

Linux Commands - Part I

1. Grep

1. Search a single pattern in a single file

Code:

```
$ grep 'memory' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep
'memory' sample.txt
An Operating System (OS) is an interface between a computer user and computer ha
rdware. An operating system is a software which performs all the basic tasks lik
e file management, memory management, process management, handling input and out
put, and controlling peripheral devices such as disk drives and printers.
Memory management refers to management of Primary Memory or Main Memory. Main me
mory is a large array of words or bytes where each word or byte has its own addr
ess.
Main memory provides a fast storage that can be accessed directly by the CPU. Fo
r a program to be executed, it must in the main memory. An Operating System does
the following activities for memory management –
    Keeps tracks of primary memory, i.e., what part of it are in use by whom, wh
at part are not in use.
    In multiprogramming, the OS decides which process will get memory when and h
ow much.
    Allocates the memory when a process requests it to do so.
    De-allocates the memory when a process no longer needs it or has been termin
ated.
```

2. List the line numbers (along with the line) the lines containing the pattern in the file

Code:

```
$ grep -n "memory" sample.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -n "memory" sample.txt
1:An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
10:Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
12:Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management -
14:   Keeps tracks of primary memory, i.e., what part of it are in use by whom, what parts are not in use.
16:   In multiprogramming, the OS decides which process will get memory when and how much.
18:   Allocates the memory when a process requests it to do so.
20:   De-allocates the memory when a process no longer needs it or has been terminated.
```

Count the number of matches.

Code:

```
$ grep -c "memory" sample.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -c "memory" sample.txt
7
```

3. Find the matchings pattern regardless of case.

Code:

```
$ grep -i operating sample.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -i operating sample.txt
An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management -
In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling. An Operating System does the following activities for processor management -
An Operating System manages device communication via their respective drivers. It does the following activities for device management -
```

4. Display two lines above and below the matching line

Code:

```
$ grep -A 2 -B 2 -n -i Operating sample.txt | head
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -A 2 -B 2 -n -i Operating sample.txt | head
1:An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
2-
3:Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
4-Definition
5-
6>An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
7-
8-Memory Management
-
10-Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
```

Here I have printed only the first 10 lines of the grep result using **head** to fit within the screen.

Alternatively,

Code:

```
$ grep -C 2 -n -i Operating sample.txt | head
```

Output

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -C 2 -n -i Operating sample.txt | head
1:An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
2-
3:Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
4-Definition
5-
6:An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
7-
8-Memory Management
-
10-Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
```

5. Search multiple patterns in a single file

Code:

```
$ grep -e "memory" -e "management" sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -e "memory" -e "management" sample.txt
An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management -
    Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
    In multiprogramming, the OS decides which process will get memory when and how much.
        Allocates the memory when a process requests it to do so.
        De-allocates the memory when a process no longer needs it or has been terminated.
In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling. An Operating System does the following activities for processor management -
An Operating System manages device communication via their respective drivers. It does the following activities for device management -
```

6. Search pattern in multiple files.

Code:

```
$ grep -i Operating sample.txt hello.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEMS/05/Lab/Week2/part1$ grep -i Operating sample.txt hello.txt
sample.txt:An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
sample.txt:Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
sample.txt:An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
sample.txt:Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management -
sample.txt:In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling. An Operating System does the following activities for processor management -
sample.txt:An Operating System manages device communication via their respective drivers. It does the following activities for device management -
hello.txt:Unix (/ju:niks/; trademarked as UNIX) is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, development starting in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.[3]
hello.txt:Initially intended for use inside the Bell System, AT&T licensed Unix to outside parties in the late 1970s, leading to a variety of both academic and commercial Unix variants from vendors including University of California, Berkeley (BSD), Microsoft (Xenix), Sun Microsystems (SunOS/Solaris), HP/HPE (HP-UX), and IBM (AIX). In the early 1990s, AT&T sold its rights in Unix to Novell, which then sold its Unix business to the Santa Cruz Operation (SCO) in 1995.[4] The UNIX trademark passed to The Open Group, a neutral industry consortium founded in 1996, which allows the use of the mark for certified operating systems that comply with the Single UNIX Specification (SUS). However, Novell continues to own the Unix copyrights, which the SCO Group, Inc. v. Novell, Inc. court case (2010) confirmed.
hello.txt:Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy". According to this philosophy, the operating system should provide a set of simple tools, each of which performs a limited, well-defined function.[5] A unified filesystem (the Unix filesystem) and an inter-process communication mechanism known as "pipes" serve as the main means of communication,[3] and a shell scripting and command language (the Unix shell) is used to combine the tools to perform complex workflows.
hello.txt:Unix distinguishes itself from its predecessors as the first portable operating system: almost the entire operating system is written in the C programming language, which allows Unix to operate on numerous platforms.[6]
```

7. Print only filenames where matching is found

Code:

```
$ grep -l -i Operating sample.txt hello.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -l -i Operating sample.txt hello.txt  
sample.txt  
hello.txt
```

8. Search a pattern in all .txt files.

Code:

```
$ grep -l -i Operating *.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -l -i Operating *.txt  
hello.txt  
sample.txt
```

9. Search in the entire directory

Code:

```
$ grep -l -i Operating *
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ grep -l -i Operating *  
hello.txt  
sample.txt  
grep: screenshots: Is a directory
```

10. List lines without the given pattern

Code:

```
$ grep -n -i Unix hello.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEMS/OS/Lab/Week2/part1$ grep -n -i Unix hello.txt
1:Unix (/ju:niks/; trademarked as UNIX) is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, development starting in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.[3]
3:Initially intended for use inside the Bell System, AT&T licensed Unix to outside parties in the late 1970s, leading to a variety of both academic and commercial Unix variants from vendors including University of California, Berkeley (BSD), Microsoft (Xenix), Sun Microsystems (SunOS/Solaris), HP/HPE (HP-UX), and IBM (AIX). In the early 1990s, AT&T sold its rights in Unix to Novell, which then sold its Unix business to the Santa Cruz Operation (SCO) in 1995.[4] The UNIX trademark passed to The Open Group, a neutral industry consortium founded in 1996, which allows the use of the mark for certified operating systems that comply with the Single UNIX Specification (SUS). However, Novell continues to own the Unix copyrights, which the SCO Group, Inc. v. Novell, Inc. court case (2010) confirmed.
5:Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy". According to this philosophy, the operating system should provide a set of simple tools, each of which performs a limited, well-defined function.[5] A unified filesystem (the Unix filesystem) and an inter-process communication mechanism known as "pipes" serve as the main means of communication,[3] and a shell scripting and command language (the Unix shell) is used to combine the tools to perform complex workflows.
7:Unix distinguishes itself from its predecessors as the first portable operating system: almost the entire operating system is written in the C programming language, which allows Unix to operate on numerous platforms.[6]
```

2. Sed

View portions of the file

1. Print the lines 2 to 7 of the input file

Code:

```
$ sed -n '2,7p' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEMS/OS/Lab/Week2/part1$ sed -n '2,7p' sample.txt
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
Definition

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
```

2. Find the entire file except lines 2 to 7

Code:

```
$ sed '2,7d' sample.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEMS/OS/Lab/Week2/part1$ sed '2,7d' sample.txt
An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
```

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management -

Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.

In multiprogramming, the OS decides which process will get memory when and how much.

Allocates the memory when a process requests it to do so.

De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling. An Operating System does the following activities for processor management -

Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.

Allocates the processor (CPU) to a process.

De-allocates processor when a process is no longer required.

Device Management

```
An Operating System manages device communication via their respective drivers. It does the following activities -
```

3. Display the lines 2 to 7 and 12 to 15 of the input file

Code:

```
$ sed -n -e '2,7p' -e '12,15p' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEMS/OS/Lab/Week2/part1$ sed -n -e '2,7p' -e '12,15p' sample.txt
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
Definition
An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management -
    Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
```

Replace/Substitute and Deleting strings

1. Replace the 2nd occurrence of OS to Operating Systems in sample.txt

Code:

```
$ grep OS sample.txt
$ sed 's/OS/Operating Systems/2' sample.txt | grep "Operating Systems"
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEMS/OS/Lab/Week2/part1$ grep OS sample.txt
An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
    In multiprogramming, the OS decides which process will get memory when and how much.
    In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling. An Operating System does the following activities for processor management -
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEMS/OS/Lab/Week2/part1$ sed 's/OS/Operating Systems/2' sample.txt | grep "Operating Systems"
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/Operating Systems, etc.
```

Alternatively

Code:

```
$ sed -n 's/OS/Operating Systems/2p' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ sed -n 's/OS/Operating Systems/2p' sample.txt
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
```

2. Replace a pattern on the 1st line**Code:**

```
$ sed -n '1 s/OS/Operating Systems/p' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ sed -n '1 s/OS/Operating Systems/p' sample.txt
An Operating System (Operating Systems) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
```

3. Replace the pattern in a range of lines**Code:**

```
$ sed -n '1,5 s/OS/Operating Systems/p' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ sed -n '1,5 s/OS/Operating Systems/p' sample.txt
An Operating System (Operating Systems) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
```

4. Delete “OS” pattern matching line in sample.txt

```
$ sed '/OS/d' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ sed '/OS/d' sample.txt

Definition

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management - 

    Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.

    Allocates the memory when a process requests it to do so.

    De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

    Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.

    Allocates the processor (CPU) to a process.

    De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management -
```

3. Awk

1. Print the lines matching the pattern

Code:

```
$ awk '/Operating Systems/ {print}' sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ awk '/Operating Systems/ {print}' sample.txt
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.
```

2. Print the 1st and 3rd column of input file

Code:

```
$ awk '{print $1,$3}' records.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ awk '{print $1,$3}' records.txt
Name Gender
Adhesh Male
Mukesh Male
Akshun Male
Amy Female
Aparajith Male
```

3. Print the second row of the input file.

Code:

```
$ awk 'NR == 2 {print}' records.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ awk 'NR == 2 {print}' records.txt
Adhesh COE18B001      Male
```

4. Print the 2nd field of the 3rd row in the input file

Code:

```
$ awk 'NR == 3 {print $2}' records.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ awk 'NR == 3 {print $2}' records.txt
CED15I002
```

5. Print the row with a specific value for a column

Code:

```
$ awk '{if($1 == "Adhesh") print}' records.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ awk '{if($1 == "Adhesh") print}' records.txt  
Adhesh COE18B001 Male
```

6. Print the records where the length of column entry is greater than 8 or less than 4**Code:**

```
$ awk '{if((length($1) >= 8) || (length($1) <=3)) print $1}' records.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ awk '{if((length($1) >= 8) || (length($1) <=3)) print $1}' records.txt  
Amy  
Aparajith
```

7. Print numbers from 0 to 10**Code:**

```
$ awk 'BEGIN {for(i=0;i<11;++i) print i}'
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part1$ awk 'BEGIN {for(i=0;i<11;++i) print i}'  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

4. Kill

1. Kill process with given pid

Code:

```
$ kill 3100
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps aux | grep atril
adheshr+ 3100 0.2 1.2 103048680 102340 ? Sl 09:32 0:00 atril /home/adheshreghu/Documents/SEM5/OS/Lab/Week2/LinuxhelpManuals/linuxcommands.pdf
adheshr+ 3106 0.0 0.0 187764 4144 ? Sl 09:32 0:00 /usr/lib/atril/atril
adheshr+ 3472 6.6 1.6 103290948 130524 ? Sl 09:39 0:02 atril /home/adheshreghu/Documents/SEM5/OS/Lab/Week2/LinuxhelpManuals/50_Most_Frequently_Used_UNIX_Linux_Commands_With_Examples.pdf
adheshr+ 3544 0.0 0.0 14428 1080 pts/0 S+ 09:39 0:00 grep --color=auto atril
adheshreghu@adheshreghu-Inspiron-5570:~$ kill 3100
adheshreghu@adheshreghu-Inspiron-5570:~$ ps aux | grep atril
adheshr+ 3106 0.0 0.0 187764 4144 ? Sl 09:32 0:00 /usr/lib/atril/atril
adheshr+ 3472 2.8 1.6 103282752 130528 ? Sl 09:39 0:02 atril /home/adheshreghu/Documents/SEM5/OS/Lab/Week2/LinuxhelpManuals/50_Most_Frequently_Used_UNIX_Linux_Commands_With_Examples.pdf
adheshr+ 3579 0.0 0.0 14428 980 pts/0 S+ 09:40 0:00 grep --color=auto atril
```

2. Print a list of all available signals

Code:

```
$ kill -l
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
 11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
 16) SIGSTKFLT    17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
 21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
 26) SIGVTALRM    27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
 31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
 38) SIGRTMIN+4   39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
 43) SIGRTMIN+9   44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
 48) SIGRTMIN+14  49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
 53) SIGRTMAX-11  54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
 58) SIGRTMAX-6   59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
 63) SIGRTMAX-1   64) SIGRTMAX
```

3. Send a kill signal to kill a process.

Code:

```
$ kill -s SIGKILL 3472
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps aux | grep atril
adheshr+ 3106 0.0 0.0 187764 4144 ? Sl 09:32 0:00 /usr/lib/atril/atrild
adheshr+ 3472 0.6 1.6 103282996 130516 ? Sl 09:39 0:03 atril /home/adheshreghu/Documents/SEM5/OS/Lab/Week2/LinuxhelpManuals/50_Most_Frequently_Used_UNIX_Linux_Commands_With_Examples.pdf
adheshr+ 3707 0.0 0.0 14428 1028 pts/0 R+ 09:47 0:00 grep --color=auto atril
adheshreghu@adheshreghu-Inspiron-5570:~$ kill -s SIGKILL 3472
adheshreghu@adheshreghu-Inspiron-5570:~$ ps aux | grep atril
adheshr+ 3717 0.0 0.0 14428 1060 pts/0 S+ 09:47 0:00 grep --color=auto atril
```

4. Kill all processes in one go.

Code:

```
$ kill -s KILL -1
```

5. PS

1. List processes running in the current shell.

Code:

```
$ ps
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps
 PID TTY      TIME CMD
 3434 pts/0    00:00:00 bash
 4078 pts/0    00:00:00 ps
```

2. Show all current running processes.

Code:

```
$ ps -ax
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps -ax
PID TTY      STAT   TIME COMMAND
 1 ?        Ss      0:01 /sbin/init splash
 2 ?        S       0:00 [kthreadd]
 3 ?        I<     0:00 [rcu_gp]
 4 ?        I<     0:00 [rcu_par_gp]
 5 ?        I       0:00 [kworker/0:0-eve]
 6 ?        I<     0:00 [kworker/0:0H-kb]
 9 ?        I<     0:00 [mm_percpu_wq]
10 ?       S       0:00 [ksoftirqd/0]
11 ?       I       0:03 [rcu_sched]
12 ?       S       0:00 [migration/0]
13 ?       S       0:00 [idle_inject/0]
14 ?       S       0:00 [cpuhp/0]
15 ?       S       0:00 [cpuhp/1]
16 ?       S       0:00 [idle_inject/1]
17 ?       S       0:00 [migration/1]
18 ?       S       0:00 [ksoftirqd/1]
20 ?       I<     0:00 [kworker/1:0H-kb]
21 ?       S       0:00 [cpuhp/2]
22 ?       S       0:00 [idle_inject/2]
23 ?       S       0:00 [migration/2]
24 ?       S       0:00 [ksoftirqd/2]
25 ?       I       0:00 [kworker/2:0-eve]
26 ?       T<     0:00 [kworker/2:0H-kb]
```

3. List processes running as root.

Code:

```
$ ps -u root
```

Similarly to print processes running under another user

Code:

```
$ ps -u adheshreghu
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps -u root
 PID TTY      TIME CMD
  1 ?    00:00:01 systemd
  2 ?    00:00:00 kthreadd
  3 ?    00:00:00 rcu_gp
  4 ?    00:00:00 rcu_par_gp
  5 ?    00:00:00 kworker/0:0-eve
  6 ?    00:00:00 kworker/0:0H-kb
  9 ?    00:00:00 mm_percpu_wq
 10 ?   00:00:00 ksoftirqd/0
 11 ?   00:00:03 rcu_sched
 12 ?   00:00:00 migration/0
 13 ?   00:00:00 idle_inject/0
 14 ?   00:00:00 cpuhp/0
 15 ?   00:00:00 cpuhp/1
 16 ?   00:00:00 idle_inject/1
 17 ?   00:00:00 migration/1
 18 ?   00:00:00 ksoftirqd/1
 20 ?   00:00:00 kworker/1:0H-kb
 21 ?   00:00:00 cpuhp/2
 22 ?   00:00:00 idle_inject/2
 23 ?   00:00:00 migration/2
 24 ?   00:00:00 ksoftirqd/2
 25 ?   00:00:00 kworker/2:0-eve
 26 ?   00:00:00 kworker/2:0H-kb
```

