubuntu:~$ awk '{print $1,$2,$1*$2}' file1.txt > file2.txt
shri@ubuntu:~$ cat file2.txt
14 15 210
17 18 306
20 21 420
23 24 552
```

Bc – Basic Calculator

```
shri@ubuntu:~$ bc -l
bc 1.06.94
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation,
        Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.

1+2
3

3+2
5

quit
```

```
shri@ubuntu:~$ bc
bc 1.06.94
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation,
        Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.

9*2
18

9+2
11

10-10
0
```

Quit - Will take you out of the calculator.

- bc command is used for command line calculator. It is similar to basic calculator. By using which we can do basic mathematical calculations.

```
shri@ubuntu:~$ cat > file1.txt
1 + 2
quit

shri@ubuntu:~$ bc file1.txt
bc 1.06.94
Copyright 1991-1994, 1997, 1998, 2000, 2004,
        2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO
        WARRANTY.
For details type `warranty'.
3
shri@ubuntu:~$
```

Bg and jobs

It is helpful to list the jobs that are running in the background.

```
shri@ubuntu:~$ jobs
[1]+  Running                 gedit file1.txt &
shri@ubuntu:~$ bg
bash: bg: job 1 already in background
shri@ubuntu:~$
```

Lets zip it

- **bzip2 COMMAND:**
  bzip2 linux command is used to compress the file. Each file is replaced by a compressed version of itself with .bz2 extension

- - 1 Performs fast compression,creating a relatively large files. This is an important option over here
- When the file is compressed with -1 the size was 17706 bytes and now the filesize is 2394 bytes. The 9 makes best compression but the default is 6.

Now Zipping

shri@ubuntu:~$ cat file1.txt

1 + 2

Quit

shri@ubuntu:~$ bzip2 -c -1 file1.txt > file1.txt.bz2

shri@ubuntu:~$ ls -lrt | grep *.bz2

-rw-r--r-- 1 shri shri 54 2010-03-30 02:42 file1.txt.bz2

shri@ubuntu:~$

Now Zipping Better

$ bzip2 -c -9 hiox.txt > hscripts.txt.bz2

$ ls -l

-rw-rw-r-- 1 hiox hiox 9150000 Sep 26 18:37 hiox.txt

-rw-rw-r-- 1 hiox hiox 17706 Sep 27 12:38 hiox.txt.bz2

-rw-rw-r-- 1 hiox hiox 2394 Sep 27 13:01 hscripts.txt.bz2

Lets Catch ‘C’

- cal
- cat
- cd
- chattr
- chgrp
- chkconfig
- chmod
- chown
- chpasswd
- clear
- cmp
- cp
- cpio
- cut

CALENDAR - CAL

```
shri@ubuntu:~$ cal
    March 2010
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

The option 3 will give you Current, Previous and Next Months Calendar.
And 5 is the 5th Month - May

```
shri@ubuntu:~$ cal -3 5 2009
    April 2009          May 2009            June 2009
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
        1  2  3  4              1  2     1  2  3  4  5  6
 5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
12 13 14 15 16 17 18  10 11 12 13 14 15 16  14 15 16 17 18 19 20
19 20 21 22 23 24 25  17 18 19 20 21 22 23  21 22 23 24 25 26 27
26 27 28 29 30        24 25 26 27 28 29 30  28 29 30
                 31
```

C - Here

cal
cat
cd
chattr
chgrp
chkconfig
chmod
chown
chpasswd
clear
cmp
cp
cpio
cut

- We have spent time on Cat, cd, cp, clear, Cal and chmod.
- So let us see rest of the advanced commands here.

Going advanced …

- **chattr COMMAND:**
  chattr command is used to change the file attributes. This is an admin command. Root user only can change the file attributes/Process.

  +i Make the file as Read-Only.

  -i Remove the Read-Only.

  +a Can't open file for writing.

  -a Open file for writing.

