Linux For Beginners

Name : Abdul Khadar & Malik Basha
reviewer : Maheswara Reddy

Contents:

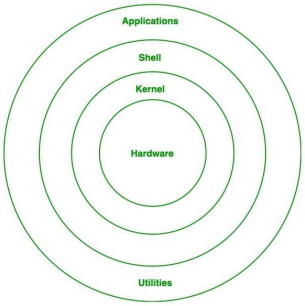
0.1 Introduction to Linux

In 1990, PCs were fully powered by UNIX but still free software was unavailable. In 1991, Linus Torvalds began developing an operating system called Linux which is a member of the large family UNIX-like OS. Linux was developed for an IBM- compatible personal computer based on Intel 80386 microprocessor. All features of UNIX were added in Linux within a few years, which gave rise to a mature operating system called Linux. Today Linux is supported by all kinds of workstations, home user PC and both server and client.

What is Linux

Linux is a free open source operating system based on UNIX that was created in 1991 by Linus Torvalds. Users can modify and create variations of the source code, known as distributions, for computers and other devices.

Linux Architecture



Linux Distributions List

Ubuntu
Linux Mint

Debian
Red Hat Enterprise / CentOS
Fedora

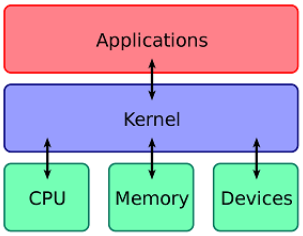
Linux Features

Free and Open-Source
multiple distributions - ubuntu,Linux Mint, Debian, RedHat, Centos, Fedora
Access to source code
Better malware protection
Command line interface

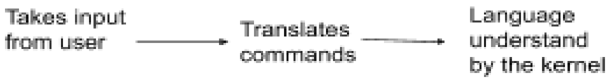
What is A shell

A shell is a program that provides an interface between a user and an operating system OS kernel. An OS starts a shell for each user when the user logs in or opens a terminal or console window.

A kernel is the very core of a typical operating system. A kernel is a type of low-level programming that has its interfacing with the hardware on top of which all the applications run like - Disks, Ram, CPU etc.



A kernel is a program - It controls all computer operations, coordinates all executing utilities, ensures that executing utilities do not interfere with each other and schedules and manages all system processes.



By interfacing with a kernel, a shell provides a way for a user to execute utilities and programs.

What is Shell Script

A shell script is a list of commands in a computer program that is run by the Unix shell which is a command line interpreter.

Types of Shells

Bourne Shell -

Bourne shell (sh) :

The Bourne shell (sh), written by Steve Bourne at AT&T Bell Labs, is the original UNIX shell. A Bourne shell drawback is that it lacks features for interactive use, recall previous commands (history), built-in arithmetic and logical expression handling.

Command full-path name: /bin/sh and /sbin/sh
Non-root user default prompt: $
Root user default prompt : #

Korn Shell (ksh) :

Korn Shell (ksh), written by David Korn at AT&T Bell Labs, is a superset of the Bourne shell. It includes convenient programming features like built-in arithmetic and C-like arrays, functions, and string-manipulation facilities.

Command full-path name: /bin/ksh
Non-root user default prompt: $
Root user default prompt : #

Bourne Again shell (bash):

Bourne Again shell (bash) - compatible with the Bourne shell. It incorporates useful features from the Korn and C shells.

Command full-path name: /bin/bash.
Non-root user default prompt: bash-x.xx$
Root user default prompt : bash-x.xx#

C Shell:

The C shell (csh)- Is a UNIX enhancement written by Bill Joy at the University of California at Berkeley. Incorporated features for interactive use, such as aliases and command history also includes convenient programming features, such as built-in arithmetic and a C-like expression syntax.

Command full-path name: /bin/csh.
Non-root user default prompt: %
Root user default prompt : #

Difference between sh and bash

Sh	Bash
SHell	Bourne Again SHell
Developed by Stephen R. Bourne	Developed by Brain Fox
Predecessor of bash	Successor of sh
sh is the not default SHELL	Bash is the default SHELL
#!/bin/sh	#!/bin/bash
It has less functionality.	It has more Functionality with up-gradation
not as easy as bash	Easy to use

1. File commands

1.1 Command : **ls**

Syntax: $ls

Explanation: It will show the full list or content of your directory.