The output of processes running as owner is

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps -u adheshreghu
 PID TTY      TIME CMD
1474 ?    00:00:00 systemd
1475 ?    00:00:00 (sd-pam)
1491 ?    00:00:00 gnome-keyring-d
1494 ?    00:00:00 sh
1509 ?    00:00:01 dbus-daemon
1581 ?    00:00:00 ssh-agent
1599 ?    00:00:00 xfce4-session
1603 ?    00:00:00 xfconfd
1610 ?    00:00:22 xfwm4
1614 ?    00:00:03 xfce4-panel
1616 ?    00:00:05 Thunar
1618 ?    00:00:00 xfdesktop
1619 ?    00:00:01 xfsettingsd
1625 ?    00:00:00 polkit-gnome-au
1627 ?    00:00:00 nm-applet
1629 ?    00:00:00 light-locker
1631 ?    00:00:00 at-spi-bus-laun
1632 ?    00:00:00 blueman-applet
1644 ?    00:00:00 applet.py
1649 ?    00:00:00 update-notifier
1650 ?    00:00:00 xfce4-power-man
1658 ?    00:00:00 dbus-daemon
```

4. Print the thread of a particular process

Code:

```
$ ps -L
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ pidof atril
4319
adhesreghu@adhesreghu-Inspiron-5570:~$ ps -L 4319
 PID  LWP TTY      STAT   TIME COMMAND
 4319  4319 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4322 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4323 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4329 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4330 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4331 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4332 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4333 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4334 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4337 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
 4319  4339 ?      Sl     0:00  atril /home/adhesreghu/Documents/SEM5/OS/Lab/Week2/Linuxhel
```

5. Find the pid of a given process name

Code:

```
$ ps -C atril
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ ps -C atril
 PID  TTY      TIME CMD
 4319 ?      00:00:00 atril
```

Alternatively

Code:

```
$ pidof atril
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ pidof atril
4319
```

6. Find process from pid.

Code:

```
$ ps -fp 4
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps -fp 4
UID      PID  PPID  C STIME TTY          TIME CMD
root      4      2  0 09:23 ?        00:00:00 [rcu_par_gp]
```

7. Find processes matching a pattern

Code:

```
$ ps aux | grep atril
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ps aux | grep atril
adheshrr+ 4319  0.0  1.2 103048636 98256 ?      Sl   10:04   0:01 atril /home/adheshreghu/Documents/SEM5/OS/Lab/Week2/LinuxhelpManuals/linuxcommands.pdf
adheshrr+ 4326  0.0  0.0 187764  4248 ?      Sl   10:04   0:00 /usr/lib/atril/atrild
adheshrr+ 5282  0.0  0.0 14428  1004 pts/0    S+   10:33   0:00 grep --color=auto atril
adheshreghu@adheshreghu-Inspiron-5570:~$
```

6. Mv

1. Move an item from one directory to another

Code:

```
$ mv sample1.txt part1/
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands - Part I.pdf'  sample1.txt
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip      sample2.txt
'Exercise-I LINUX COMMANDS.pdf'   LinuxhelpManuals  samplezip.zip
Input             newfile2.txt                  screenshots
'Linux Commands- Part II.pdf'   part1
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ mv sample1.txt part1/
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls part1/
hello.txt  records.txt  sample1.txt  sample.txt  screenshots
```

2. Rename a file

Code:

```
$ mv newfile.txt newfile2.txt
```

Moving within the same directory would rename the folder to the new filename

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands.pdf'  part1           screenshots
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip        sample1.txt
'Exercise-I LINUX COMMANDS.pdf'   LinuxhelpManuals    sample2.txt
Input             newfile.txt          samplezip.zip
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ mv newfile.txt newfile2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands.pdf'  part1           screenshots
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip        sample1.txt
'Exercise-I LINUX COMMANDS.pdf'   LinuxhelpManuals    sample2.txt
Input             newfile2.txt         samplezip.zip
```

3. Overwrite a file from one directory to another asking user for confirmation.

Code:

```
$ mv -i sample2.txt part1/
```

Output:

Here part1 folder already has sample2.txt, we wish to overwrite the file.

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls && ls part1
Ex-1.txt          'Linux Commands - Part I.pdf'  sample2.txt
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip        samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'   LinuxhelpManuals    screenshots
Input             newfile2.txt
'Linux Commands- Part II.pdf'   part1
hello.txt records.txt sample1.txt sample2.txt sample.txt screenshots
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ mv -i sample2.txt part1/
mv: overwrite 'part1/sample2.txt'? y
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls && ls part1
Ex-1.txt          'Linux Commands- Part II.pdf'  newfile2.txt
'Exercise-I LINUX COMMANDS.docx'  'Linux Commands - Part I.pdf'  part1
'Exercise-I LINUX COMMANDS.pdf'   linuxcp.zip        samplezip.zip
Input             LinuxhelpManuals    screenshots
hello.txt records.txt sample1.txt sample2.txt sample.txt screenshots
```

4. Display informative messages as you move a file from a directory to another

Code:

```
$ mv -v Input part1/
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ mv -v Input part1/
renamed 'Input' -> 'part1/Input'
```

5. Move a directory to another directory (create one if it does not exist).

Code:

```
$ mv part1 part2
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands - Part I.pdf'  part1
'Exercise-I LINUX COMMANDS.docx'      linuxcp.zip    samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'       LinuxhelpManuals screenshots
'Linux Commands- Part II.pdf'        newfile2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ mv part1 part2
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.docx'      linuxcp.zip    samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'       LinuxhelpManuals screenshots
'Linux Commands- Part II.pdf'        newfile2.txt
```

7. Rm

1. Delete a file. (Ask confirmation before deleting)

Code:

```
$ rm -i newfile.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip      samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'   LinuxhelpManuals screenshots
'Linux Commands- Part II.pdf'   newfile2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ rm -i newfile2.txt
rm: remove regular file 'newfile2.txt'? y
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands - Part I.pdf'  samplezip.zip
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip      screenshots
'Exercise-I LINUX COMMANDS.pdf'   LinuxhelpManuals
'Linux Commands- Part II.pdf'   part2
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Delete a directory.

Code:

```
$ rm -r linuxcp
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.docx'  linuxcp          samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'   linuxcp.zip      screenshots
'Linux Commands- Part II.pdf'   LinuxhelpManuals
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ rm -r linuxcp
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands - Part I.pdf'  samplezip.zip
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip      screenshots
'Exercise-I LINUX COMMANDS.pdf'   LinuxhelpManuals
'Linux Commands- Part II.pdf'   part2
```

3. Force delete files (do not prompt if one of the file does not exist)

Code:

```
$ rm -f 1 2 3
```

Here file '3' does not exist.

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
1                                'Linux Commands- Part II.pdf'  part2
2                                'Linux Commands - Part I.pdf'  samplezip.zip
Ex-1.txt                           linuxcp                         screenshots
'Exercise-I LINUX COMMANDS.docx'   linuxcp.zip
'Exercise-I LINUX COMMANDS.pdf'    LinuxhelpManuals

adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ rm -f 1 2 3
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt                           'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.docx'   linuxcp                         samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'    linuxcp.zip                     screenshots
'Linux Commands- Part II.pdf'     LinuxhelpManuals
```

4. Delete all '.v' files in the current directory.

Code:

```
$ rm *.v
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
1.v                               'Linux Commands- Part II.pdf'  part2
2.v                               'Linux Commands - Part I.pdf'  samplezip.zip
Ex-1.txt                           linuxcp                         screenshots
'Exercise-I LINUX COMMANDS.docx'   linuxcp.zip
'Exercise-I LINUX COMMANDS.pdf'    LinuxhelpManuals

adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ rm *.v
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt                           'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.docx'   linuxcp                         samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'    linuxcp.zip                     screenshots
'Linux Commands- Part II.pdf'     LinuxhelpManuals
```

5. Display verbose as you delete

Code:

```
$ rm -v file
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt                           'Linux Commands- Part II.pdf'  LinuxhelpManuals
'Exercise-I LINUX COMMANDS.docx'   'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.pdf'    linuxcp                         samplezip.zip
file                               linuxcp.zip                     screenshots

adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ rm -v file
removed 'file'
```

8. Ln

The **ln** command is used to create either hard or symbolic links.

1. Create a hard link of a file

Code:

```
$ ln file link
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands- Part II.pdf'  LinuxhelpManuals
'Exercise-I LINUX COMMANDS.docx'  'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.pdf'   linuxcp           samplezip.zip
file              linuxcp.zip                   screenshots
link              LinuxhelpManuals
```



```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ln file link
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands- Part II.pdf'  part2
'Exercise-I LINUX COMMANDS.docx'  'Linux Commands - Part I.pdf'  samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'   linuxcp           screenshots
file              linuxcp.zip                   LinuxhelpManuals
link              LinuxhelpManuals
```

2. Create a symbolic link to a file

Code:

```
$ ln -s file hardlink
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          'Linux Commands- Part II.pdf'  part2
'Exercise-I LINUX COMMANDS.docx'  'Linux Commands - Part I.pdf'  samplezip.zip
'Exercise-I LINUX COMMANDS.pdf'   linuxcp           screenshots
file              linuxcp.zip                   LinuxhelpManuals
link              LinuxhelpManuals
```



```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ln -s file hardlink
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
link              'Linux Commands- Part II.pdf'  LinuxhelpManuals
'Exercise-I LINUX COMMANDS.docx'  'Linux Commands - Part I.pdf'  part2
'Exercise-I LINUX COMMANDS.pdf'   linuxcp           samplezip.zip
file              linuxcp.zip                   screenshots
hardlink          linuxcp.zip
```

3. Create a link in a given directory.

Code:

```
$ ln -t part2 file
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ln -t part2  
file  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls part2  
file      Input      sample1.txt  sample.txt  
hello.txt  records.txt  sample2.txt  screenshots
```

9. Cp

1. Copy a file into another file. (Basic Usage)

Code:

```
$ cp file filecp
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls  
Ex-1.txt          linuxcp  
'Exercise-I LINUX COMMANDS.docx'  linuxcp.zip  
'Exercise-I LINUX COMMANDS.pdf'  
file              LinuxhelpManuals  
'Linux Commands- Part II.pdf'    part2  
'Linux Commands - Part I.pdf'   samplezip.zip  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cp file fil  
ecp  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls  
Ex-1.txt          'Linux Commands- Part II.pdf'  part2  
'Exercise-I LINUX COMMANDS.docx'  'Linux Commands - Part I.pdf'  samplezip.zip  
'Exercise-I LINUX COMMANDS.pdf'  
file              linuxcp  
filecp            LinuxhelpManuals  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Copy a file to another directory

Code:

```
$ cp file part2/filecp
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          filecp          linuxcp.zip      screenshots
'Exercise-I LINUX COMMANDS.docx' 'Linux Commands- Part II.pdf' LinuxhelpManuals
'Exercise-I LINUX COMMANDS.pdf'   'Linux Commands - Part I.pdf' part2
file              linuxcp          samplezip.zip
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls part2
hello.txt  Input records.txt  sample1.txt  sample2.txt  screenshots
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cp file part2/filecp
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls part2
filecp  hello.txt  Input records.txt  sample1.txt  sample2.txt  screenshots
```

3. Copy multiple files at the same time to another directory.**Code:**

```
$ cp file file2 part2/
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          file1          linuxcp.zip      screenshots
'Exercise-I LINUX COMMANDS.docx' 'Linux Commands- Part II.pdf' LinuxhelpManuals
'Exercise-I LINUX COMMANDS.pdf'   'Linux Commands - Part I.pdf' part2
file              linuxcp          samplezip.zip
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cp file file1 part2/
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls part2
file  file1  hello.txt  Input  records.txt  sample1.txt  sample2.txt  screenshots
```

4. Copy a directory or folder.**Code:**

```
$ cp -r part2/ part2cp/
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          file1          linuxcp.zip      screenshots
'Exercise-I LINUX COMMANDS.docx' 'Linux Commands- Part II.pdf' LinuxhelpManuals
'Exercise-I LINUX COMMANDS.pdf'   'Linux Commands - Part I.pdf' part2
file              linuxcp          samplezip.zip
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cp -r part2/ part2cp/
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          file1          linuxcp.zip      samplezip.zip
'Exercise-I LINUX COMMANDS.docx' 'Linux Commands- Part II.pdf' LinuxhelpManuals  screenshots
'Exercise-I LINUX COMMANDS.pdf'   'Linux Commands - Part I.pdf' part2
file              linuxcp          part2cp
```

5. Verbose command during copy command

Code:

```
$ cp -v file filecp
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt           file1      linuxcp.zip      screenshots
'Exercise-I LINUX COMMANDS.docx'  'Linux Commands- Part II.pdf'  LinuxhelpManuals
'Exercise-I LINUX COMMANDS.pdf'   'Linux Commands - Part I.pdf'   part2
file               linuxcp      samplezip.zip
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cp -v file filecp
'file' -> 'filecp'
```

10. Find

1. Find files in the current directory by name.

Code:

```
$ find . -name file
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find . -name file
./part2/file
./file
```

2. Find files under the home directory.

Code:

```
$ find /home -name file
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find /home -name part2
find: '/home/adheshreghu/.dbus': Permission denied
/home/adheshreghu/Documents/SEM5/OS/Lab/Week2/part2
```

3. Find all text files in a given directory.

Code:

```
$ find . -type f -name "*.txt"
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find . -type f -name "*.txt"
./Ex-1.txt
./part2/sample.txt
./part2/sample2.txt
./part2/records.txt
./part2/hello.txt
./part2/sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Find all files with 777 permission.

Code:

```
$ find . -perm 777
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents$ find . -perm 777
./NN/CNN/week2
./NN/CNN/week2/__pycache__
./NN/CNN/week2/__pycache__/nsl_utils.cpython-36.pyc
./NN/CNN/week2/ResNets
./NN/CNN/week2/ResNets/__pycache__
./NN/CNN/week2/ResNets/ResNet50.h5
./NN/CNN/week2/ResNets/Residual Networks - v2.ipynb
./NN/CNN/week2/ResNets/datasets
./NN/CNN/week2/ResNets/datasets/test_signs.h5
./NN/CNN/week2/ResNets/datasets/train_signs.h5
./NN/CNN/week2/ResNets/Residual Networks - v1.ipynb
./NN/CNN/week2/ResNets/Residual_Networks_v2a.ipynb
./NN/CNN/week2/ResNets/images
./NN/CNN/week2/ResNets/images/my_image.jpg
./NN/CNN/week2/ResNets/images/idblock2_kiank.png
./NN/CNN/week2/ResNets/images/idblock3_kiank.png
./NN/CNN/week2/ResNets/images/resnet_kiank.png
./NN/CNN/week2/ResNets/images/convblock_kiank.png
./NN/CNN/week2/ResNets/images/skip_connection_kiank.png
./NN/CNN/week2/ResNets/images/vanishing_grad_kiank.png
./NN/CNN/week2/ResNets/images/signs_data_kiank.png
./NN/CNN/week2/ResNets/.ipynb_checkpoints
./NN/CNN/week2/ResNets/.ipynb_checkpoints/Residual Networks - v2-checkpoint.ipynb
./NN/CNN/week2/ResNets/.ipynb_checkpoints/Residual_Networks_v2a-checkpoint.ipynb
./NN/CNN/week2/ResNets/.ipynb_checkpoints/Residual Networks - v1-checkpoint.ipynb
```

5. Find all empty directories

Code:

```
$ find . -type d -empty
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find . -type d -empty
./part2/screenshots
./screenshots/q4
./screenshots/q16
./screenshots/q20
./screenshots/q39
./screenshots/q38
```

6. Find all files based on user

Code:

```
$ find /home -user adheshreghu
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find /home -user adheshreghu | head
/home/adheshreghu
/home/adheshreghu/GlassFish_Server1
/home/adheshreghu/GlassFish_Server1/bin
/home/adheshreghu/GlassFish_Server1/bin/asadmin
/home/adheshreghu/GlassFish_Server1/bin/asadmin.bat
/home/adheshreghu/GlassFish_Server1/README.txt
/home/adheshreghu/GlassFish_Server1/glassfish
/home/adheshreghu/GlassFish_Server1/glassfish/bin
/home/adheshreghu/GlassFish_Server1/glassfish/bin/xjc
/home/adheshreghu/GlassFish_Server1/glassfish/bin/wscompile
```

7. Find the last 50 days of modified files in the home.

Code:

```
$ find ./home -mtime 50
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find /home -mtime 50
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/storage/permanent/chrome/idb/2918063365piupsah.sqlite
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/gmp-widevinecdm
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593666026375.3d8a9357-6813-4949-8049-14dc96fc2a.main.jsonlz4
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593601884815.fca3c107-4425-4aae-bd65-d993ef4c0de0.event.jsonlz4
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593663613927.475b667c-5f51-4631-88ea-6947358d5e02.sync.jsonlz4
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593599093990.1c1e0a14-38ea-4847-8cbb-1550b8598949.modules.jsonlz4
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593663614465.dc7f2374-ec0b-47c8-9858-93e1c3aef9e9.main.jsonlz4
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593611132084.09aaf6d9-07c5-42e2-bec2-a78fa79a668e.main.jsonlz4
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593666025791.8742b04e-7875-4714-a3d6-bca6a2097174.sync.jsonlz4
/home/adheshreghu/.mozilla/firefox/kysuzqma.default-release/dareporting/archived/2020-07/1593611131395.b26fe5f-5c55-4c40-ad8e-5fa05e6b4e13.sync.jsonlz4
find: '/home/adheshreghu/.dbus': Permission denied
/home/adheshreghu/.cache/thumbnails/normal/0a1db0209db94176a7d0185b7e743ed0.png
/home/adheshreghu/.cache/thumbnails/normal/b32f3a735745f81da74458db143bc588.png
/home/adheshreghu/Documents/SEM4/DIS/End semester assignment - 2020-DES203T.pdf
```

7. Find Accessed Files in Last 1 Hour

Code:

```
$ find /home -amin -60.
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find / -amin -60 | head
/
/bin/touch
/bin/bash
/bin/cp
/bin/less
/bin/readlink
/lib
/lib/crda
/lib/crda/pubkeys
/lib/recovery-mode
```

Note:

Mmin amin, cmin

8. Find files between 50 MB to 100 MB.

Code:

```
$ find /home -size +50M -size -100M
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find /home -size +50M -size -100M | head
find: '/home/adhesreghu/.dbus': Permission denied
/home/adhesreghu/idea-IC-193.6494.35/lib/platform-impl.jar
/home/adhesreghu/idea-IC-193.6494.35/jbr/lib/libjfxwebkit.so
/home/adhesreghu/idea-IC-193.6494.35/jbr/lib/modules
/home/adhesreghu/idea-IC-193.6494.35/plugins/android/lib/android.jar
/home/adhesreghu/idea-IC-193.6494.35/plugins/java/lib/java-impl.jar
/home/adhesreghu/snap/netbeans/common/cache/11.3/mavenindex/central/_4k_Lucene50_0.pos
/home/adhesreghu/snap/netbeans/common/cache/11.3/mavenindex/central/_4k.nvd
/home/adhesreghu/snap/netbeans/common/cache/11.3/all-resources.dat
/home/adhesreghu/snap/netbeans/common/cache/11.2/mavenindex/central/_49_Lucene50_0.pos
/home/adhesreghu/snap/netbeans/common/cache/11.2/mavenindex/central/_49.nvd
```

11. Locate

1. Locate a file.

Code:

```
$ locate part2
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ locate part2
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/Input
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/file
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/hello.txt
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/records.txt
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/sample.txt
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/sample1.txt
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/sample2.txt
/home/adhesreghu/Documents/SEM5/OS/Lab/Week2/part2/screenshots
/usr/share/doc/libblockdev-part2
/usr/share/doc/libblockdev-part2/changelog.Debian.gz
/usr/share/doc/libblockdev-part2/copyright
/var/lib/dpkg/info/libblockdev-part2:amd64.list
/var/lib/dpkg/info/libblockdev-part2:amd64.md5sums
/var/lib/dpkg/info/libblockdev-part2:amd64.shlibs
/var/lib/dpkg/info/libblockdev-part2:amd64.symbols
/var/lib/dpkg/info/libblockdev-part2:amd64.triggers
```

2. Limit the search queries to a specific number

Code:

```
$ locate part2 -n 5
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ locate part2 -n 5
/home/adheshreghu/Documents/SEM5/OS/Lab/Week2/part2
/home/adheshreghu/Documents/SEM5/OS/Lab/Week2/part2/Input
/home/adheshreghu/Documents/SEM5/OS/Lab/Week2/part2/file
/home/adheshreghu/Documents/SEM5/OS/Lab/Week2/part2/hello.txt
/home/adheshreghu/Documents/SEM5/OS/Lab/Week2/part2/records.txt
```

3. Print the number of matching entries.

Code:

```
$ locate -c part2
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ locate -c part2
17
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

4. Ignore case sensitive locate outputs

Code:

```
$ locate -i part2
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ locate -i Sample | head
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/__init__.py
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/_pycache_
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/bubble.py
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/clustered_bar.py
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/clustered_column.py
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/excelcolors.py
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/exploded_pie.py
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/filled_radar.py
/home/adheshreghu/.local/lib/python3.6/site-packages/reportlab/graphics/samples/line_chart.py
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

5. Refresh mlocate Database.

Code:

```
$ sudo updatedb
```

12. Diff

1. Compare two files.

Code:

```
$ diff sample1.txt sample2.txt
```

Output:

```
adheshrreghu@adheshrreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff sample1.txt sample2.txt
2,5c2,5
< World
< I
< am
< Adhesh
\ No newline at end of file
---
> People
> I
> Am
> Who
\ No newline at end of file
adheshrreghu@adheshrreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Compare two files in context mode.

Code:

```
$ diff -c sample1.txt sample2.txt
```

Output:

```
adheshrreghu@adheshrreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff -c sample1.txt sample2.txt
*** sample1.txt 2020-08-21 13:49:11.664061161 +0530
--- sample2.txt 2020-08-21 13:49:34.127996967 +0530
*****
*** 1,5 ****
Hello
! World
! I
! am
! Adhesh
\ No newline at end of file
--- 1,5 ----
Hello
! People
! I
! Am
! Who
\ No newline at end of file
```

3. Compare two files in unified mode.

Code:

```
$ diff -u sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff -u sample1.txt sample2.txt
--- sample1.txt 2020-08-21 13:49:11.664061161 +0530
+++ sample2.txt 2020-08-21 13:49:34.127996967 +0530
@@ -1,5 +1,5 @@
Hello
-World
-I
-am
-Adhesh
\ No newline at end of file
+People
+I
+Am
+Who
\ No newline at end of file
```

4. Ignore case sensitive while comparing files.

Code:

```
$ diff -i sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff -i sample1.txt sample2.txt
2,3c2,3
< World
< I
---
> People
> I
5c5
< Adhesh
\ No newline at end of file
---
> Who
\ No newline at end of file
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Report files are identical or not with diff command

Code:

```
$ diff -s sample1.txt sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff -s sample1.txt sample1cp.txt
Files sample1.txt and sample1cp.txt are identical
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

6. Ignore all blank lines during comparison

Code:

```
$ diff -B sample1.txt sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff -B sample1.txt sample2.txt
2,5c2,5
< World
< I
< am
< Adhesh
\ No newline at end of file
---
> People
> I
> Am
> Who
\ No newline at end of file
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

6. Compare two directories

Code:

```
$ diff -q linuxcp part2/
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff -q linuxcp part2/
Only in linuxcp: 50_Most_Frequently_Used_UNIX _ Linux Commands With Examples.pdf
Only in part2/: file
Only in part2/: file1
Only in part2/: hello.txt
Only in part2/: Input
Only in linuxcp: LinuxCommand.pdf
Only in linuxcp: linuxcommands.pdf
Only in linuxcp: other source.docx
Only in part2/: records.txt
Only in part2/: sample1.txt
Only in part2/: sample2.txt
Only in part2/: sample.txt
Only in part2/: screenshots
Only in linuxcp: TLCL-13.07.pdf
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

7. Ignore White Space While Comparing files with diff

Code:

```
$ diff -b sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ diff -b sample1.txt sample2.txt
2c2
< World
---
> People
4,5c4,5
< am
< Adhesh
\ No newline at end of file
---
> Am
> Who
\ No newline at end of file
```

13. Sort

1. Sort the contents of a file

Code:

```
$ sort sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sort sample1.txt
Adhesh
am
Hello
I
World
```

2. Sort the contents of a file in reverse order.

Code:

```
$ sort -r sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sort -r sample1.txt
World
I
Hello
am
Adhesh
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Sort the contents of a file based on the 2nd column.

Code:

```
$ sort -nk2 sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sort -nk2 sample1.txt
Hello    1
Adhesh   12
I        19
World    55
am       77
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Remove duplicates from sort.

Code:

```
$ sort -u sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1.txt
Hello    1
World    55
I        19
Adhesh   12
am       77
Adhesh   12
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sort -u sample1.txt
Adhesh   12
am       77
Hello    1
I        19
World    55
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Sort the contents of two files in a go to stdout.

Code:

```
$ sort sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sort sample1.txt sample2.txt
Adhesh 12
Adhesh 12
Am
am    77
Hello
Hello  1
I
I     19
People
Who
World  55
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

14. Pwd

1. Print the current working directory**Code:**

```
$ pwd
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ pwd
/home/adheshreghu/Documents/SEM5/OS/Lab/Week2
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Print working directory from environment even if it contains symlinks.**Code:**

```
$ pwd -L
```

3. Print actual physical current working directory by resolving all symbolic links.**Code:**

```
$ pwd -P
```

15. Cd

1. Change from current directory to /usr/local.

Code:

```
$ cd /usr/local
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ cd /usr/local  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local$ pwd  
/usr/local
```

2. Change from current directory to /usr/local/lib using absolute path.

Code:

```
$ cd /usr/local/lib
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr/local$ cd /usr/local/lib  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local/lib$ pwd  
/usr/local/lib  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local/lib$ █
```

3. Change from current directory to /usr/local/lib using relative path.

Code:

```
$ cd lib
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr/local$ cd lib  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local/lib$ pwd  
/usr/local/lib  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local/lib$ █
```

4. Go back to the previous directory.

Code:

```
$ cd -
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr/local/lib$ pwd  
/usr/local/lib  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local/lib$ cd -  
/usr/local  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local$ pwd  
/usr/local
```

4. Go to parent directory

Code:

```
$ cd ..
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr/local$ pwd  
/usr/local  
adheshreghu@adheshreghu-Inspiron-5570:/usr/local$ cd ..  
adheshreghu@adheshreghu-Inspiron-5570:/usr$ pwd  
/usr
```

5. Move two directories up from where you are now.

Code:

```
$ cd ../../..
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr/local/lib$ cd ../../..  
adheshreghu@adheshreghu-Inspiron-5570:/usr$ pwd  
/usr  
adheshreghu@adheshreghu-Inspiron-5570:/usr$
```

6. Goto home directory.

Code:

```
$ cd
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr/local$ cd  
adheshreghu@adheshreghu-Inspiron-5570:~$ pwd  
/home/adheshreghu  
adheshreghu@adheshreghu-Inspiron-5570:~$
```

16. Top

1. Display linux process using top command.

Code:

```
$ top
```

Output:

top - 14:36:05 up 5:15, 1 user, load average: 1.60, 1.57, 1.64													
Tasks: 244 total, 2 running, 171 sleeping, 3 stopped, 0 zombie													
%Cpu(s): 10.1 us, 3.5 sy, 0.0 ni, 84.1 id, 1.6 wa, 0.0 hi, 0.8 si, 0.0 st													
KiB Mem : 8046912 total, 413472 free, 4087916 used, 3545524 buff/cache													
KiB Swap: 7812092 total, 7811568 free, 524 used. 3198004 avail Mem													
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND		
2043	adheshr+	20	0	3684432	689616	186804	S	52.6	8.6	54:11.80	Web Content		
1813	adheshr+	20	0	4208988	939608	288940	S	21.5	11.7	133:59.34	MainThread		
26136	adheshr+	20	0	3033372	353116	169764	S	17.2	4.4	0:17.90	Web Content		
1127	root	20	0	2317996	181368	143096	R	9.6	2.3	32:01.43	Xorg		
2172	adheshr+	20	0	434664	42248	34388	S	5.0	0.5	32:56.68	RDD Process		
26260	adheshr+	20	0	409072	20640	15808	S	4.0	0.3	0:00.21	gnome-screensho		
1655	adheshr+	9	-11	1062336	22316	17948	S	1.3	0.3	13:10.32	pulseaudio		
1574	adheshr+	20	0	191740	25944	20072	S	1.0	0.3	3:03.49	xfwm4		
25797	adheshr+	20	0	2809664	230776	123288	S	1.0	2.9	0:15.37	Web Content		
26259	adheshr+	20	0	44208	4096	3408	R	1.0	0.1	0:00.10	top		
2454	adheshr+	20	0	3427996	502544	158700	S	0.7	6.2	5:25.25	Web Content		
2501	adheshr+	20	0	3268308	394768	124492	S	0.7	4.9	26:57.82	Web Content		
22046	adheshr+	20	0	3638352	959544	177496	S	0.7	11.9	17:36.70	Web Content		
11	root	20	0	0	0	0	I	0.3	0.0	0:27.54	rcu_sched		
1644	adheshr+	20	0	380928	18300	15316	S	0.3	0.2	0:07.13	xfsettingsd		
1711	adheshr+	20	0	694004	34072	27376	S	0.3	0.4	0:24.88	panel-9-pulseau		
13632	adheshr+	20	0	2868020	285640	107032	S	0.3	3.5	1:13.71	Web Content		
22506	adheshr+	20	0	535436	38952	27244	S	0.3	0.5	0:23.75	xfce4-terminal		
25508	root	20	0	0	0	0	I	0.3	0.0	0:03.17	kworker/6:1-eve		
26233	root	20	0	0	0	0	I	0.3	0.0	0:00.14	kworker/u16:2-i		
1	root	20	0	225664	9460	6792	S	0.0	0.1	0:01.34	systemd		
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd		

2. Display specific user process

Code:

```
$ top -u adheshreghu
```

Output:

```
top - 14:37:19 up 5:16, 1 user, load average: 1.31, 1.51, 1.62
Tasks: 244 total, 1 running, 172 sleeping, 3 stopped, 0 zombie
%Cpu(s): 9.0 us, 3.5 sy, 0.0 ni, 85.9 id, 0.5 wa, 0.0 hi, 1.1 si, 0.0 st
KiB Mem : 8046912 total, 717324 free, 3787220 used, 3542368 buff/cache
KiB Swap: 7812092 total, 7811568 free, 524 used. 3502160 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2043	adheshr+	20	0	3681360	672124	186804	S	52.8	8.4	54:48.94	Web Content
1813	adheshr+	20	0	4210812	961548	307584	S	23.4	11.9	134:23.69	MainThread
26136	adheshr+	20	0	3034588	364024	170364	S	14.5	4.5	0:32.50	Web Content
2172	adheshr+	20	0	434664	42248	34388	S	4.6	0.5	32:59.83	RDD Process
26285	adheshr+	20	0	409080	20296	15452	S	3.3	0.3	0:00.20	gnome-screensho
22046	adheshr+	20	0	3348144	662516	178032	S	2.6	8.2	17:47.13	Web Content
1574	adheshr+	20	0	191740	25944	20072	S	1.0	0.3	3:04.17	xfwm4
1655	adheshr+	9	-11	1062336	22316	17948	S	1.0	0.3	13:11.14	pulseaudio
25797	adheshr+	20	0	2809664	229588	123288	S	1.0	2.9	0:15.96	Web Content
2454	adheshr+	20	0	3427996	502592	158700	S	0.7	6.2	5:25.77	Web Content
26284	adheshr+	20	0	44208	4156	3468	R	0.7	0.1	0:00.09	top
1975	adheshr+	20	0	2716144	210716	92560	S	0.3	2.6	1:19.42	WebExtensions
13632	adheshr+	20	0	2868020	284172	107032	S	0.3	3.5	1:14.01	Web Content
1434	adheshr+	20	0	77196	8288	6672	S	0.0	0.1	0:00.11	systemd
1435	adheshr+	20	0	114272	2876	84	S	0.0	0.0	0:00.00	(sd-pam)
1451	adheshr+	20	0	281272	7932	6968	S	0.0	0.1	0:00.14	gnome-keyring-d
1454	adheshr+	20	0	4624	1736	1640	S	0.0	0.0	0:00.03	sh
1471	adheshr+	20	0	50836	5192	3780	S	0.0	0.1	0:07.00	dbus-daemon
1545	adheshr+	20	0	11300	320	0	S	0.0	0.0	0:00.07	ssh-agent
1563	adheshr+	20	0	332000	14920	13080	S	0.0	0.2	0:00.11	xfce4-session
1567	adheshr+	20	0	59392	5280	4576	S	0.0	0.1	0:00.14	xfconfd
1578	adheshr+	20	0	270932	23080	18852	S	0.0	0.3	0:29.12	xfce4-panel

3. Highlight process running on top.**Code:**

```
$ top
```

Then Press **z**.