$chattr +a file1.txt

$chattr: Operation not permitted while setting flags on file1.tx

As this is an Admin commands.. We got the above message!

chattr
chgrp
chkconfig
cmp
cpio
cut
chown
chpasswd

Going advanced

chattr
chgrp
chkconfig
cmp
cpio
cut
chown
chpasswd

- **chgrp COMMAND:**
  chgrp command is used to change the group of the file or directory. This is an admin command. Root user only can change the group of the file or directory.
- This is again a privileged command. So cant make use of it.

Going advanced

- **chown COMMAND:**
  chown command is used to change the owner / user of the file or directory. This is an admin command, root user only can change the owner of a file or directory.

  SYNTAX:
  The Syntax is ***chown [options] newowner filename/directoryname***

The owner of the 'file1.txt' file is shri, Change to new user root.

-rw-r--r-- 1 shri shri 12 2010-03-30 02:34 file1.txt

shri@ubuntu:~$ chown root file1.txt

chown: changing ownership of `file1.txt': Operation not permitted

Cmp command

- **cmp COMMAND:**
  cmp linux command compares two files and tells you which line numbers are different.

  SYNTAX:
  The Syntax is
  cmp [options..] file1 file2

  OPTIONS:

- c Output differing bytes as characters.
- l Print the byte number (decimal) and the differing byte values (octal) for each difference.
- s Prints nothing for differing files, return exit status only.

```
shri@ubuntu:~$ cmp -c  file1.txt file2.txt
file1.txt file2.txt differ: byte 2, line 1 is  40   64 4
shri@ubuntu:~$ cmp -l  file1.txt file2.txt
 2  40  64
 3  53  40
 4  40  61
 5  62  65
 6  12  40
 7 161  62
 8 165  61
 9 151  60
10 164  12
11  12  61
12  12  67
cmp: EOF on file1.txt
shri@ubuntu:~$ cmp -s  file1.txt file2.txt
shri@ubuntu:~$ cmp file1.txt file2.txt
file1.txt file2.txt differ: byte 2, line 1
```

cut

- **cut COMMAND:**
  cut command is used to cut out selected fields of each line of a file. The cut command uses delimiters to determine where to split fields.

  SYNTAX:
  The Syntax is
  cut [options]

  OPTIONS:
  -c Specifies character positions. -b Specifies byte positions.

- **EXAMPLE:**
  Lets create a file file1.txt and let it have the following data:

Data in file1.txt

- This is, an example program,for cut command.

- cut -c1-3 text.txt

- **Output:**
  Thi

- Cut the first three letters from the above line.

LETS GET INTO D

- date command
  date
- The above command will print

Wed Jul 23 10:52:34 IST 2008

- **df COMMAND:**
  df command is used to report how much free disk space is available for each mount you have. The first column show the name of the disk partition as it appears in the /dev directory. Subsequent columns show total space, blocks allocated and blocks available.

```
shri@ubuntu:~$ df
Filesystem        1K-blocks    Used Available Use% Mounted on
/dev/sda1          4878132   2476452   2153880  54% /
udev               254668      224    254444   1% /dev
none               254668      180    254488   1% /dev/shm
none               254668       92    254576   1% /var/run
none               254668        0    254668   0% /var/lock
none               254668        0    254668   0% /lib/init/rw
shri@ubuntu:~$
```

LETS GET INTO D

```
shri@ubuntu:~$ du
76      ./.gconfd
8       ./.cache/gedit
28      ./.cache
116     ./.mozilla/firefox/day8z6r3.default/Cache
4       ./.mozilla/firefox/day8z6r3.default/extensions
12      ./.mozilla/firefox/day8z6r3.default/chrome
12      ./.mozilla/firefox/day8z6r3.default/bookmarkbackups
3932    ./.mozilla/firefox/day8z6r3.default
3940    ./.mozilla/firefox
4       ./.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
8       ./.mozilla/extensions
3952    ./.mozilla
4       ./.gnome2_private
4       ./t12
12      ./.update-manager-core
8       ./.gnome2/gedit
4       ./.gnome2/nautilus-scripts
4       ./.gnome2/panel2.d/default/launchers
8       ./.gnome2/panel2.d/default
12      ./.gnome2/panel2.d
8       ./.gnome2/keyrings
28      ./.gnome2/accels
64      ./.gnome2
4       ./Documents
20      ./.thumbnails/normal
24      ./.thumbnails
0       ./.gvfs
4       ./.update-notifier
4       ./.config/gnome-session/saved-session
8       ./.config/gnome-session
```

du command is used to report how much disk space a file or directory occupies.