Example -

```
azureuser@unbutu:~$ ls
A1  B
azureuser@unbutu:~$
```

1. 2 Command : **ls -al**

Syntax: $ls

Explanation: formatted listing with hidden files.

Example -

```
azureuser@unbutu:~$ ls -al
total 68
drwxr-xr-x  7 azureuser azureuser 4096 Jan 18 09:07 .
drwxr-xr-x  3 root      root      4096 Jan 17 07:53 ..
-rw-------  1 azureuser azureuser 3612 Jan 18 13:14 .bash_history
-rw-r--r--  1 azureuser azureuser  220 Apr  4  2018 .bash_logout
-rw-r--r--  1 azureuser azureuser 3771 Apr  4  2018 .bashrc
drwx------  2 azureuser azureuser 4096 Jan 17 07:54 .cache
```

1.3 Command : **cd**

Syntax: cd <dirname>

Explanation: is used to change the current working directory.

Example -

```
azureuser@unbutu:~$ cd B
azureuser@unbutu:~/B$
```

1.4 Command : **pwd**

Syntax: pwd

Explanation: pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root.

Example -

```
azureuser@unbutu:~$ pwd
/home/azureuser
```

1.5 Command : **rm**

Syntax: rm [OPTION]. Filename

Explanation: rm stands for remove here. rm command is used to remove objects such as files, directories.

Example -

Ex1 -

```
azureuser@unbutu:~/B$ ls
A1  B1  D1  E1
azureuser@unbutu:~/B$ rm A1
azureuser@unbutu:~/B$ ls
B1  D1  E1
azureuser@unbutu:~/B$
```

Ex2 - Remove more files

```
azureuser@unbutu:~/B$ ls
B1  D1  E1
azureuser@unbutu:~/B$ rm B1 D1
azureuser@unbutu:~/B$ ls
E1
azureuser@unbutu:~/B$
```

Ex3-

```
azureuser@unbutu:~$ rm -f B1
azureuser@unbutu:~$ ls
B  D1  Devops  Devops.tar  E1  dest.txt  devops
azureuser@unbutu:~$
```

1.6 Command : **cp**

Syntax:

```
cp [OPTION] Source Destination
cp [OPTION] Source Directory
cp [OPTION] Source-1 Source-2 Source-3 Source-n Directory
```

Explanation: cp stands for copy. This command is used to copy files or groups of files or directory.

Example -

```
azureuser@unbutu:~$ cat dest.txt
learning azure devops
```

```
azureuser@unbutu:~$ touch E1
azureuser@unbutu:~$ ls
B  B1  D1  E1  dest.txt
azureuser@unbutu:~$ cp dest.txt E1
azureuser@unbutu:~$ cat E1
learning azure devops
azureuser@unbutu:~$
```

Ex2 -

```
azureuser@unbutu:~$ ls
B  B1  D1  E1  dest.txt
azureuser@unbutu:~$ cp E1 B
azureuser@unbutu:~$ ls
B  B1  D1  E1  dest.txt
azureuser@unbutu:~$ cd B
azureuser@unbutu:~/B$ ls
A1  E1
azureuser@unbutu:~/B$
```

Ex3 -

```
azureuser@unbutu:~$ ls
B  B1  D1  E1  dest.txt
azureuser@unbutu:~$ cp B1 D1 E1 B
azureuser@unbutu:~$ ls
B  B1  D1  E1  dest.txt
azureuser@unbutu:~$ cd B
azureuser@unbutu:~/B$ ls
A1  B1  D1  E1
azureuser@unbutu:~/B$
```

1.7 Command : mv

Syntax: mv [Option] source destination

Explanation: mv stands for move. mv is used to move one or more files or directories from one place to another