Output:

top - 14:38:56 up 5:18, 1 user, load average: 1.15, 1.40, 1.56											
Tasks: 245 total, 1 running, 172 sleeping, 3 stopped, 0 zombie											
%Cpu(s): 11.3 us, 3.8 sy, 0.0 ni, 84.2 id, 0.0 wa, 0.0 hi, 0.7 si, 0.0 st											
KiB Mem : 8046912 total, 611536 free, 3839060 used, 3596316 buff/cache											
KiB Swap: 7812092 total, 7811568 free, 524 used 3395828 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2172	adheshr+	20	0	453676	100460	53404	S	53.1	1.2	33:39.84	RDD Process
2043	adheshr+	20	0	3668292	569828	170616	S	35.3	7.1	55:23.14	Web Content
1813	adheshr+	20	0	4281656	966368	285224	S	24.4	12.0	135:00.58	MainThread
26136	adheshr+	20	0	3035612	360520	170984	S	14.9	4.5	0:48.72	Web Content
1655	adheshr+	9	-11	1062336	22324	17956	S	1.3	0.3	13:12.18	pulseaudio
26371	adheshr+	20	0	409076	20628	15784	S	1.3	0.3	0:00.15	gnome-screensho
2454	adheshr+	20	0	3428420	507728	159124	S	1.0	6.3	5:26.92	Web Content
22046	adheshr+	20	0	3348088	687204	178056	S	1.0	8.5	18:07.84	Web Content
25797	adheshr+	20	0	2825576	241360	140204	S	1.0	3.0	0:17.18	Web Content
26284	adheshr+	20	0	44208	4156	3468	R	1.0	0.1	0:00.82	top
1574	adheshr+	20	0	191748	25944	20972	S	0.3	0.3	3:05.24	xfwm4
13632	adheshr+	20	0	2918856	336428	157924	S	0.3	4.2	1:14.52	Web Content
1434	adheshr+	20	0	77196	8288	6672	S	0.0	0.1	0:00.11	systemd
1435	adheshr+	20	0	114272	2876	84	S	0.0	0.0	0:00.00	(sd-pam)
1451	adheshr+	20	0	281272	7932	6968	S	0.0	0.1	0:00.14	gnome-keyring-d
1454	adheshr+	20	0	4624	1736	1640	S	0.0	0.0	0:00.03	sh
1471	adheshr+	20	0	50836	5192	3780	S	0.0	0.1	0:07.05	dbus-daemon
1545	adheshr+	20	0	11308	320	0	S	0.0	0.0	0:00.07	ssh-agent
1563	adheshr+	20	0	332000	14920	13080	S	0.0	0.2	0:00.11	xfce4-session
1567	adheshr+	20	0	59392	5280	4576	S	0.0	0.1	0:00.14	xfconfd
1578	adheshr+	20	0	270932	23080	18852	S	0.0	0.3	0:29.62	xfce4-panel
1580	adheshr+	20	0	525248	45696	27768	S	0.0	0.6	0:33.22	Thunar

4. Show absolute path of processes listed using top command**Code:**

```
$ top
```

Then Press c.

Output:

```
top - 14:43:05 up 5:22, 1 user, load average: 1.89, 1.69, 1.65
Tasks: 260 total, 1 running, 171 sleeping, 3 stopped, 0 zombie
%Cpu(s): 13.9 us, 3.8 sy, 0.0 ni, 80.4 id, 0.7 wa, 0.0 hi, 1.3 si, 0.0 st
KiB Mem : 8046912 total, 918212 free, 3530932 used, 3597768 buff/cache
KiB Swap: 7812092 total, 7811568 free, 524 used. 3704096 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
2172 adheshr+ 20 0 442304 61200 42032 S 38.0 0.8 35:39.13 /usr/lib/firefox/firefox -contentproc -p+
2043 adheshr+ 20 0 3658724 595684 169372 S 30.0 7.4 56:43.61 /usr/lib/firefox/firefox -contentproc -c+
1813 adheshr+ 20 0 4261240 752616 255760 S 24.1 9.4 136:41.35 /usr/lib/firefox/firefox
2501 adheshr+ 20 0 3286740 415140 142944 S 19.1 5.2 27:13.04 /usr/lib/firefox/firefox -contentproc -c+
26136 adheshr+ 20 0 3035612 367640 170504 S 13.2 4.6 1:38.09 /usr/lib/firefox/firefox -contentproc -c+
1127 root 20 0 2314952 179408 141136 S 10.9 2.2 32:58.05 /usr/lib/xorg/Xorg -core :0 -seat seat0 +
26453 adheshr+ 20 0 409064 20620 15796 S 4.6 0.3 0:00.24 gnome-screenshot -a
1574 adheshr+ 20 0 191740 25944 20072 S 1.3 0.3 3:07.78 xfwm4 --replace
1655 adheshr+ 9 -11 1062336 22336 17968 S 1.3 0.3 13:14.86 /usr/bin/pulseaudio --start --log-target=
1885 adheshr+ 20 0 2868188 225148 135176 S 1.3 2.8 1:02.31 /usr/lib/firefox/firefox -contentproc -c+
26442 adheshr+ 20 0 44236 4048 3364 R 1.0 0.1 0:00.35 top
22046 adheshr+ 20 0 3331704 626380 178060 S 0.7 7.8 19:00.55 /usr/lib/firefox/firefox -contentproc -c+
1153 mysql 20 0 1360808 181588 15832 S 0.3 2.3 0:18.55 /usr/sbin/mysql --daemonize --pid-file=+
1644 adheshr+ 20 0 380928 18300 15316 S 0.3 0.2 0:07.50 xfsettingsd
2454 adheshr+ 20 0 3428420 507068 159124 S 0.3 6.3 5:28.61 /usr/lib/firefox/firefox -contentproc -c+
22506 adheshr+ 20 0 535436 38952 27244 S 0.3 0.5 0:25.82 /usr/bin/xfce4-terminal
25073 root 20 0 0 0 0 I 0.3 0.0 0:02.63 [kworker/u16:1-p]
25797 adheshr+ 20 0 2808640 219384 123252 S 0.3 2.7 0:19.83 /usr/lib/firefox/firefox -contentproc -c+
26233 root 20 0 0 0 0 I 0.3 0.0 0:00.75 [kworker/u16:2-i]
26406 root 20 0 0 0 0 I 0.3 0.0 0:00.53 [kworker/6:11-mm]
1 root 20 0 225664 9460 6792 S 0.0 0.1 0:01.35 /sbin/init splash
2 root 20 0 0 0 0 S 0.0 0.0 0:00.01 [kthreadd]
```

5. Kill a process using top.

Code:

```
$ top
```

Then Press k.

```
top - 14:45:21 up 5:24, 1 user, load average: 1.70, 1.71, 1.66
Tasks: 263 total, 1 running, 174 sleeping, 3 stopped, 0 zombie
%Cpu(s): 15.1 us, 3.7 sy, 0.0 ni, 78.4 id, 1.3 wa, 0.0 hi, 1.5 si, 0.0 st
KiB Mem : 8046912 total, 815844 free, 3635792 used, 3595276 buff/cache
KiB Swap: 7812092 total, 7811568 free, 524 used. 3629640 avail Mem
PID to signal/kill [default pid = 2172] 26481
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
2172 adheshr+ 20 0 442316 60256 42044 S 70.6 0.7 36:32.58 RDD Process
2043 adheshr+ 20 0 3652592 589316 169440 S 23.5 7.3 57:25.85 Web Content
1813 adheshr+ 20 0 4191300 748576 264944 S 17.6 9.3 137:24.70 MainThread
26136 adheshr+ 20 0 3035612 371332 170504 S 11.8 4.6 1:57.25 Web Content
26525 adheshr+ 20 0 44208 3980 3300 R 11.8 0.0 0:00.05 top
1127 root 20 0 2385432 208420 170148 S 5.9 2.6 33:12.63 Xorg
25797 adheshr+ 20 0 2808640 219780 123292 S 5.9 2.7 0:21.03 Web Content
1 root 20 0 225664 9460 6792 S 0.0 0.1 0:01.35 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kb
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
10 root 20 0 0 0 0 S 0.0 0.0 0:00.95 ksoftirqd/0
11 root 20 0 0 0 0 I 0.0 0.0 0:28.13 rcu_sched
12 root rt 0 0 0 0 S 0.0 0.0 0:00.06 migration/0
13 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0
14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
15 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
16 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/1
17 root rt 0 0 0 0 S 0.0 0.0 0:00.19 migration/1
18 root 20 0 0 0 0 S 0.0 0.0 0:00.73 ksoftirqd/1
```

17. df

1. Check File System Disk Space Usage.

Code:

```
$ df
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            3999504      0  3999504   0% /dev
tmpfs           804692     1644   803048   1% /run
/dev/sda5      313797912 184178060 113610076  62% /
tmpfs           4023456    245704  3777752   7% /dev/shm
tmpfs            5120        4     5116   1% /run/lock
tmpfs           4023456      0  4023456   0% /sys/fs/cgroup
/dev/loop2       98944     98944      0 100% /snap/core/9804
/dev/loop0       26368     26368      0 100% /snap/heroku/3960
/dev/loop1       452096    452096      0 100% /snap/netbeans/30
/dev/loop4       447744    447744      0 100% /snap/netbeans/24
/dev/loop3       26368     26368      0 100% /snap/heroku/3962
/dev/loop5       99328     99328      0 100% /snap/core/9665
/dev/sda3       524268    12784   511484   3% /boot/efi
tmpfs           804688      20   804668   1% /run/user/1000
adheshreghu@adheshreghu-Inspiron-5570:/usr$
```

2. Display Information of all File System Disk Space Usage

Code:

```
$ df -a
```

Output:

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
sysfs	0	0	0	-	/sys
proc	0	0	0	-	/proc
udev	3999504	0	3999504	0%	/dev
devpts	0	0	0	-	/dev/pts
tmpfs	804692	1644	803048	1%	/run
/dev/sda5	313797912	184178124	113610012	62%	/
securityfs	0	0	0	-	/sys/kernel/security
tmpfs	4023456	207604	3815852	6%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	4023456	0	4023456	0%	/sys/fs/cgroup
cgroup	0	0	0	-	/sys/fs/cgroup/unified
cgroup	0	0	0	-	/sys/fs/cgroup/systemd
pstore	0	0	0	-	/sys/fs/pstore
efivarfs	0	0	0	-	/sys/firmware/efi/efivars
cgroup	0	0	0	-	/sys/fs/cgroup/devices
cgroup	0	0	0	-	/sys/fs/cgroup/hugetlb
cgroup	0	0	0	-	/sys/fs/cgroup/net_cls,net_prio
cgroup	0	0	0	-	/sys/fs/cgroup/pids
cgroup	0	0	0	-	/sys/fs/cgroup/blkio
cgroup	0	0	0	-	/sys/fs/cgroup/rdma
cgroup	0	0	0	-	/sys/fs/cgroup/freezer
cgroup	0	0	0	-	/sys/fs/cgroup/cpu,cpuacct
cgroup	0	0	0	-	/sys/fs/cgroup/cpuset
cgroup	0	0	0	-	/sys/fs/cgroup/memory
cgroup	0	0	0	-	/sys/fs/cgroup/perf_event
systemd-1	-	-	-	-	/proc/sys/fs/binfmt_misc
mqueue	0	0	0	-	/dev/mqueue
debugfs	0	0	0	-	/sys/kernel/debug
hugetlbfs	0	0	0	-	/dev/hugepages
fusectl	0	0	0	-	/sys/fs/fuse/connections
configfs	0	0	0	-	/sys/kernel/config
/dev/loop2	98944	98944	0	100%	/snap/core/9804
/dev/loop0	26368	26368	0	100%	/snap/heroku/3960
/dev/loop1	452096	452096	0	100%	/snap/netbeans/30
/dev/loop4	447744	447744	0	100%	/snap/netbeans/24
/dev/loop3	26368	26368	0	100%	/snap/heroku/3962
/dev/loop5	99328	99328	0	100%	/snap/core/9665
/dev/sda3	524268	12784	511484	3%	/boot/efi
binfmt_misc	0	0	0	-	/proc/sys/fs/binfmt_misc
tmpfs	804688	20	804668	1%	/run/user/1000
qvfd-fuse	0	0	0	-	/run/user/1000/qvfd

3. Show Disk Space Usage in Human Readable Format

Code:

```
$ df -h
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G   0    3.9G  0% /dev
tmpfs           786M  1.7M  785M  1% /run
/dev/sda5        300G  176G  109G  62% /
tmpfs           3.9G  196M  3.7G  5% /dev/shm
tmpfs           5.0M  4.0K  5.0M  1% /run/lock
tmpfs           3.9G   0    3.9G  0% /sys/fs/cgroup
/dev/loop2       97M   97M   0  100% /snap/core/9804
/dev/loop0       26M   26M   0  100% /snap/heroku/3960
/dev/loop1       442M  442M   0  100% /snap/netbeans/30
/dev/loop4       438M  438M   0  100% /snap/netbeans/24
/dev/loop3       26M   26M   0  100% /snap/heroku/3962
/dev/loop5       97M   97M   0  100% /snap/core/9665
/dev/sda3        512M  13M   500M  3% /boot/efi
tmpfs           786M  20K   786M  1% /run/user/1000
adheshreghu@adheshreghu-Inspiron-5570:/usr$
```

4. Display Information of /home File System

Code:

```
$ df -h /home
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:/usr$ df -h /home
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda5        300G  176G  109G  62% /
adheshreghu@adheshreghu-Inspiron-5570:/usr$
```

5. Display Information of File System in MB

Code:

```
$ df -h -m
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:/usr$ df -h -m
Filesystem      1M-blocks  Used Available Use% Mounted on
udev            3906       0     3906   0% /dev
tmpfs           786        2      785   1% /run
/dev/sda5      306444  179860   110949  62% /
tmpfs           3930      258     3672   7% /dev/shm
tmpfs            5         1       5   1% /run/lock
tmpfs           3930       0     3930   0% /sys/fs/cgroup
/dev/loop2        97        97       0 100% /snap/core/9804
/dev/loop0        26        26       0 100% /snap/heroku/3960
/dev/loop1        442      442       0 100% /snap/netbeans/30
/dev/loop4        438      438       0 100% /snap/netbeans/24
/dev/loop3        26        26       0 100% /snap/heroku/3962
/dev/loop5        97        97       0 100% /snap/core/9665
/dev/sda3        512       13      500   3% /boot/efi
tmpfs           786        1      786   1% /run/user/1000
adhesreghu@adhesreghu-Inspiron-5570:/usr$
```

6. Display File System Inodes**Code:**

```
$ df -i
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:/usr$ df -i
Filesystem      INodes   IUsed   IFree IUse% Mounted on
udev            999876    665   999211   1% /dev
tmpfs          1005864   1090  1004774   1% /run
/dev/sda5      19996672 1270140 18726532   7% /
tmpfs          1005864    210   1005654   1% /dev/shm
tmpfs          1005864      5   1005859   1% /run/lock
tmpfs          1005864     18   1005846   1% /sys/fs/cgroup
/dev/loop2      12796    12796      0 100% /snap/core/9804
/dev/loop0      28237    28237      0 100% /snap/heroku/3960
/dev/loop1      4407     4407      0 100% /snap/netbeans/30
/dev/loop4      4402     4402      0 100% /snap/netbeans/24
/dev/loop3      28237    28237      0 100% /snap/heroku/3962
/dev/loop5      12862    12862      0 100% /snap/core/9665
/dev/sda3          0        0       0   - /boot/efi
tmpfs          1005864     28   1005836   1% /run/user/1000
adhesreghu@adhesreghu-Inspiron-5570:/usr$
```

7. Exclude Certain File System Type**Code:**

```
$ df -x ext3
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ df -x ext3
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            3999504       0   3999504   0% /dev
tmpfs           804692     1644   803048   1% /run
/dev/sda5      313797912 184176852 113611284 62% /
tmpfs           4023456   237092   3786364   6% /dev/shm
tmpfs            5120        4     5116   1% /run/lock
tmpfs           4023456       0   4023456   0% /sys/fs/cgroup
/dev/loop2       98944     98944       0 100% /snap/core/9804
/dev/loop0       26368     26368       0 100% /snap/heroku/3960
/dev/loop1       452096    452096       0 100% /snap/netbeans/30
/dev/loop4       447744    447744       0 100% /snap/netbeans/24
/dev/loop3       26368     26368       0 100% /snap/heroku/3962
/dev/loop5       99328     99328       0 100% /snap/core/9665
/dev/sda3       524268    12784   511484   3% /boot/efi
tmpfs           804688       20   804668   1% /run/user/1000
adheshreghu@adheshreghu-Inspiron-5570:~$
```

Note: `-t` is used to include a certain file system type.