E – Only Echo!

F

Fg:

fg command is used to place a job in foreground.

Run some process in background.Use fg… the process will be brought to foreground.

```
shri@ubuntu:~$ gedit file1.txt &
[1] 4970
shri@ubuntu:~$ fg 1
gedit file1.txt
```

F - Finger

```
shri@ubuntu:~$ finger
Login     Name      Tty     Idle  Login Time   Office     Office Phone
shri      shriram   tty7     4d  Mar 26 20:02 (:0)
shri      shriram   pts/0        Mar 26 21:19 (:0.0)
shri@ubuntu:~$
shri@ubuntu:~$ finger shri
Login: shri                                     Name: shriram
Directory: /home/shri                Shell: /bin/bash
On since Fri Mar 26 20:02 (PDT) on tty7 from :0
   4 days 11 hours idle
On since Fri Mar 26 21:19 (PDT) on pts/0 from :0.0
No mail.
No Plan.
```

- finger command displays the user's login name, real name, terminal name and write status (as a "*" after the terminal name if write permission is denied), idle time, login time, office location and office phone number

File..

- **file COMMAND:**
  file command tells you if the object you are looking at is a file or a directory.

File * will list you the types of files available in that system.

```
shri@ubuntu:~$ file *.txt
f1.txt:     ASCII text
f2.txt:     ASCII text
file1.txt:   ASCII text
file2.txt:   ASCII text
file_t1.txt: empty
file_t2.txt: empty
file.txt:    ASCII text
linux.txt:   ASCII text
result.txt:  ASCII text
shriram.txt: ASCII text
test2.txt:   ASCII text
```

You are free after this slide.

- **free COMMAND:**
  free command displays information about free and used memory on the system.

```
shri@ubuntu:~$ free
             total       used       free     shared    buffers     cached
Mem:        509336     480580      28756          0     140528     189548
-/+ buffers/cache:     150504     358832
Swap:       281096         48     281048
shri@ubuntu:~$
```

G

groupadd
groupdel
groupmod
groups

All these are used to Add groups, Delete Groups, Modify an Existing group etc., as all these are admin commands you will end up in not having permission for executing these commands..

Typing groups will get u the details of all the available groups in you system.
shri@ubuntu:~$ groups
shri adm dialout cdrom plugdev lpadmin admin sambashare

Can We Halt? – System Related Commands

- To halt the system:
  halt
  This command is similar to poweroff, which shutdown the system.
- To Poweroff the system:
  poweroff
  Poweroff command used for turnoff the system.
- To reboot the system:
  reboot
  Reboot command used for reboots/restarts the system.

Host

- **host COMMAND:**
  host command is used to find the ip address of the given domain name and also prints the domain name for the given ip.

```
shri@ubuntu:~$ host vit.ac.in
vit.ac.in has address 192.168.64.3
vit.ac.in mail is handled by 5 alt1.aspmx.l.google.com.\032.
vit.ac.in mail is handled by 5 alt2.aspmx.l.google.com.
vit.ac.in mail is handled by 10 aspmx2.googlemail.com.\032.
vit.ac.in mail is handled by 10 aspmx3.googlemail.com.\032.
vit.ac.in mail is handled by 10 aspmx4.googlemail.com.\032.
vit.ac.in mail is handled by 10 aspmx5.googlemail.com.\032.
vit.ac.in mail is handled by 1 aspmx.l.google.com.\032.
shri@ubuntu:~$ host yahoo.co.in
yahoo.co.in has address 68.180.206.184
yahoo.co.in has address 206.190.60.37
yahoo.co.in mail is handled by 10 in32.mxauth.yahoo.com.
shri@ubuntu:~$ host 68.180.206.184
184.206.180.68.in-addr.arpa domain name pointer w2.rc.vip.sp1.yahoo.com.
shri@ubuntu:~$
```

Host id and Host Name

- **hostid COMMAND:**
  hostid command prints the numeric identifier or id of the current host in hexadecimal.