Example -

Ex1 -

```
azureuser@unbutu:~$ ls
A1  B  B1  C1  D1
azureuser@unbutu:~$ mv A1 B
azureuser@unbutu:~$ ls
B  B1  C1  D1
azureuser@unbutu:~$ cd B
azureuser@unbutu:~/B$ ls
A1
azureuser@unbutu:~/B$
```

Ex2 - If the destination file doesn’t exist, it will be created.

```
azureuser@unbutu:~$ ls
B  B1  C1  D1
azureuser@unbutu:~$ cat C1
learning azure devops
azureuser@unbutu:~$ mv C1 dest.txt
azureuser@unbutu:~$ ls
B  B1  D1  dest.txt
azureuser@unbutu:~$ cat dest.txt
learning azure devops
azureuser@unbutu:~$
```

1.8 Command : **touch**

Syntax: $touch filename

Explanation: The touch command is a standard command used in the UNIX/Linux operating system which is used to create, change and modify timestamps of a file.

Example -

```
azureuser@unbutu:~$ touch devops
azureuser@unbutu:~$ ls
B  B1  D1  E1  dest.txt  devops
azureuser@unbutu:~$
```

1. 9 Command : **cat**

Syntax: $cat filename

Explanation: It reads data from the file and gives their content as output. It helps us to create, view, and concatenate files.

Example -

Ex1 -

```
azureuser@unbutu:~$ cat A1
Abc
```

Ex2 -

```
azureuser@unbutu:~$ cat> B1
Azure Devops^C
```

1.10 Command : **more**

Syntax: more [option] filename

Explanation: more commands are used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files). The more command also allows the user to scroll up and down through the page.

Example -

Ex1 -

```
azureuser@unbutu:~$ more C1
learning devops
azureuser@unbutu:~$ more -d dest.txt
```

Ex2 -

```
azureuser@unbutu:~$ more -d dest.txt
learning azure devops
Afghanistan
Albania
Algeria
Andorra
Angola
Antigua and Barbuda

--more--(89%)[Press space to continue, 'q' to quit.]
```

1.11 Command : **less**

Syntax: less <file>
Explanation: less command displays a file page-wise. By pressing spacebar, it displays next page and to quit , type ‘q’.
Example -

```
azureuser@unbutu:~$ less C1
Learning Azure Devops
Practicing linux Commands
:
```

1.12 Command : **Head**

Syntax: head [option] file_name
Explanation: The Linux head command prints the first lines of one or more files (or piped data) to standard output. By default, it shows the first 10 lines.
Example -

```
azureuser@unbutu:~$ head dest.txt
learning azure devops
Afghanistan
Albania
Algeria
Andorra
Angola
Antigua and Barbuda
Argentina
Armenia
Angola
azureuser@unbutu:~$
azureuser@unbutu:~$ head -5 dest.txt
learning azure devops
Afghanistan
Albania
Algeria
Andorra
```

```
azureuser@unbutu:~$
```

1.13 Command : tail

Syntax: tail [OPTION]... [FILE]...

Explanation: It is complementary to head command. The tail command, as the name implies, prints the last N number of data of the given input. By default, it prints the last 10 lines of the specified files.

Example -

Ex1 -

```
azureuser@unbutu:~$ tail dest.txt
Angola
Antigua and Barbuda
Argentina
Armenia
Angola
Antigua and Barbuda
Argentina
Armenia
Argentina
Armenia
azureuser@unbutu:~$
```

Ex2-

```
azureuser@unbutu:~$ tail -3 dest.txt
Armenia
Argentina
Armenia
azureuser@unbutu:~$
```

1.14 Command : mkdir

Syntax: mkdir [options...] [directories ...]

Explanation: mkdir command in Linux allows the user to create directories and multiple directories as well.