18. du

1. Check information on disk usage.

Code:

```
$ du
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ du | head
4      ./part2/screenshots
32     ./part2
168    ./screenshots/q8
52     ./screenshots/q2
20     ./screenshots/q13
60     ./screenshots/q28
88     ./screenshots/q22
192    ./screenshots/q18
16     ./screenshots/q21
20     ./screenshots/q5
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Check information on disk usage in human readable format.

Code:

```
$ du -h
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ du -h | head
4.0K  ./part2/screenshots
32K   ./part2
168K  ./screenshots/q8
52K   ./screenshots/q2
20K   ./screenshots/q13
60K   ./screenshots/q28
88K   ./screenshots/q22
192K  ./screenshots/q18
16K   ./screenshots/q21
20K   ./screenshots/q5
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Get summary of grand total disk usage of a directory.

Code:

```
$ du -s .
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ du -s part2/
32    part2/
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Get information on disk usage on all files and directories.

Code:

```
$ du -a
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ du -a | head
0     ./filecp
1740   ./Linux Commands - Part I.pdf
0     ./file1
8      ./Ex-1.txt
2304   ./Linux Commands- Part II.pdf
4     ./part2/Input
0     ./part2/file1
4     ./part2/sample.txt
0     ./part2/file
4     ./part2/screenshots
```

5. Get summary of total disk usage of a directory.

Code:

```
$ du -c
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part2$ du -c
4      ./screenshots
32      .
32      total
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/part2$
```

19. cat

1. Display Contents of File

Code:

```
$ cat sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. View Contents of Multiple Files in terminal

Code:

```
$ cat sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1.txt sample2.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12
Hello
People
I
Am
Who
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Display Line Numbers in File

Code:

```
$ cat -n sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat -n sample1.txt
 1 Hello    1
 2 World   55
 3 I        19
 4 Adhesh   12
 5 am      77
 6 Adhesh  12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

4. Display \$ at the EOF

Code:

```
$ cat -e sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat -e sample1.txt
Hello  1$
World  55$
I      19$
Adhesh 12$
am    77$
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

5. Display Tab separated Lines in File

Code:

```
$ cat -T sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat -T sample1.txt
Hello^I1
World^I55
I^I19
Adhesh^I12
am^I77
Adhesh^I12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

6. Use Standard Output with Redirection Operator

Code:

```
$ cat sample1.txt > sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1.txt > sample1cp.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1cp.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

7. Appending Standard Output with Redirection Operator

Code:

```
$ cat sample1.txt >> sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1.txt >> sample1cp.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1cp.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

8. Redirecting Standard Input with Redirection Operator

Code:

```
$ cat < sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat < sample1.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

20. Mkdir

1. Create a directory

Code:

```
$ mkdir newfolder
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          file1           linuxcp           sample1cp.txt   screenshots
'Exercise-I LINUX COMMANDS.docx'  filecp          linuxcp.zip      sample1.txt
'Exercise-I LINUX COMMANDS.pdf'   'Linux Commands- Part II.pdf'  LinuxhelpManuals  sample2.txt
file              'Linux Commands - Part I.pdf'  part2          samplezip.zip
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ mkdir newfolder
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls
Ex-1.txt          file1           linuxcp           part2          samplezip.zip
'Exercise-I LINUX COMMANDS.docx'  filecp          linuxcp.zip      screenshots
'Exercise-I LINUX COMMANDS.pdf'   'Linux Commands- Part II.pdf'  LinuxhelpManuals  sample1.txt
file              'Linux Commands - Part I.pdf'  newfolder       sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Create a directory structure and make sure parent directories are created in the process.

Code:

```
$ mkdir -p nf1/nf2/nf3
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$ mkdir -p nf1/nf2/nf3
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$ ls -R
.:
nf1
./nf1:
nf2
./nf1/nf2:
nf3
./nf1/nf2/nf3:
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$
```

3. Set permission for the folder being created.

Code:

```
$ mkdir -m 777 nf4
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$ mkdir -m 777 nf4
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$ ls -l
total 8
drwxrwxr-x 3 adheshreghu adheshreghu 4096 Aug 21 16:39 nf1
drwxrwxrwx 2 adheshreghu adheshreghu 4096 Aug 21 16:40 nf4
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$
```

4. Verbose details with mkdir.**Code:**

```
$ mkdir -v nf5
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$ mkdir -v nf5
mkdir: created directory 'nf5'
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$ ls
nf1 nf4 nf5
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2/newfolder$
```

21. Head

1. Print first 10 lines of a file.**Code:**

```
$ head sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head sample1cp.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12Hello 1
World 55
I 19
Adhesh 12
am 77
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Print first 7 lines of a file.

Code:

```
$ head -n 7 sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head -n 7 sample1cp.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12Hello 1
World 55
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Print the first 12 bytes of a file.

Code:

```
$ head -c 12 sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head -c 12 sample1cp.txt
Hello 1
Worldadheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Print the head of more than one file.

Code:

```
$ head -q sample1cp.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head -q sample1cp.txt sample2.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12Hello 1
World 55
I 19
Adhesh 12
am 77
Hello
People
I
Am
Whoadheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Print the filename followed by the head of the file.

Code:

```
$ head -v sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head -v sample1cp.txt
==> sample1cp.txt <==
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12Hello 1
World 55
I 19
Adhesh 12
am 77
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

6. Print lines between 6 to 11

Code:

```
$ head -n 11 index.txt | tail +6
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head -n 11
index.txt
1 Ex-1.txt
2 Exercise-I LINUX COMMANDS.docx
3 Exercise-I LINUX COMMANDS.pdf
4 f1
5 f2
6 index.txt
7 Linux Commands- Part II.pdf
8 Linux Commands - Part I.pdf
9 linuxcp
10 linuxcp.zip
11 LinuxhelpManuals
```

22. Tail

1. Print the last 10 lines of a file.

Code:

```
$ tail sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ tail sample1cp.txt
Adhesh 12Hello
People
I
Am
WhoHello      1
World  55
I      19
Adhesh 12
am    77
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Print the last 7 lines of a file.

Code:

```
$ tail -n 7 sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ tail -n 7 sample1cp.txt
Am
WhoHello      1
World  55
I      19
Adhesh 12
am    77
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Print the from line number 12 to end.

Code:

```
$ tail +12 sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ tail +12 sample1cp.txt
People
I
Am
WhoHello      1
World  55
I      19
Adhesh 12
am    77
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Print the last 12 bytes of a file.

Code:

```
$ tail -c 12 sample1cp.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ tail -c 12 sample1cp.txt
77
Adhesh 12adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

4. Print the tail of more than one file.

Code:

```
$ tail -qv sample1cp.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ tail -qv sample1cp.txt sample2.txt
==> sample1cp.txt <==
Adhesh 12Hello
People
I
Am
WhoHello      1
World    55
I        19
Adhesh   12
am      77
Adhesh   12
==> sample2.txt <==
Hello
People
I
Am
Whoadheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

Note: **-v** prints the filename before each tail.

5. Tail command to monitor the growth (usually for log files)

Code:

```
$ tail -f logfile
```

23. Less

1. Print the content of a file one page at a time.

Code:

```
$ less sample1.txt
```

Output:

```
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12
sample1.txt (END)
```

2. Print the contents of a file along with line numbers.

Code:

```
$ less -N sample1.txt
```

Output:

```
1 Hello 1
2 World 55
3 I 19
4 Adhesh 12
5 am 77
6 Adhesh 12
sample1.txt (END)
```

3. Print the contents of a file on stdout and leave it on the screen even on exit.

Code:

```
$ less -X sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ less -X sample1.txt
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Using less, watch a file for changes.

Code:

```
$ less +F logfile
```

Output:

```
~  
~  
~  
Hello 1  
World 55  
I 19  
Adhesh 12  
am 77  
Adhesh 12  
Waiting for data... (interrupt to abort)
```

24. More

1. Print the content of a file one page at a time.

Code:

```
$ more sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ more sample1.txt  
Hello 1  
World 55  
I 19  
Adhesh 12  
am 77  
Adhesh 12  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Open a file at a particular number.

Code:

```
$ more +3 sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ more +3 sample1.txt  
I 19  
Adhesh 12  
am 77  
Adhesh 12  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Clear the screen and then print the text.

Code:

```
$ more -p sample1.txt
```

Output:

```
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Display the pages on the same area by overlapping the previously displayed text.

Code:

```
$ more -c sample1.txt
```

Output:

```
Hello 1
World 55
I 19
Adhesh 12
am 77
Adhesh 12
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Squeeze multiple blank lines into one single blank line.

Code:

```
$ more -s sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ more sample1.txt
Hello 1
World 55
I 19
Adhesh 12
am 77

Adhesh 12
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ more -s sample1.txt
Hello 1
World 55
I 19
Adhesh 12
am 77

Adhesh 12
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

25. Cal

1. Display the calendar.

Code:

```
$ cal
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cal
August 2020
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
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Display the calendar for the entire year

Code:

```
$ cal 2020
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cal 2020
          2020
January           February          March
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 3 4 5 6 7 8     1 2 3 4 5 6 7
 5 6 7 8 9 10 11    2 3 4 5 6 7 8     8 9 10 11 12 13 14
12 13 14 15 16 17 18   9 10 11 12 13 14 15 15 16 17 18 19 20 21
19 20 21 22 23 24 25   16 17 18 19 20 21 22 22 23 24 25 26 27 28
26 27 28 29 30 31     23 24 25 26 27 28 29 29 30 31

April            May             June
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 3 4 5 6 7 8     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

July              August          September
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 3 4 5 6 7 8     1 2 3 4 5
 5 6 7 8 9 10 11    2 3 4 5 6 7 8     6 7 8 9 10 11 12
12 13 14 15 16 17 18   9 10 11 12 13 14 15 13 14 15 16 17 18 19
19 20 21 22 23 24 25   16 17 18 19 20 21 22 20 21 22 23 24 25 26
26 27 28 29 30 31     23 24 25 26 27 28 29 27 28 29 30
                           30 31

October          November        December
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 3 4 5 6 7     1 2 3 4 5
 4 5 6 7 8 9 10    8 9 10 11 12 13 14 6 7 8 9 10 11 12
11 12 13 14 15 16 17   15 16 17 18 19 20 21 13 14 15 16 17 18 19
18 19 20 21 22 23 24   22 23 24 25 26 27 28 20 21 22 23 24 25 26
25 26 27 28 29 30 31   29 30         27 28 29 30 31
```

3. Display the calendar for a particular month.**Code:**

```
$ cal 4 2017
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cal 4 2017
          April 2017
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
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Display the calendar but turn off the highlighting of date.

Code:

```
$ ncal -h
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ncal -h
      August 2020
Su   2  9 16 23 30
Mo   3 10 17 24 31
Tu   4 11 18 25
We   5 12 19 26
Th   6 13 20 27
Fr   7 14 21 28
Sa  1  8 15 22 29
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

5. Display the calendar of the month along with the week number.

Code:

```
$ ncal -w
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ncal -w
      August 2020
Su   2  9 16 23 30
Mo   3 10 17 24 31
Tu   4 11 18 25
We   5 12 19 26
Th   6 13 20 27
Fr   7 14 21 28
Sa  1  8 15 22 29
 30 31 32 33 34 35
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

5. Display the calendar for 3 months before this month and 2 two month after.

Code:

```
$ cal -B3 -A2
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cal -B3 -A2
      May 2020          June 2020          July 2020
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
           1  2       1  2  3  4   5  6       1  2  3  4
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   5  6  7  8  9 10 11
10 11 12 13 14 15 16 14 15 16 17 18 19 20 12 13 14 15 16 17 18
17 18 19 20 21 22 23 21 22 23 24 25 26 27 19 20 21 22 23 24 25
24 25 26 27 28 29 30 28 29 30               26 27 28 29 30 31
31

     August 2020        September 2020        October 2020
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
           1       1  2  3  4  5       1  2  3
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   4  5  6  7  8  9 10
 9 10 11 12 13 14 15 13 14 15 16 17 18 19 11 12 13 14 15 16 17
16 17 18 19 20 21 22 20 21 22 23 24 25 26 18 19 20 21 22 23 24
23 24 25 26 27 28 29 27 28 29 30               25 26 27 28 29 30 31
30 31
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Change the starting day of the week to Monday.**Code:**

```
$ ncal -M
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ncal -M
      August 2020
Mo      3 10 17 24 31
Tu      4 11 18 25
We      5 12 19 26
Th      6 13 20 27
Fr      7 14 21 28
Sa      1  8 15 22 29
Su      2  9 16 23 30
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

26. Date

1. Display the date.

Code:

```
$ date
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ date
Fri Aug 21 17:58:08 IST 2020
```

2. Display the date in UTC.

Code:

```
$ date -u
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ date -u
Fri Aug 21 12:29:28 UTC 2020
```

3. Format the date.

Code:

```
$ date --date="Feb 2 2010"
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ date --date="Feb 2 2010"
Tue Feb  2 00:00:00 IST 2010
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Display past or future dates

Code:

```
$ date --date="2 months ago"
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ date --date="2 months ago"
Sun Jun 21 18:02:45 IST 2020
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

Code:

```
$ date --date="next monday"
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ date --date="next monday"
Mon Aug 24 00:00:00 IST 2020
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

27. Whatis

1. Get information of a command using whatis

Code:

```
$ whatis whatis
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whatis whatis
whatis (1)           - display one-line manual page descriptions
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Print the debugging information as well.