  SYNTAX:
  The Syntax is
  hostid

```
shri@ubuntu:~$ hostid
007f0101
shri@ubuntu:~$
```

- **hostid COMMAND:**
  hostid command prints the numeric identifier or id of the current host in hexadecimal.

  SYNTAX:
  The Syntax is
  hostid

```
shri@ubuntu:~$ hostname
UBUNTU
shri@ubuntu:~$
```

id

id COMMAND:

id command prints the effective(current) and real userid(UID)s and groupid(GID)s.

SYNTAX:

The Syntax is

id

shri@ubuntu:~$ id

uid=1000(shri) gid=1000(shri) groups=4(adm),20(dialout),24(cdrom),46(plugdev),104(lpadmin),115(admin),120(sambashare),1000(shri)

info

- info command is used to display the readable online documentation for the specified command .
- Typing just info will lead you screen to be filled with hell a lot of data!

- shri@ubuntu:~$ info man
- shri@ubuntu:~$ info cp

- **IFCONFIG**
- ifconfig command displays information about the network interfaces attached to the system and also used to configure the network interface.

To Assign IP address to Network Interface[Ethernet Card]:

ifconfig eth0 192.168.0.12 up

The above command will Assign IP address 192.168.0.12 to Ethernet card with name eth0.

To inactivate the Network Interface[Ethernet Card]:

ifconfig eth0 down

The above command inactivates the ethernet card.

```
shri@ubuntu:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b5:ee:52
          inet addr:192.168.98.131  Bcast:192.168.98.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb5:ee52/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:814 errors:0 dropped:0 overruns:0 frame:0
          TX packets:233 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:94010 (94.0 KB)  TX bytes:31518 (31.5 KB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:240 (240.0 B)  TX bytes:240 (240.0 B)
```

netstat

- nestat command displays statistics information and current state of network connections, protocol, ports/ sockets and devices.

- **unix 3 [ ] STREAM CONNECTED 9803 @/tmp/dbus-3Imoh4QFZa**
- **unix 3 [ ] STREAM CONNECTED 9802**
- **unix 3 [ ] STREAM CONNECTED 9801 /tmp/orbit-shri/linc-7b7-0-28f9f13a74fe6**
- **unix 3 [ ] STREAM CONNECTED 9800**
- **unix 3 [ ] STREAM CONNECTED 9799 /tmp/orbit-shri/linc-773-0-455a56f8d87af**
- **unix 3 [ ] STREAM CONNECTED 9796**
- **unix 3 [ ] STREAM CONNECTED 9792 @/tmp/.X11-unix/X0**
- **unix 3 [ ] STREAM CONNECTED 9791**
- **unix 3 [ ] STREAM CONNECTED 9790 /tmp/orbit-shri/linc-7b6-0-3923529828c29**

route

- route command displays routing table resides in kernel and also used to modify the routing table.
- The tables which specifies how packets are routed to a host is called routing table.

```
shri@ubuntu:~$ route –n  (Where –n is net)
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.98.0    0.0.0.0         255.255.255.0   U     1      0        0 eth0
0.0.0.0         192.168.98.2    0.0.0.0         UG    0      0        0 eth0
shri@ubuntu:~$
```

Yes… Am Saying Bye

- yes command repeatedly prints the given string separated by a space and followed by a newline until it is killed. If no string is given, it just prints 'y' repeatedly until it is killed. It is normally used in scripts, its output is piped to a command or program that prompts you to do this or that (do you want to delete this file press 'y' or 'n')

```
shri@ubuntu:~$ yes shriram | more
shriram
shriram
shriram
shriram
shriram
shriram
shriram
shriram
shriram
shriram
shriram
```

Let me say bye to commands here!

Itz your turn ☺