Example -

Ex1 - single directories

```
azureuser@unbutu:~/B$ ls
E1
azureuser@unbutu:~/B$ mkdir AzureDevops
azureuser@unbutu:~/B$ ls
AzureDevops  E1
azureuser@unbutu:~/B$
```

Ex2 - multiple directories

```
azureuser@unbutu:~/B$ mkdir {test1,test2,test3}
azureuser@unbutu:~/B$ ls
AzureDevops  E1  test1  test2  test3
```

```
azureuser@unbutu:~/B$
```

Ex3 - Creates a directory and sets full read, write, execute permissions for all users

```
azureuser@unbutu:~/B$ mkdir –m777 test6
azureuser@unbutu:~/B$ ls
AzureDevops  E1  test1  test2  test3  test5  test6  –m700  –m777
azureuser@unbutu:~/B$
```

1.15 Command : rmdir

Syntax: rmdir <dirname>

Explanation: This command is used to delete a directory.

Example -

```
azureuser@unbutu:~$ mkdir Azure
azureuser@unbutu:~$ ls
Azure  B  B1  D1  E1  dest.txt
azureuser@unbutu:~$ rmdir Azure
azureuser@unbutu:~$ ls
B  B1  D1  E1  dest.txt
azureuser@unbutu:~$
```

2 System information

2.1 Command : whoiam

Syntax: $whoiam

Explanation: It displays the username of the current user.

Example -

```
azureuser@unbutu:~$ whoami
azureuser
```

2.2 Command : date

Syntax: date [option]

Explanation: date command is used to display the system date and time.

Example -

Ex1 -

```
azureuser@unbutu:~$ date
Thu Jan 19 06:02:29 UTC 2023
```

Ex2 - -u Option: Displays the time in GMT

```
azureuser@unbutu:~$ date -u
Thu Jan 19 06:04:25 UTC 2023
```

2.3 Command : man

Syntax: $ man [COMMAND NAME]

Explanation: Man stands for manual and is used to get as much information as is possible about any commands that can be run within the terminal

Example -

Ex1 -

```
azureuser@unbutu:~$ man ls

LS(1)                        User Commands                        LS(1)

NAME
      ls - list directory contents

SYNOPSIS
      ls [OPTION]... [FILE]...

DESCRIPTION
      List  information  about  the FILEs (the current directory by default).
      Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
      fied.
```

Ex2 -

```
azureuser@unbutu:~$ man echo

ECHO(1)                      User Commands                      ECHO(1)

NAME
      echo - display a line of text

SYNOPSIS
      echo [SHORT-OPTION]... [STRING]...
      echo LONG-OPTION

DESCRIPTION
      Echo the STRING(s) to standard output.
```

2.4 Command : **du**

Syntax: du [OPTION]... [FILE]....

Explanation: locate command in Linux is used to find the files by name.

Example -

```
azureuser@unbutu:~$ du dest.txt
4       dest.txt
azureuser@unbutu:~$ du -h /home/azureuser/dest.txt
4.0K    /home/azureuser/dest.txt
azureuser@unbutu:~$
```

2.5 Command : **free**

Syntax: free [options]

Explanation: The Linux free command outputs a summary of RAM usage, including total, used, free, shared, and available memory and swap space.

Example -

```
azureuser@unbutu:~$ free
              total        used        free      shared  buff/cache   available
Mem:         934096      157716      213328         612      563052      622928
Swap:             0           0           0
azureuser@unbutu:~$
```

3. Searching

3.1 Command : grep

Syntax: grep [options] pattern [files]

Explanation: The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.

Example -

Ex1 -

```
azureuser@unbutu:~$ grep -w "Azure" C1
Learning Azure Devops
```

Ex2 -

```
azureuser@unbutu:~$ grep -n "Azure" C1
1:Learning Azure Devops
```

Ex3 -

```
azureuser@unbutu:~$ grep -l "Azure" *
grep: B: Is a directory
C1
```

4. Process Management

4.1 Command : ps

Syntax: $ps [options]

Explanation: ps for viewing information related with the processes on a system which stands as abbreviation for “Process Status”.