Code:

```
$ whatis -d whatis
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whatis -d \whatis
From the config file /etc/manpath.config:

Mandatory mandir '/usr/man'.
Mandatory mandir '/usr/share/man'.
Mandatory mandir '/usr/local/share/man'.
Path '/bin' mapped to mandir '/usr/share/man'.
Path '/usr/bin' mapped to mandir '/usr/share/man'.
Path '/sbin' mapped to mandir '/usr/share/man'.
Path '/usr/sbin' mapped to mandir '/usr/share/man'.
Path '/usr/local/bin' mapped to mandir '/usr/local/man'.
Path '/usr/local/bin' mapped to mandir '/usr/local/share/man'.
Path '/usr/local/sbin' mapped to mandir '/usr/local/man'.
Path '/usr/local/sbin' mapped to mandir '/usr/local/share/man'.
Path '/usr/X11R6/bin' mapped to mandir '/usr/X11R6/man'.
Path '/usr/bin/X11' mapped to mandir '/usr/X11R6/man'.
Path '/usr/games' mapped to mandir '/usr/share/man'.
Path '/opt/bin' mapped to mandir '/opt/man'.
Path '/opt/sbin' mapped to mandir '/opt/man'.
Global mandir '/usr/man', catdir '/var/cache/man/fsstnd'.
Global mandir '/usr/share/man', catdir '/var/cache/man'.
Global mandir '/usr/local/man', catdir '/var/cache/man/oldlocal'.
Global mandir '/usr/local/share/man', catdir '/var/cache/man/local'.
Global mandir '/usr/X11R6/man', catdir '/var/cache/man/X11R6'.
Global mandir '/opt/man', catdir '/var/cache/man/opt'.
Added section 'l'.
Added section 'n'.
Added section 'l'.
Added section '8'.
```

3. Get information of a command matching a given string

Code:

```
$ whatis -r man
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whatis -r man
blueman-adapters (1) - an utility to set adapter properties
blueman-applet (1) - a tray applet for managing bluetooth
blueman-assistant (1) - application for configuring and pairing bluetooth devices
blueman-browse (1) - Helper application for launching an obex browser
blueman-manager (1) - bluetooth device manager
blueman-report (1) - a tool to create a commented log and upload it to GitHub for support or bug reports
blueman-sendto (1) - application for sending files to bluetooth devices
blueman-services (1) - Configure local bluetooth services
caca_manage_canvas (3caca) - (unknown subject)
caca_unmanage_canvas (3caca) - (unknown subject)
catman (8) - create or update the pre-formatted manual pages
cmake-commands (7) - CMake Language Command Reference
dh_installman (1) - install man pages into package build directories
dh_installmanpages (1) - old-style man page installer (deprecated)
Exporter::Tiny::Manual::Etc (3pm) - odds and ends
Exporter::Tiny::Manual::Exporting (3pm) - creating an exporter using Exporter::Tiny
Exporter::Tiny::Manual::Importing (3pm) - importing from Exporter::Tiny-based modules
Exporter::Tiny::Manual::QuickStart (3pm) - the quickest way to get up and running with Exporter::Tiny
kernel-command-line (7) - Kernel command line parameters
latex2man (1) - (unknown subject)
light-locker-command (1) - controls light-locker
Lintian::Command (3) - Utilities to execute other commands from lintian code
Lintian::Command::Simple (3) - Run commands without pipes
Lintian::Lab::Manifest (3) - Lintian Lab manifest
Lintian::Lab::ManifestDiff (3) - Difference representation between two Manifests
Lintian::Reporting::ResourceManager (3) - A simple resource manager for html_reports
man (1) - an interface to the on-line reference manuals
man (7) - macros to format man pages
```

4. Display whatis output without trimming.**Code:**

```
$ whatis -l man
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/W
eek2$ whatis -l man
man (7) - macros to format man pages
man (1) - an interface to the on-line reference ma
nuals
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/W
eek2$ whatis man
man (7) - macros to format man pages
man (1) - an interface to the on-line reference...
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/W
eek2$
```

5. Search only in given sections of the man.**Code:**

```
$ whatis -s 1 man
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whatis -s 1 man
man (1)           - an interface to the on-line reference manuals
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whatis -s 3 man
man: nothing appropriate.
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

28. Whereis

1. Get information of where a command is located.

Code:

```
$ whereis whereis
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whereis whereis
whereis: /usr/bin/whereis /usr/share/man/man1/whereis.1.gz
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Search for only binary files

Code:

```
$ whereis -b ls
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whereis -b ls
ls: /bin/ls
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. To locate the man page of a command.

Code:

```
$ whereis -m ls
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whereis -m ls
ls: /usr/share/man/man1/ls.1.gz
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Change or limit the places of search for binaries.

Code:

```
$ whereis -B /bin -f ls
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ whereis -B /bin -f ls
ls: /usr/share/man/man1/ls.1.gz /bin/ls
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

Note:

Similar options are available for sources and manual sections.

-S: Limits the place of search for sources

-M: Limits the place of search for manual sections.

5. Display the files in the current directory that have no documentation file.

Code:

```
$ whereis -m -u *
```

29. Tee

1. Display a file from stdin to stdout.

Code:

```
$ ls | tee
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ls | tee
apache-tomcat-7.0.100
armsim
csaprac...
Desktop
Documents
Downloads
eclipse
eclipse-installer
eclipse-installer (2)
eclipse-workspace
eng.traineddata
floobits
GlassFish_Server
GlassFish_Server1
graphics-installer.sh
idea-IC-193.6494.35
Installed_software
Inverter_Expl.sim
JaspersoftWorkspace
```

2. Write stdin to a file.

Code:

```
$ ls | tee index.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls | tee index.txt
Ex-1.txt
Exercise-I LINUX COMMANDS.docx
Exercise-I LINUX COMMANDS.pdf
index.txt
Linux Commands- Part II.pdf
Linux Commands - Part I.pdf
linuxcp
linuxcp.zip
LinuxhelpManuals
part2
samplelcp.txt
samplel1.txt
sample2.txt
samplezip.zip
screenshots
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head index.txt
Ex-1.txt
Exercise-I LINUX COMMANDS.docx
Exercise-I LINUX COMMANDS.pdf
index.txt
Linux Commands- Part II.pdf
Linux Commands - Part I.pdf
linuxcp
linuxcp.zip
LinuxhelpManuals
part2
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Write stdin to multiple files.**Code:**

```
$ ls | tee f1 f2
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls | tee f1 f2
Ex-1.txt
Exercise-I LINUX COMMANDS.docx
Exercise-I LINUX COMMANDS.pdf
f1
f2
index.txt
Linux Commands- Part II.pdf
Linux Commands - Part I.pdf
linuxcp
linuxcp.zip
LinuxhelpManuals
part2
sample1cp.txt
sample1.txt
sample2.txt
samplezip.zip
screenshots
```

4. Append the stdin to a file.**Code:**

```
$ echo "Hello" | tee -a f2
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ echo "Hello" | tee -a index.txt
Hello
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ tail index.txt
linuxcp
linuxcp.zip
LinuxhelpManuals
part2
sample1cp.txt
sample1.txt
sample2.txt
samplezip.zip
screenshots
Hello
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Add line numbers to a text file**Code:**

```
$ echo "Hello" | tee -a f2
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat -n index.txt | tee index.txt
1 Ex-1.txt
2 Exercise-I LINUX COMMANDS.docx
3 Exercise-I LINUX COMMANDS.pdf
4 f1
5 f2
6 index.txt
7 Linux Commands- Part II.pdf
8 Linux Commands - Part I.pdf
9 linuxcp
10 linuxcp.zip
11 LinuxhelpManuals
12 part2
13 sample1cp.txt
14 sample1.txt
15 sample2.txt
16 samplezip.zip
17 screenshots
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ head index.txt
1 Ex-1.txt
2 Exercise-I LINUX COMMANDS.docx
3 Exercise-I LINUX COMMANDS.pdf
4 f1
5 f2
6 index.txt
7 Linux Commands- Part II.pdf
8 Linux Commands - Part I.pdf
9 linuxcp
10 linuxcp.zip
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

30. Wc

- 1. Print the number of lines in a file.**

Code:

```
$ wc -l index.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat index.txt
1 Ex-1.txt
2 Exercise-I LINUX COMMANDS.docx
3 Exercise-I LINUX COMMANDS.pdf
4 f1
5 f2
6 index.txt
7 Linux Commands- Part II.pdf
8 Linux Commands - Part I.pdf
9 linuxcp
10 linuxcp.zip
11 LinuxhelpManuals
12 part2
13 sample1cp.txt
14 sample1.txt
15 sample2.txt
16 samplezip.zip
17 screenshots
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ wc -l index.txt
17 index.txt
```

2. Print the number of words in a file.

Code:

```
$ wc -w index.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ wc -w index.txt
45 index.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Displays the count of bytes in a file.

Code:

```
$ wc -c index.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ wc -c index.txt
368 index.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Print only the length of the longest line in a file.

Code:

```
$ wc -L index.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ wc -L index.txt
38 index.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

31. Join

The contents in sample1.txt and sample2.txt are:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1.txt
1 Hello
2 World
3 I
4 Adhesh
5 am
6 Adhesh
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample2.txt
1 Hello
2 People
3 I
4 Am
5 Who
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

1. Join two input files.

Code:

```
$ join sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ join sample1.txt sample2.txt
1 Hello Hello
2 World People
3 I I
4 Adhesh Am
5 am Who
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Print unpairable files from file1 as well.

Code:

```
$ join sample1.txt sample2.txt -a 1
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ join sample1.txt sample2.txt -a 1
1 Hello Hello
2 World People
3 I I
4 Adhesh Am
5 am Who
6 Adhesh
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Print only the unpairable line in output.

Code:

```
$ join sample1.txt sample2.txt -v 1
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ join sample1.txt sample2.txt -v 1
6 Adhesh
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Print the join of two files even if the fields are not sorted (Supress the error message)

Code:

```
$ join --nocheck-order sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ join --nocheck-order sample1.txt sample2.txt
5 Hello People
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Print the join of two files even if the fields have a delimiter.

Code:

```
$ join -t, sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ join -t, sample1.txt sample2.txt
1, Hello, Hello
2, World, People
3, I, I
4, Adhesh, Am
5, am, Who
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

32. Comm

The contents in sample1.txt and sample2.txt are:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample1.txt
1      Hello
2      World
3      I
4      Adhesh
5      am
6      Adhesh
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample2.txt
1      Hello
2      People
3      I
4      Am
5      Who
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

1. Compare two sorted files line by line

Code:

```
$ comm sample1.txt sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ comm sample1.txt sample2.txt
      1      Hello
      2      People
2      World
3      I
      3      I
4      Adhesh
      4      Am
5      am
      5      Who
6      Adhesh
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

The above output contains of three columns where first column is separated by zero tab and contains names only present in sample1.txt ,second column contains names only present in sample2.txt and separated by one tab and the third column contains names common to both the files and is separated by two tabs from the beginning of the line.

2. Suppress the first column (lines unique to the first file) in the output of comm.

Code:

```
$ comm -1 sample1.txt sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ comm -1 sample1.txt sample2.txt
1      Hello
2      People
3      I
4      Am
5      Who
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Suppress the second column (lines unique to the second file) in the output of comm.

Code:

```
$ comm -2 sample1.txt sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ comm -2 sample1.txt sample2.txt
1      Hello
2      World
3      I
4      Adhesh
5      am
6      Adhesh
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Suppress the third column (lines common to both) in the output of comm.

Code:

```
$ comm -3 sample1.txt sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ comm -3 sample1.txt sample2.txt
2      People
2      World
3      I
3      I
4      Adhesh
4      Am
5      am
5      Who
6      Adhesh
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Do not check that the input is correctly sorted.

Code:

```
$ comm --nocheck-order sample1.txt sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ comm --nocheck-order sample1.txt sample2.txt
          1      Hello
          2      People
2      World
3      I
3      I
4      Adhesh
4      Am
5      am
5      Who
6      Adhesh
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

33. Tr

Tr is a useful command line utility that translates and/or deletes characters from stdin input, and writes to stdout.

1. Translate the contents of a file into uppercase.

Code:

```
$ cat sample2.txt | tr [:lower:] [:upper:]
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample2.txt | tr [:lower:] [:upper:]
1      HELLO
2      PEOPLE
3      I
4      AM
5      WHO
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Remove tab spaces in a file.

Code:

```
$ cat sample2.txt | tr -d '\t'
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample2.txt | tr -d '\t'
1Hello
2People
3I
4Am
5Who
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Remove repeated sequences of characters in a sequence (for instance double spaces).

Code:

```
$ cat sample2.txt | tr -s ''
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample2.txt
1....Hello
2...People
3...I
4..Am
5....Who
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample2.txt | tr -s '.'
1.Hello
2.People
3.I
4.Am
5.Who
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Delete characters that are in the given set. (Complement)

Code:

```
$ cat sample2.txt | tr -c '[[:digit:]]' '.'
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cat sample2.txt | tr -c '[[:digit:]]' '.'
1.....2.....3.....4.....5.....adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week
2$
```

34. Cut

Cut remove sections from each line of files

1. Cut specific bytes of characters from each line in a file.

Code:

```
$ cut -b 1,2,3 sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cut -b 4,5,6 sample2.txt
..H
.Pe
.I
Am
..W
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Cut specific range of bytes of characters from each line in a file.

Code:

```
$ cut -b 1-3,6- sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cut -b 1-3,6- sample2.txt
1..Hello
2..eople
3..
4..
5..Who
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Cut specific characters from each line in a file.

Code:

```
$ cut -c 1-3,6- sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cut -c 1-3,6- sample2.txt
1..Hello
2..eople
3..
4..
5..Who
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Cut specific fields from each line in a file.

Code:

```
$ cut -f 1 sample1.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cut -f 1 sample1.txt
1
2
3
4
5
6
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Cut the complement of the given field

Code:

```
$ cut --complement -f 1 sample1.txt
```

Output:

```
adhesheghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ cut --complement -f 1 sample1.txt
Hello
World
I
Adhesh
am
Adhesh
adhesheghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

35. Paste

Paste merge lines of files.

1. Merge two files parallelly.

Code:

```
$ paste sample1.txt sample2.txt
```

Output:

```
adhesheghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ paste sample1.txt sample2.txt
Hello  Hello
World  People
I      I
Adhesh Who
am     Apple
Adhesh
adhesheghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Merge two files parallelly with a delimiter in between.

Code:

```
$ paste -d "!" sample1.txt sample2.txt
```

Output:

```
adhesheghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ paste -d "!" sample1.txt sample2.txt
Hello!Hello
World!People
I!I
Adhesh!Who
am!Apple
Adhesh!
adhesheghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Merge two files in a sequential manner.

Code:

```
$ paste -s sample1.txt sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ paste -s sample1.txt sample2.txt
Hello  World  I      Adhesh am      Adhesh
Hello  People I      Who     Apple
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Combine 3 consecutive lines in a file.

Code:

```
$ paste - - - < sample1.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ paste - - - < sample1.txt
Hello  World  I
Adhesh am      Adhesh
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

36. NI

NI prints number of lines in a file

1. Print a file along with line numbers.

Code:

```
$ nl sample2.txt
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ nl sample2.txt
 1 Hello
 2 People
 3 I
 4 Who
 5 Apple
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Print line numbers for line spaces as well.

Code:

```
$ nl -b a sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ nl -b a sample2.txt
1 Hello
2
3 People
4 I
5 Who
6
7 Apple
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

3. Customize the increment value while printing the file with line numbers.

Code:

```
$ nl -i 2 sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ nl -i 2 sample2.txt
1 Hello

3 People
5 I
7 Who

9 Apple
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ █
```

4. Print line numbers for 2 consecutive line spaces as well.

Code:

```
$ nl -b a -l 2 sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ nl -b a -l 2 sample2.txt
1 Hello
2
3 People
4
5
6 I
7
8 Who
9
10 Apple
```

5. Print line numbers but make the starting number different.**Code:**

```
$ nl -v 2 -i 2 sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ nl -v 2 -i 2 sample1.txt
2 Hello
4 World
6 I
8 Adhesh
10 am
12 Adhesh
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

-v sets the starting number.

6. Print line numbers with custom separators.**Code:**

```
$ nl -s ":" sample.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ nl -s ":" sample1.txt
1::Hello
2::World
3::I
4::Adhesh
5::am
6::Adhesh
```

37. Su

Su - change user ID or become superuser

1. Switch to a different user

Code:

```
$ su -l adheshreghu
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ su -l adheshreghu
Password:
. /opt/Xilinx/14.7/ISE_DS/common/.settings64.sh /opt/Xilinx/14.7/ISE_DS/common
. /opt/Xilinx/14.7/ISE_DS/EDK/.settings64.sh /opt/Xilinx/14.7/ISE_DS/EDK
. /opt/Xilinx/14.7/ISE_DS/PlanAhead/.settings64.sh /opt/Xilinx/14.7/ISE_DS/PlanA
head
. /opt/Xilinx/14.7/ISE_DS/ISE/.settings64.sh /opt/Xilinx/14.7/ISE_DS/ISE
adheshreghu@adheshreghu-Inspiron-5570:~$
```

2. Run a command as a different user

Code:

```
$ su -c ls adheshreghu
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ su -c ls adheshreghu
Password:
apache-tomcat-7.0.100  'eclipse-installer (2)'  Installed software  NewFolder  Servers
armsim                  eclipse-workspace        InverterExpl.sim  'New Folder'  snap
csapract...              eng.traineddata       JaspersoftWorkspace Pictures  target
Desktop                 floobits                libstdc++.so.6      prefix32   Templates
Documents               GlassFish_Server       mozilla.pdf       project   Tutorial
Downloads               GlassFish_Server1     Music             Public    Videos
eclipse                 graphics-installer.sh myapplication  Release.key
eclipse-installer        idea-IC-193.6494.35 NetBeansProjects sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~$
```

3. Use a different shell

Code:

```
$ su -s /usr/bin/zsh
```

This command opens a root user account in **Z shell**.