Example -

Ex1 -

```
azureuser@unbutu:~$ ps
  PID TTY          TIME CMD
10164 pts/0    00:00:00 bash
10182 pts/0    00:00:00 ps
azureuser@unbutu:~$
```

Ex2-

```
azureuser@unbutu:~$ ps -r
  PID TTY      STAT   TIME COMMAND
```

```
10186 pts/0    R+     0:00 ps -r
azureuser@unbutu:~$
```

5. File permissions

5.1 Command : **chmod**
Syntax: chmod [mode-numerical/alphabetical] filename
Explanation: the chmod command is used to change the access mode of a file.
Example -

5.1.1 Numerical Notation

Ex1 - changing access to owner

```
azureuser@unbutu:~$ ls -la dest.txt
-rwxrwxrwx 1 azureuser azureuser 23 Jan 17 10:43 dest.txt
azureuser@unbutu:~$ chmod 477 dest.txt
azureuser@unbutu:~$ ls -la dest.txt
-r--rwxrwx 1 azureuser azureuser 23 Jan 17 10:43 dest.txt
```

Ex2 - changing access to group

```
azureuser@unbutu:~$ chmod 427 dest.txt
azureuser@unbutu:~$ ls -la dest.txt
-r---w-rwx 1 azureuser azureuser 23 Jan 17 10:43 dest.txt
azureuser@unbutu:~$
```

Ex3 - changing access to others

```
azureuser@unbutu:~$ chmod 421 dest.txt
azureuser@unbutu:~$ ls -la dest.txt
-r---w---x 1 azureuser azureuser 23 Jan 17 10:43 dest.txt
```

5.1.2 Alphabetical notation

Ex4 - Giving read,write,execute permissions

```
azureuser@unbutu:~$ ls -la dest.txt
-r---w---x 1 azureuser azureuser 23 Jan 17 10:43 dest.txt
azureuser@unbutu:~$ chmod u+wx,g+rx,o+rw dest.txt
azureuser@unbutu:~$ ls -la dest.txt
-rwxrwxrwx 1 azureuser azureuser 23 Jan 17 10:43 dest.txt
```

6. useful commands

6.1 Command : echo

Syntax: echo [option] [string]
Explanation: display a line of text
Example -

```
azureuser@unbutu:~$ echo "Azure Practice"
Azure Practice
```

6.2 Command : sudo

Syntax: sudo [command]
Explanation: Sudo stands for SuperUser DO and is used to access restricted files and operations.
Example -
Without Prefix Sudo :

```
azureuser@unbutu:~$ apt-get update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
```

With Prefix Sudo :

```
azureuser@unbutu:~$ sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
```

6.3 Command : clear

Syntax: $clear
Explanation: is used to clear the terminal screen.
Example -
before clear command:

```
azureuser@unbutu:~$ ls
A1
```

After clear command:

```
azureuser@unbutu:~$
```

6.4 Command : sort

Syntax: sort [options] Filename
Explanation: The 'sort' command sorts the file content in an alphabetical order.
Example -

```
azureuser@unbutu:~$ cat D1
Devops
Azure
azureuser@unbutu:~$ sort D1
Azure
Devops
azureuser@unbutu:~$ sort -r D1
Devops
```

```
Azure
azureuser@unbutu:~$
```

6.5 Command : su

Syntax: $su user

Explanation: it is used for switching to another user during a normal login session.

Example -

```
azureuser@unbutu:~$ su Ab
Password:
Ab@unbutu:/home/azureuser$ ls
B  B1  D1  E1  dest.txt
Ab@unbutu:/home/azureuser$ cd ../..
Ab@unbutu:/$ ls
bin   home            lib64       opt   sbin  tmp      vmlinuz.old
boot  initrd.img      lost+found  proc  snap  usr
dev   initrd.img.old  media       root  srv   var
etc   lib             mnt         run   sys   vmlinuz
Ab@unbutu:/$
```

6.6 Command : id

Syntax: id [OPTION]... [USER]

Explanation: print real and effective user and group IDs

Example -

```
azureuser@unbutu:~$ id
uid=1000(azureuser) gid=1000(azureuser)
groups=1000(azureuser),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44
(video),46(plugdev),108(lxd),114(netdev)
azureuser@unbutu:~$ id Ab
uid=1001(Ab) gid=1001(Ab) groups=1001(Ab)
azureuser@unbutu:~$ id -u Ab
1001
```

```
azureuser@unbutu:~$ id -g Ab
1001
azureuser@unbutu:~$
```

6.7 Command : **cut**

Syntax: cut OPTION... [FILE]...

Explanation: It is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field.

Example -

```
azureuser@unbutu:~$ cut -c1 dest.txt
I
A
A
A
A
A
azureuser@unbutu:~$ cut -c1-2 dest.txt
Ie
Af
Al
Al
An
azureuser@unbutu:~$
```

6.8 Command : **history**

Syntax: $ history

Explanation: history command is used to view the previously executed command.

Example -

```
azureuser@unbutu:~$ history
    1  clear
    2  ls
    3  cat > A1
    4  cat A1
    5  ls
    6  cat A1
    7  cat > A1
    8  cat A1
    9  clear
   10  ls -l
   11  chown Ab A1
   12  chown root A1
   13  sudo chown Ab A1
   14  chown Ab A1.txt
   15  ls -l
   16  chown newowner A1
   17  sudo chown Ab:azuruser A1
   18  su
   19  ss
   20  clear
   21  sudo su
   22  clear
   23  pwd
   24  echo "Azure Practice"
   25  apt-get update
   26  sudo apt-get update
   27  clear
```

6.9 Command : **locate**

Syntax: locate [OPTION]... PATTERN...

Explanation: locate command in Linux is used to find the files by name.

Example -

```
azureuser@unbutu:~$ locate dest.txt
/home/azureuser/dest.txt
azureuser@unbutu:~$
```

6.10 Command : **top**

Syntax: $top [options]

Explanation: this command shows a real-time view of running processes in Linux and displays kernel-managed tasks. The command also provides a system information summary that shows resource utilization, including CPU and memory usage.

Example -

Ex1-

```
top - 12:34:09 up 1 day,  4:47,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  95 total,   1 running,  49 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   934096 total,   203960 free,   158000 used,   572136 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   622396 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1678 root      20   0  460648  27468   9868 S  0.3  2.9   1:28.37 python3
    1 root      20   0  160576   9796   6552 S  0.0  1.0   0:03.49 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
```

Ex2 -

```
azureuser@unbutu:~$ top -h
```

```
  procps-ng 3.3.12
Usage:
  top -hv | -bcHiOSs -d secs -n max -u|U user -p pid(s) -o field -w [cols]
azureuser@unbutu:~$
```

6.11 Command : w

Syntax: $w

Explanation: The w command is a built-in tool that allows administrators to view information about users that are currently logged in. This includes their username, where they are logged in from, and what they are currently doing.

Example -

```
azureuser@unbutu:~$ w
 12:41:44 up 1 day,  4:55,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
azureuse pts/0    182.75.74.86     12:35    0.00s  0.02s  0.00s w
azureuser@unbutu:~$
```

6.12 Command : uptime

Syntax: uptime [-options]

Explanation: It is used to find out how long the system is active (running).

Example -

Ex1 -

```
azureuser@unbutu:~$ uptime
 12:44:05 up 1 day,  4:57,  1 user,  load average: 0.00, 0.00, 0.00
azureuser@unbutu:~$
```

Ex2 -

```
azureuser@unbutu:~$ uptime -s
2023-01-17 07:46:16
azureuser@unbutu:~$
```

6.13 Command : ifconfig

Syntax: ifconfig [OPTION]

Explanation: ifconfig command without any argument displays the details of all the active interfaces. This command also displays the assigned ip address of active interfaces.

Example -

```
azureuser@unbutu:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.4  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::6245:bdff:fe72:e1e0  prefixlen 64  scopeid 0x20<link>
        ether 60:45:bd:72:e1:e0  txqueuelen 1000  (Ethernet)
        RX packets 362368  bytes 132195910 (132.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 408458  bytes 108556562 (108.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1588  bytes 139740 (139.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1588  bytes 139740 (139.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

azureuser@unbutu:~$
```

Ex2 -

```
azureuser@unbutu:~$ ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500   362582      0      0 0        408718      0      0      0 BMRU
lo       65536     1588      0      0 0          1588      0      0      0 LRU
azureuser@unbutu:~$
```