4. Use a Different User in the Same Environment

Code:

```
$ su -p adheshreghu
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ su -p adheshreghu
Password:
. /opt/Xilinx/14.7/ISE_DS/common/.settings64.sh /opt/Xilinx/14.7/ISE_DS/common
. /opt/Xilinx/14.7/ISE_DS/EDK/.settings64.sh /opt/Xilinx/14.7/ISE_DS/EDK
. /opt/Xilinx/14.7/ISE_DS/PlanAhead/.settings64.sh /opt/Xilinx/14.7/ISE_DS/PlanAhead
. /opt/Xilinx/14.7/ISE_DS/ISE/.settings64.sh /opt/Xilinx/14.7/ISE_DS/ISE
adheshreghu@adheshreghu-Inspiron-5570:~$ echo $HOME
/home/adheshreghu
```

38. Chmod

chmod - change file mode bits

1. Change the user permission to only read for a given file

Code:

```
$ chmod u=r sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls sample2.txt
sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-rw-rw-r-- 1 adheshreghu adheshreghu 37 Aug 21 21:21 sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ chmod u=r sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-r--r--r-- 1 adheshreghu adheshreghu 37 Aug 21 21:21 sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Give (Add) execute permission for all users

Code:

```
$ chmod a+x sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-rw-r--r-- 1 adheshreghu adheshreghu 37 Aug 21 21:21 sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ chmod a+x sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-rwx-wxwx 1 adheshreghu adheshreghu 37 Aug 21 21:21 sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Recursively remove the write permission for users for a directory

Code:

```
$ chmod -R u-w sample2.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ ls -l Week2/
total 15652
-rw-r--r-- 1 adheshreghu adheshreghu 6005 Aug 19 00:29 Ex-1.txt
-rw-r--r-- 1 adheshreghu adheshreghu 17285 Aug 13 17:04 'Exercise-I LINUX COMMANDS.docx'
-rw-r--r-- 1 adheshreghu adheshreghu 120472 Aug 13 17:04 'Exercise-I LINUX COMMANDS.pdf'
-rw-r--r-- 1 adheshreghu adheshreghu 249 Aug 21 19:30 f1
-rw-r--r-- 1 adheshreghu adheshreghu 249 Aug 21 19:30 f2
-rw-r--r-- 1 adheshreghu adheshreghu 368 Aug 21 19:33 index.txt
-rw-r--r-- 1 adheshreghu adheshreghu 2356355 Aug 19 11:31 'Linux Commands- Part II.pdf'
-rw-r--r-- 1 adheshreghu adheshreghu 1781329 Aug 19 11:31 'Linux Commands - Part I.pdf'
drwxrwxr-x 2 adheshreghu adheshreghu 4096 Aug 21 14:19 linuxcp
-rw-r--r-- 1 adheshreghu adheshreghu 11685946 Aug 17 23:58 linuxcp.zip
drwxrwxr-x 2 adheshreghu adheshreghu 4096 Aug 13 17:05 LinuxhelpManuals
drwxrwxr-x 3 adheshreghu adheshreghu 4096 Aug 21 12:37 part2
-rw-r--r-- 1 adheshreghu adheshreghu 163 Aug 21 16:45 sample1cp.txt
-rw-r--r-- 1 adheshreghu adheshreghu 30 Aug 21 21:09 sample1.txt
-rw-r--r-- 1 adheshreghu adheshreghu 37 Aug 21 21:21 sample2.txt
-rw-r--r-- 1 adheshreghu adheshreghu 535 Aug 18 00:01 samplezip.zip
drwxrwxr-x 42 adheshreghu adheshreghu 4096 Aug 18 00:03 screenshots
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ chmod -R u-w Week2/part2/
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ ls -l Week2/
total 15652
-rw-r--r-- 1 adheshreghu adheshreghu 6005 Aug 19 00:29 Ex-1.txt
-rw-r--r-- 1 adheshreghu adheshreghu 17285 Aug 13 17:04 'Exercise-I LINUX COMMANDS.docx'
-rw-r--r-- 1 adheshreghu adheshreghu 120472 Aug 13 17:04 'Exercise-I LINUX COMMANDS.pdf'
-rw-r--r-- 1 adheshreghu adheshreghu 249 Aug 21 19:30 f1
-rw-r--r-- 1 adheshreghu adheshreghu 249 Aug 21 19:30 f2
-rw-r--r-- 1 adheshreghu adheshreghu 368 Aug 21 19:33 index.txt
-rw-r--r-- 1 adheshreghu adheshreghu 2356355 Aug 19 11:31 'Linux Commands- Part II.pdf'
-rw-r--r-- 1 adheshreghu adheshreghu 1781329 Aug 19 11:31 'Linux Commands - Part I.pdf'
drwxrwxr-x 2 adheshreghu adheshreghu 4096 Aug 21 14:19 linuxcp
-rw-r--r-- 1 adheshreghu adheshreghu 11685946 Aug 17 23:58 linuxcp.zip
drwxrwxr-x 2 adheshreghu adheshreghu 4096 Aug 13 17:05 LinuxhelpManuals
drwxrwxr-x 3 adheshreghu adheshreghu 4096 Aug 21 12:37 part2
-rw-r--r-- 1 adheshreghu adheshreghu 163 Aug 21 16:45 sample1cp.txt
-rw-r--r-- 1 adheshreghu adheshreghu 30 Aug 21 21:09 sample1.txt
-rw-r--r-- 1 adheshreghu adheshreghu 37 Aug 21 21:21 sample2.txt
-rw-r--r-- 1 adheshreghu adheshreghu 535 Aug 18 00:01 samplezip.zip
drwxrwxr-x 42 adheshreghu adheshreghu 4096 Aug 18 00:03 screenshots
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$
```

4. Remove the read, write, and execute permission for all users except the file's owner.

Code:

```
$ chmod og-rwx sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample1.txt
-rw-rw-r-- 1 adheshreghu adheshreghu 30 Aug 21 21:09 sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ chmod og-rwx sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample1.txt
-rw----- 1 adheshreghu adheshreghu 30 Aug 21 21:09 sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

5. Change permission in numeric method

Code:

```
$ chmod 777 sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample1.txt
-rw----- 1 adheshreghu adheshreghu 30 Aug 21 21:09 sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ chmod 777 sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample1.txt
-rwxrwxrwx 1 adheshreghu adheshreghu 30 Aug 21 21:09 sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

39. Chown

chown - change file owner and group

1. Change the ownership of a given file

Code:

```
$ sudo chown root sample1.txt
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample1.txt
-rwxrwxrwx 1 adheshreghu adheshreghu 30 Aug 21 21:09 sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sudo chown root sample1.txt
[sudo] password for adheshreghu:
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample1.txt
-rwxrwxrwx 1 root adheshreghu 30 Aug 21 21:09 sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Change the owner and group of a given file

Code:

```
$ sudo chown root:users sample2.txt
```

Output:

```
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-rwxr--r-- 1 adhesreghu adhesreghu 37 Aug 21 21:21 sample2.txt
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sudo chown root:users sample2.txt
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-rwxr--r-- 1 root users 37 Aug 21 21:21 sample2.txt
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Change the group of a file.

Code:

```
$ sudo chown :adheshreghu sample2.txt
```

Output:

```
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-rwxr--r-- 1 root users 37 Aug 21 21:21 sample2.txt
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sudo chown :adheshreghu sample2.txt
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l sample2.txt
-rwxr--r-- 1 root adhesreghu 37 Aug 21 21:21 sample2.txt
adhesreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Recursively change the ownership of a directory.

Code:

```
$ sudo chown -R root:root part2/
```

Output:

```
adhesheghu@adhesheghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l part2/
total 28
-r--rw-r-- 1 adhesheghu adhesheghu 0 Aug 21 12:37 file
-r--rw-r-- 1 adhesheghu adhesheghu 0 Aug 21 12:37 file1
-r--rw-r-- 1 adhesheghu adhesheghu 1804 Aug 18 21:29 hello.txt
-r--rw-r-- 1 adhesheghu adhesheghu 1473 Aug 17 12:45 Input
-r--rw-r-- 1 adhesheghu adhesheghu 135 Aug 18 22:27 records.txt
-r--rw-r-- 1 adhesheghu adhesheghu 906 Aug 17 23:51 sample1.txt
-r--rw-r-- 1 adhesheghu adhesheghu 630 Aug 19 21:30 sample2.txt
-r--rw-r-- 1 adhesheghu adhesheghu 2241 Aug 18 21:04 sample.txt
dr-xrwxr-x 2 adhesheghu adhesheghu 4096 Aug 18 22:53 screenshots
adhesheghu@adhesheghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ sudo chown -R root:root part2/
adhesheghu@adhesheghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ ls -l part2/
total 28
-r--rw-r-- 1 root root 0 Aug 21 12:37 file
-r--rw-r-- 1 root root 0 Aug 21 12:37 file1
-r--rw-r-- 1 root root 1804 Aug 18 21:29 hello.txt
-r--rw-r-- 1 root root 1473 Aug 17 12:45 Input
-r--rw-r-- 1 root root 135 Aug 18 22:27 records.txt
-r--rw-r-- 1 root root 906 Aug 17 23:51 sample1.txt
-r--rw-r-- 1 root root 630 Aug 19 21:30 sample2.txt
-r--rw-r-- 1 root root 2241 Aug 18 21:04 sample.txt
dr-xrwxr-x 2 root root 4096 Aug 18 22:53 screenshots
adhesheghu@adhesheghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

40. Ifconfig

ifconfig - configure network interfaces

1. View all network settings.

Code:

```
$ ifconfig
```

Output:

```
adhesheghu@adhesheghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ ifconfig
enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether d0:94:66:f9:eb:fd txqueuelen 1000  (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        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 5196 bytes 517542 (517.5 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 5196 bytes 517542 (517.5 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.8 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 fe80::3e43:ec8c:20d7:6e49 prefixlen 64 scopeid 0x20<link>
          ether 5c:ea:1d:60:7c:ff txqueuelen 1000  (Ethernet)
            RX packets 5214008 bytes 3780729521 (3.7 GB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2089791 bytes 488078960 (488.0 MB)
```

2. Display Information of All Network Interfaces

Code:

```
$ ifconfig -a
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ifconfig -a
enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether d0:94:66:f9:eb:fd txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      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 4092 bytes 412841 (412.8 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 4092 bytes 412841 (412.8 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.8 netmask 255.255.255.0 broadcast 192.168.0.255
      inet6 fe80::3e43:ec8c:20d7:6e49 prefixlen 64 scopeid 0x20<link>
        ether 5c:ea:1d:60:7c:ff txqueuelen 1000 (Ethernet)
        RX packets 4949202 bytes 3685246184 (3.6 GB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1854401 bytes 395699550 (395.6 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. View Network Settings of Specific Interface

Code:

```
$ ifconfig lo
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ifconfig lo
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 4096 bytes 413261 (413.2 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 4096 bytes 413261 (413.2 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

4. Enable a network interface.

Code:

```
$ ifconfig eth0 up
```

5. Disable a network interface.

Code:

```
$ ifconfig eth0 down
```

6. Assign IP address to network interface.

Code:

```
$ ifconfig eth0 172.16.25.125
```

41. Ping

ping -send ICMP ECHO_REQUEST to network hosts

1. Run a simple ping to see if the target host is reachable.

Code:

```
$ ping www.google.com
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ ping www.google.com
PING www.google.com (172.217.163.36) 56(84) bytes of data.
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=1 ttl=119 time=20.9 ms

64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=2 ttl=119 time=8.40 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=3 ttl=119 time=7.19 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=4 ttl=119 time=6.87 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=5 ttl=119 time=15.2 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=6 ttl=119 time=8.07 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=7 ttl=119 time=7.54 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=8 ttl=119 time=8.09 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=9 ttl=119 time=11.0 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=10 ttl=119 time=15.4 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=11 ttl=119 time=38.9 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=12 ttl=119 time=8.81 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=13 ttl=119 time=21.5 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=14 ttl=119 time=11.5 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=15 ttl=119 time=14.7 ms
64 bytes from maa05s01-in-f4.le100.net (172.217.163.36): icmp_seq=16 ttl=119 time=6.90 ms
^C
--- www.google.com ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15024ms
rtt min/avg/max/mdev = 6.876/13.207/38.922/8.110 ms
adhesreghu@adhesreghu-Inspiron-5570:~$
```

2. Set the interval between sending each packet.

Code:

```
$ ping -i 3 -c 5 www.google.com
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ ping -i 3 -c 5 www.google.com
PING www.google.com (172.217.163.36) 56(84) bytes of data.
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=1 ttl=119 time=8.52 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=2 ttl=119 time=8.39 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=3 ttl=119 time=7.14 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=4 ttl=119 time=12.9 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=5 ttl=119 time=9.76 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 12013ms
rtt min/avg/max/mdev = 7.149/9.365/12.987/1.991 ms
```

3. Send only 6 packets to the target.

Code:

```
$ ping -c 6 www.google.com
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ ping -c 6 www.google.com
PING www.google.com (172.217.163.36) 56(84) bytes of data.
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=1 ttl=119 time=9.26 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=2 ttl=119 time=6.13 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=3 ttl=119 time=11.7 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=4 ttl=119 time=13.0 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=5 ttl=119 time=8.21 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=6 ttl=119 time=10.2 ms

--- www.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 6.132/9.774/13.069/2.271 ms
adhesreghu@adhesreghu-Inspiron-5570:~$
```

4. To determine the response of your network under high-load conditions, you can run a “flood ping” which sends requests as fast as possible.

Code:

```
$ ping -f www.google.com
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ ping -f www.google.com
PING www.google.com (172.217.163.36) 56(84) bytes of data.
ping: cannot flood; minimal interval allowed for user is 200ms
adhesreghu@adhesreghu-Inspiron-5570:~$
```

5. To limit the number of network hops (TTL – Time-to-live) that probes traverse, use the **-t flag.**

Code:

```
$ ping -t 10 www.google.com
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ ping -t 10 www.google.com
PING www.google.com (172.217.163.36) 56(84) bytes of data.
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=1 ttl=119 time=7.40 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=2 ttl=119 time=9.45 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=3 ttl=119 time=8.93 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=4 ttl=119 time=9.78 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=5 ttl=119 time=8.73 ms
64 bytes from maa05s01-in-f4.1e100.net (172.217.163.36): icmp_seq=6 ttl=119 time=11.1 ms
^C
--- www.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 7.407/9.236/11.104/1.125 ms
adheshreghu@adheshreghu-Inspiron-5570:~$
```

42. Traceroute

traceroute -print the route packets trace to network host

1. Print the route a packet takes to reach the host.

Code:

```
$ traceroute www.google.com
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ traceroute www.google.com
traceroute to www.google.com (172.217.163.36), 30 hops max, 60 byte packets
1 _gateway (192.168.0.1) 4.990 ms 6.667 ms 6.657 ms
2 100.74.0.1 (100.74.0.1) 14.767 ms 15.441 ms 17.992 ms
3 Chennai-Core.youbroadband.in (203.187.244.1) 25.189 ms 26.787 ms 33.265 ms
4 mylapore-cmts01.youbroadband.in (203.187.244.118) 29.607 ms 31.047 ms 35.432 ms
5 33-244-187-203.static.youbroadband.in (203.187.244.33) 41.939 ms 42.983 ms 42.998 ms
6 * * *
7 * * *
8 maa05s01-in-f4.1e100.net (172.217.163.36) 19.412 ms 8.643 ms 11.030 ms
adheshreghu@adheshreghu-Inspiron-5570:~$
```

2. Use IP4 while tracing routes.

Code:

```
$ traceroute -4 www.google.com
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ traceroute -4 www.google.com
traceroute to www.google.com (172.217.163.36), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1)  3.312 ms  4.257 ms  5.232 ms
 2 100.74.0.1 (100.74.0.1)  12.455 ms  14.577 ms  18.107 ms
 3 Chennai-Core.youbroadband.in (203.187.244.1)  36.487 ms  41.852 ms  44.292 ms
 4 mylapore-cmts01.youbroadband.in (203.187.244.118)  24.224 ms  26.391 ms  28.888 ms
 5 33-244-187-203.static.youbroadband.in (203.187.244.33)  31.210 ms  33.918 ms  38.689 ms
 6 * * *
 7 * * *
 8 maa05s01-in-f4.1e100.net (172.217.163.36)  42.624 ms  38.626 ms  38.569 ms
adheshreghu@adheshreghu-Inspiron-5570:~$
```

3. Use IP6 while tracing routes.

Code:

```
$ traceroute -6 www.google.com
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ traceroute -6 www.google.com
traceroute to www.google.com (2404:6800:4007:80b::2004), 30 hops max, 80 byte packets
 1 _gateway (fe80::12be:f5ff:fe7b:7a81%wlp3s0)  2.936 ms  3.760 ms  3.803 ms
 2 _gateway (fe80::12be:f5ff:fe7b:7a81%wlp3s0)  4.260 ms !N  5.073 ms !N  5.129 ms !N
adheshreghu@adheshreghu-Inspiron-5570:~$
```

4. Traceroute without fragmenting data packet.

Code:

```
$ traceroute -F www.google.com
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ traceroute -F www.google.com
traceroute to www.google.com (172.217.163.36), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1)  6.089 ms  6.800 ms  7.784 ms
 2 100.74.0.1 (100.74.0.1)  11.893 ms  13.917 ms  16.150 ms
 3 Chennai-Core.youbroadband.in (203.187.244.1)  21.435 ms  28.054 ms  30.319 ms
 4 mylapore-cmts01.youbroadband.in (203.187.244.118)  48.677 ms  53.619 ms  55.631 ms
 5 33-244-187-203.static.youbroadband.in (203.187.244.33)  34.121 ms  37.273 ms  38.945 ms
 6 * * *
 7 * * *
 8 maa05s01-in-f4.1e100.net (172.217.163.36)  46.490 ms  43.077 ms  38.907 ms
adheshreghu@adheshreghu-Inspiron-5570:~$
```

5. Route the packet through a gate.

Code:

```
$ traceroute -g 192.168.43.45 google.com
```

Output:

```
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 * * *

adhesreghu@adhesreghu-Inspiron-5570:~$
```

6. Traceroute, but do not resolve ip addresses to domain names.

Code:

```
$ traceroute -n google.com
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ traceroute -n google.com
traceroute to google.com (172.217.167.142), 30 hops max, 60 byte packets
1 192.168.0.1 3.172 ms 4.049 ms 4.975 ms
2 100.74.0.1 10.596 ms 12.647 ms 14.610 ms
3 203.187.244.1 18.657 ms 27.186 ms 31.681 ms
4 203.187.244.118 22.280 ms 24.587 ms 29.824 ms
5 203.187.244.33 34.135 ms 38.249 ms 38.943 ms
6 * * *
7 * * *
8 172.217.167.142 39.308 ms 37.293 ms 36.439 ms

adhesreghu@adhesreghu-Inspiron-5570:~$
```

43. Uname

uname - print system information

1. Print all system information

Code:

```
$ uname -a
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ uname -a
Linux adheshreghu-Inspiron-5570 5.4.0-42-generic #46~18.04.1-Ubuntu SMP Fri Jul 10 07:21:24 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
adheshreghu@adheshreghu-Inspiron-5570:~$ █
```

The order of printing is Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system.