6.14 Command : **sed**

Syntax: sed 'argument' filename

Explanation: SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution)

Example -

```
azureuser@unbutu:~$ sed "azure1" dest.txt
learning
zure1
azureuser@unbutu:~$
```

6.15 Command : **tree**

Syntax: $tree

Explanation: Tree is a recursive directory listing program that produces a depth indented listing of files

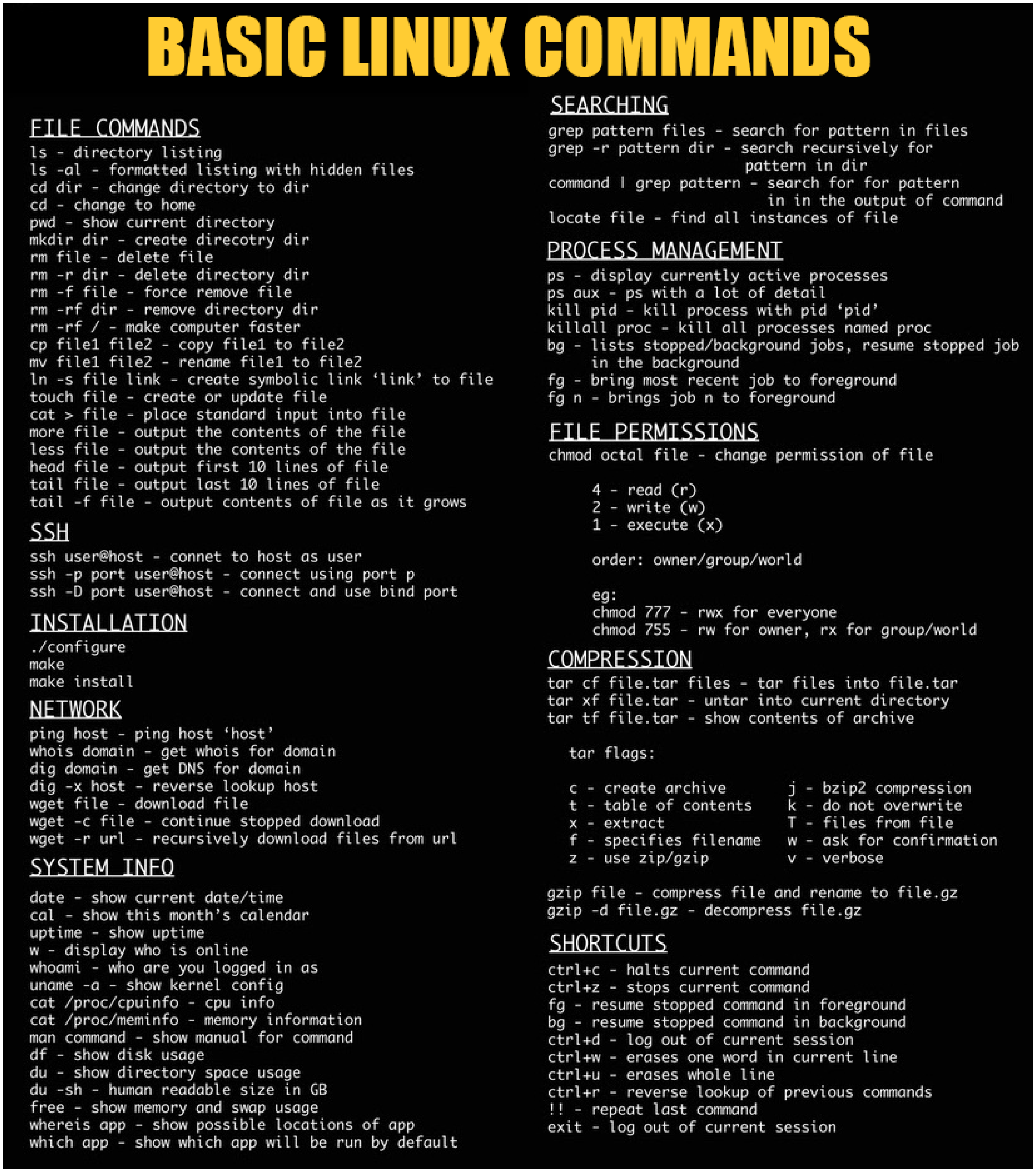
Example -

```
azureuser@unbutu:~$ tree
.
├── B
│   ├── AzureDevops
│   ├── E1
│   ├── test1
│   ├── test2
│   ├── test3
│   ├── test5
│   ├── test6
│   ├── –m700
│   └── –m777
├── B1
├── D1
├── Devops
│   ├── Learning.txt
│   ├── azure.txt
│   └── devops.txt
├── Devops.tar
├── E1
├── dest.txt
└── devops

10 directories, 10 files
azureuser@unbutu:~$
```

Reference

Youtube link - https://www.youtube.com/watch?v=Wgi-OfbP2Gw



BASIC LINUX COMMANDS

FILE COMMANDS

```
ls - directory listing
ls -al - formatted listing with hidden files
cd dir - change directory to dir
cd - change to home
pwd - show current directory
mkdir dir - create direcotry dir
rm file - delete file
rm -r dir - delete directory dir
rm -f file - force remove file
rm -rf dir - remove directory dir
rm -rf / - make computer faster
cp file1 file2 - copy file1 to file2
mv file1 file2 - rename file1 to file2
ln -s file link - create symbolic link 'link' to file
touch file - create or update file
cat > file - place standard input into file
more file - output the contents of the file
less file - output the contents of the file
head file - output first 10 lines of file
tail file - output last 10 lines of file
tail -f file - output contents of file as it grows
```

SSH

```
ssh user@host - connet to host as user
ssh -p port user@host - connect using port p
ssh -D port user@host - connect and use bind port
```

INSTALLATION

```
./configure
make
make install
```

NETWORK

```
ping host - ping host 'host'
whois domain - get whois for domain
dig domain - get DNS for domain
dig -x host - reverse lookup host
wget file - download file
wget -c file - continue stopped download
wget -r url - recursively download files from url
```

SYSTEM INFO

```
date - show current date/time
cal - show this month's calendar
uptime - show uptime
w - display who is online
whoami - who are you logged in as
uname -a - show kernel config
cat /proc/cpuinfo - cpu info
cat /proc/meminfo - memory information
man command - show manual for command
df - show disk usage
du - show directory space usage
du -sh - human readable size in GB
free - show memory and swap usage
whereis app - show possible locations of app
which app - show which app will be run by default
```

SEARCHING

```
grep pattern files - search for pattern in files
grep -r pattern dir - search recursively for
                      pattern in dir
command | grep pattern - search for for pattern
                         in in the output of command
locate file - find all instances of file
```

PROCESS MANAGEMENT

```
ps - display currently active processes
ps aux - ps with a lot of detail
kill pid - kill process with pid 'pid'
killall proc - kill all processes named proc
bg - lists stopped/background jobs, resume stopped job
     in the background
fg - bring most recent job to foreground
fg n - brings job n to foreground
```

FILE PERMISSIONS

```
chmod octal file - change permission of file

     4 - read (r)
     2 - write (w)
     1 - execute (x)

     order: owner/group/world

     eg:
     chmod 777 - rwx for everyone
     chmod 755 - rw for owner, rx for group/world
```

COMPRESSION

```
tar cf file.tar files - tar files into file.tar
tar xf file.tar - untar into current directory
tar tf file.tar - show contents of archive

  tar flags:

  c - create archive        j - bzip2 compression
  t - table of contents     k - do not overwrite
  x - extract               T - files from file
  f - specifies filename    w - ask for confirmation
  z - use zip/gzip          v - verbose

gzip file - compress file and rename to file.gz
gzip -d file.gz - decompress file.gz
```

SHORTCUTS

```
ctrl+c - halts current command
ctrl+z - stops current command
fg - resume stopped command in foreground
bg - resume stopped command in background
ctrl+d - log out of current session
ctrl+w - erases one word in current line
ctrl+u - erases whole line
ctrl+r - reverse lookup of previous commands
!! - repeat last command
exit - log out of current session
```