2. Print only kernel name

Code:

```
$ uname -s
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ uname -s
Linux
adheshreghu@adheshreghu-Inspiron-5570:~$ █
```

3. Print hostname of the network node

Code:

```
$ uname -n
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ uname -n
adheshreghu-Inspiron-5570
adheshreghu@adheshreghu-Inspiron-5570:~$ █
```

4. Print the kernel release date.

Code:

```
$ uname -r
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ uname -r
5.4.0-42-generic
adheshreghu@adheshreghu-Inspiron-5570:~$
```

5. Print the current version of kernel.

Code:

```
$ uname -v
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ uname -v
#46~18.04.1-Ubuntu SMP Fri Jul 10 07:21:24 UTC 2020
adheshreghu@adheshreghu-Inspiron-5570:~$
```

6. Print the machine hardware name.

Code:

```
$ uname -m
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ uname -m
x86_64
adheshreghu@adheshreghu-Inspiron-5570:~$
```

7. Print the name of the operating system.

Code:

```
$ uname -o
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ uname -o
GNU/Linux
adheshreghu@adheshreghu-Inspiron-5570:~$
```

44. Passwd

passwd - change user password

1. Change password for system users

Code:

```
$ passwd
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ passwd
Changing password for adheshreghu.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
Password unchanged
Enter new UNIX password: [REDACTED]
```

2. Display password status information.

Code:

```
$ passwd -S adheshreghu
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ passwd -S adheshreghu
adheshreghu P 02/25/2020 0 99999 7 -1
adheshreghu@adheshreghu-Inspiron-5570:~$ [REDACTED]
```

3. Display password status for all accounts

Code:

```
$ passwd -Sa
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ sudo passwd -Sa
[sudo] password for adheshreghu:
root L 02/25/2020 0 99999 7 -1
daemon L 02/03/2020 0 99999 7 -1
bin L 02/03/2020 0 99999 7 -1
sys L 02/03/2020 0 99999 7 -1
sync L 02/03/2020 0 99999 7 -1
games L 02/03/2020 0 99999 7 -1
man L 02/03/2020 0 99999 7 -1
lp L 02/03/2020 0 99999 7 -1
mail L 02/03/2020 0 99999 7 -1
```

45. Mount

mount - mount a file system

1. Display mounted file systems.

Code:

```
$ mount
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=3999504k,nr_inodes=999876,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=804692k,mode=755)
/dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
```

2. Display only ext4 partitions.

Code:

```
$ mount -t ext4
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~$ mount -t ext4
/dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)
adhesreghu@adhesreghu-Inspiron-5570:~$
```

3. Mount a file system

Code:

```
$ sudo mount /dev/sdb1 /mnt/media
```

4. Unmount a file system

Code:

```
$ umount DIRECTORY  
$ umount DEVICE_NAME
```

46. Free

free - Display amount of free and used memory in the system

1. Display information about the memory and the swap in KB.

Code:

```
$ free
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ free  
      total        used        free      shared  buff/cache   available  
Mem:    8046912     2684716     1676872      370564     3685324     4681652  
Swap:  7812092        524     7811568  
adheshreghu@adheshreghu-Inspiron-5570:~$
```

2. Display Memory Usage in Human Readable Format.

Code:

```
$ free -h
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ free -h  
      total        used        free      shared  buff/cache   available  
Mem:    7.7G       2.6G       1.5G      345M       3.5G       4.4G  
Swap:  7.5G       524K      7.4G  
adheshreghu@adheshreghu-Inspiron-5570:~$
```

3. Display Memory Usage in megabytes

Code:

```
$ free --mega
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ free --mega
      total        used         free        shared   buff/cache   available
Mem:    8046       2768       1603        360       3674       4607
Swap:  7812          0       7811
adheshreghu@adheshreghu-Inspiron-5570:~$
```

4. To display a line showing the column totals.

Code:

```
$ free -h -t
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ free -h -t
      total        used         free        shared   buff/cache   available
Mem:    7.7G       2.7G       1.4G        416M       3.6G       4.3G
Swap:  7.5G      524K       7.4G
Total: 15G       2.7G       8.9G
adheshreghu@adheshreghu-Inspiron-5570:~$
```

47. Fold

fold - wrap each input line to fit in specified width

1. Limit the display of a file by number of columns.

Code:

```
$ fold -w30 f1
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ fold -w30 f1
The terms "linefeed" and \n
\r\ncarriage return" date back
to the typewriter. The platen
, the cylinder that the paper
was wrapped around, was mounted
on a moveable carriage. The
carriage moved one character\n
\rwidth to the left each time
you hit a key. To start a new
line, you pushed a lever that
brought the carriage back to
its original position, and which
rotated the roller and moved
the paper upwards by the height
of one line. This action was
known as the carriage return,
and the rotation of the cylinder
(and the advancement of
the paper) was known as a line
feed.
```

2. Limit the display of a file by number of bytes.

Code:

```
$ fold -b50 f1
```

Output:

```
@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ fold -b50 f1
The terms "linefeed" and "carriage return"
date back to the typewriter. The platen, the cylinder that the paper was wrapped around, was mounted on a moveable carriage. The carriage moved one character's width to the left each time you hit a key. To start a new line, you pushed a lever that brought the carriage back to its original position, and which rotated the roller and moved the paper upwards by the height of one line. This action was known as the carriage return, and the rotation of the cylinder (and the advancement of the paper) was known as a linefeed.

The lever was replaced by a key when the typewriter became electrified. The key was labeled Carriage Return or just Return. Some early computers such as the BBC Micro still used the name Return on what we now call the Enter key.

You can't see newline characters, as a rule. You can only see their effect. The newline character forces software that displays or processes text to start a new line. @adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

3. Break the lines on spaces so that words are not broken.

Code:

```
$ fold -w50 -s f1
```

Output:

```
@adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ fold -w50 -s f1
The terms "linefeed" and "carriage return" date back to the typewriter. The platen, the cylinder that the paper was wrapped around, was mounted on a moveable carriage. The carriage moved one character's width to the left each time you hit a key. To start a new line, you pushed a lever that brought the carriage back to its original position, and which rotated the roller and moved the paper upwards by the height of one line. This action was known as the carriage return, and the rotation of the cylinder (and the advancement of the paper) was known as a linefeed.
```

48. Wget

wget - The non-interactive network downloader.

1. Download a single file using wget

Code:

```
$ wget http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ wget http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
--2020-08-21 23:03:34-- http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
Resolving ftp.gnu.org (ftp.gnu.org)... 209.51.188.20, 2001:470:142:3::b
Connecting to ftp.gnu.org (ftp.gnu.org)|209.51.188.20|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 446966 (436K) [application/x-gzip]
Saving to: 'wget-1.5.3.tar.gz'

wget-1.5.3.tar.gz      100%[=====] 436.49K   329KB/s    in 1.3s

2020-08-21 23:03:38 (329 KB/s) - 'wget-1.5.3.tar.gz' saved [446966/446966]

adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Download the file with a different name.

Code:

```
$ wget -O wget.zip http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~$ wget -O wget.zip http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
--2020-08-21 23:04:51-- http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
Resolving ftp.gnu.org (ftp.gnu.org)... 209.51.188.20, 2001:470:142:3::b
Connecting to ftp.gnu.org (ftp.gnu.org)|209.51.188.20|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 446966 (436K) [application/x-gzip]
Saving to: 'wget.zip'

wget.zip      100%[=====] 436.49K   2.83MB/s    in 0.2s

2020-08-21 23:04:52 (2.83 MB/s) - 'wget.zip' saved [446966/446966]

adheshreghu@adheshreghu-Inspiron-5570:~$
```

3. Resume uncompleted download.

Code:

```
$ wget -c  
http://mirrors.hns.net.in/centos/6.3/isos/x86_64/CentOS-6.3-x86_64-LiveDVD.iso
```

4. Download files in the background.

Code:

```
$ wget -b /wget/log.txt  
ftp://ftp.iinet.net.au/debian/debian-cd/6.0.5/i386/iso-dvd/debian-6.0.5-i386-DVD  
-1.iso
```

5. Download reading URL from a file

Code:

```
$ wget -i filename
```

49. xargs

xargs - build and execute command lines from standard input

1. Xargs with find and rm - to delete files that match the find pattern

Code:

```
$ find *.zip | xargs rm
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find *.zip  
linuxcp.zip  
samplezip.zip  
wgeto.zip  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find *.zip | xargs rm  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find *.zip  
find: '*.zip': No such file or directory  
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

2. Archive all text files

Code:

```
$ find / -name *.txt -type f -print | xargs tar -cvzf text.tar.gz
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ find Week2 -name *.txt -type f -print | xargs tar -cvzf text.tar.gz
Week2/Ex-1.txt
Week2/part2/sample.txt
Week2/part2/sample2.txt
Week2/part2/records.txt
Week2/part2/hello.txt
Week2/part2/sample1.txt
Week2/sample2.txt
Week2/sample1cp.txt
Week2/sample1.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ ls
sphinx-Doc  text.tar.gz  Week1  Week2
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$
```

3. Use xargs with find and grep. Search a pattern in the find result.

Code:

```
$ find *.txt | grep "sample"
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ find *.txt | grep "sample"
sample1cp.txt
sample1.txt
sample2.txt
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

4. Print the commands that are executed using -t.

Code:

```
$ echo 'one two three' | xargs -t rm
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ echo 'one two three' | xargs -t rm
rm one two three
```

5. Limit Output Per Line Using -n Option

Code:

```
$ echo 'a b c d e f' | xargs -n 3
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$ echo 'a b c d e f' | xargs -n 3
a b c
d e f
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab/Week2$
```

50. Man

man - an interface to the on-line reference manuals

1. Print the whole manual of a given command

Code:

```
$ man man
```

Output:

```
MAN(1)                                Manual pager utils                               MAN(1)

NAME
    man - an interface to the on-line reference manuals

SYNOPSIS
    man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-m sys-
    tem,...]] [-M path] [-S list] [-e extension] [-i|-I] [--regex|--wildcard] [--names-only]
    [-a] [-u] [--no-subpages] [-P pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation]
    [--no-justification] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z] [[sec-
    tion] page[.section] ...] ...
    man -k [apropos options] regexp ...
    man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
    man -f [whatis options] page ...
    man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-P pager]
    [-r prompt] [-7] [-E encoding] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
    file ...
    man -w|-W [-C file] [-d] [-D] page ...
    man -c [-C file] [-d] [-D] page ...
    man [-?V]

DESCRIPTION
    man is the system's manual pager. Each page argument given to man is normally the name
    of a program, utility or function. The manual page associated with each of these argu-
    ments is then found and displayed. A section, if provided, will direct man to look only
    in that section of the manual. The default action is to search in all of the available
    sections following a pre-defined order ("1 n l 8 3 2 3posix 3pm 3perl 3am 5 4 9 6 7" by
    default, unless overridden by the SECTION directive in /etc/manpath.config), and to show
    Manual page man(1) line 1 (press h for help or q to quit)
```

2. Display specific sections of the manual.

Code:

```
$ man 7 signal
```

Output:

```
SIGNAL(7)           Linux Programmer's Manual           SIGNAL(7)

NAME
    signal - overview of signals

DESCRIPTION
    Linux supports both POSIX reliable signals (hereinafter "standard signals") and POSIX
    real-time signals.

Signal dispositions
    Each signal has a current disposition, which determines how the process behaves when it
    is delivered the signal.

    The entries in the "Action" column of the tables below specify the default disposition
    for each signal, as follows:

        Term   Default action is to terminate the process.
        Ign    Default action is to ignore the signal.
        Core   Default action is to terminate the process and dump core (see core(5)).
        Stop   Default action is to stop the process.
        Cont   Default action is to continue the process if it is currently stopped.

    A process can change the disposition of a signal using sigaction(2) or signal(2). (The
    latter is less portable when establishing a signal handler; see signal(2) for details.)
Manual page signal(7) line 1 (press h for help or q to quit)
```

3. Display the section in which the given command is present.

Code:

```
$ man -f signal
```

Output:

```
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ man -f signal
signal (7)          - overview of signals
signal (2)          - ANSI C signal handling
adheshreghu@adheshreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$
```

4. Display all the available manual pages of the command in succession.

Code:

```
$ man -a signal
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ man -a signal
--Man-- next: signal(7) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ █
```

5. Search for a given command as a regular expression in all manuals.

Code:

```
$ man -k man
```

Output:

```
adhesreghu@adhesreghu-Inspiron-5570:~/Documents/SEM5/OS/Lab$ man -k man
HEAD (1p)          - Simple command line user agent
Thunar (1)          - File Manager for the Xfce Desktop Environment
UPower (7)          - System-wide Power Management
Xcursor (3)          - Cursor management library
aa_help (3)          - AA-lib help string for the default command line parser.
aa_parseoptions (3)  - parse the standard command line options used by AA-lib.
accessdb (8)          - dumps the content of a man-db database in a human readable format
acllocal (1)          - manual page for acllocal 1.15.1
acllocal-1.15 (1)      - manual page for acllocal 1.15.1
aconnect (1)          - ALSA sequencer connection manager
add_key (2)          - add a key to the kernel's key management facility
alsa-info.sh (1)      - command-line utility to gather information about the ALSA subsystem
alsabat (1)          - command-line sound tester for ALSA sound card driver
alsactl_init (7)      - alsa control management - initialization
alsaloop (1)          - command-line PCM loopback
amixer (1)          - command-line mixer for ALSA soundcard driver
anacron (8)          - runs commands periodically
aplay (1)          - command-line sound recorder and player for ALSA soundcard driver
apropos (1)          - search the manual page names and descriptions
apt (8)          - command-line interface
apt-cdrom (8)        - APT CD-ROM management utility
apt-get (8)          - APT package handling utility - - command-line interface
apt-key (8)          - APT key management utility
apt-offline (8)      - Offline APT Package manager
aptd (1)          - package managing daemon proving a D-Bus interface
aptdcon (1)          - command line client for aptdaemon
Archive::Cpio (3pm)    - module for manipulations of cpio archives
arecord (1)          - command-line sound recorder and player for ALSA soundcard driver
```

51. Shut Down

Shutdown from terminal

`shutdown` may be used to halt, power-off or reboot the machine.

Code:

```
$ shutdown
```

To reboot the machine

```
$ shutdown -r
```

Cancel a pending schedule

```
$ shutdown -c
```