

OS-LAB (WEEK1)

--Akhil khubchandani(PES2201800321)

1. Basic LINUX Commands

1. ls - The command lists all files in the directory that match the name of the directory. If the name is left blank, it lists all the files in the current directory.

Syntax : **ls [flags] [name(s)]**

ls -r : Displays the list of directories in reverse order

ls -l : Displays the directories in long list format.

```
akhil@linux:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos webserver.pcap
akhil@linux:~$ ls -r
webserver.pcap Videos Templates Public Pictures Music Downloads Documents Desktop
akhil@linux:~$ ls -l
total 36
drwxr-xr-x 3 akhil akhil 4096 Sep  5 00:17 Desktop
drwxr-xr-x 2 akhil akhil 4096 Sep  5 03:03 Documents
drwxr-xr-x 2 akhil akhil 4096 Sep  5 23:47 Downloads
drwxr-xr-x 2 akhil akhil 4096 Sep  5 03:03 Music
drwxr-xr-x 2 akhil akhil 4096 Sep  5 23:46 Pictures
drwxr-xr-x 2 akhil akhil 4096 Sep  5 03:03 Public
drwxr-xr-x 2 akhil akhil 4096 Sep  5 03:03 Templates
drwxr-xr-x 2 akhil akhil 4096 Sep  5 03:03 Videos
-rw-r--r-- 1 tcpdump tcpdump 1115 Sep  5 23:06 webserver.pcap
akhil@linux:~$
```

2. df - The df command shows the size, used space and available space on the mounted filesystems on your computer.

syntax : **df [OPTION]... [FILE]...**

The two most used and important options for this command are:

-h : human readable format

-x (exclude) : allows you to tell df to exclude certain files / systems you are not interested in

```

akhil@linux:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            3978400         0    3978400  0% /dev
tmpfs           804812         1896    802916  1% /run
/dev/sdb4       28705700 12580416 14644068 47% /
tmpfs           4024044      115156   3908888  3% /dev/shm
tmpfs           5120          4        5116  1% /run/lock
tmpfs           4024044         0    4024044  0% /sys/fs/cgroup
/dev/sdb5       52605328 1146948 48756456  3% /home
/dev/sda1       262144       37388   224756 15% /boot/efi
tmpfs           804808        24    804784  1% /run/user/127
tmpfs           804808        20    804788  1% /run/user/1000

```

```

akhil@linux:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G   0    3.8G   0% /dev
tmpfs           786M  1.9M  785M   1% /run
/dev/sdb4       28G   12G   14G  47% /
tmpfs           3.9G  113M  3.8G   3% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.9G   0    3.9G   0% /sys/fs/cgroup
/dev/sdb5       51G   1.1G   47G   3% /home
/dev/sda1       256M   37M  220M  15% /boot/efi

```

```

akhil@linux:~$ df -h -x squashfs
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G   0    3.8G   0% /dev
tmpfs           786M  1.9M  785M   1% /run
/dev/sdb4       28G   12G   14G  47% /
tmpfs           3.9G  113M  3.8G   3% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.9G   0    3.9G   0% /sys/fs/cgroup
/dev/sdb5       51G   1.1G   47G   3% /home
/dev/sda1       256M   37M  220M  15% /boot/efi
tmpfs           786M   24K  786M   1% /run/user/127
tmpfs           786M   20K  786M   1% /run/user/1000

```

3. free - This command displays the total amount of free space available along with the amount of memory used and swap memory in the system, and also the buffers and caches used by the kernel.

Syntax : **free [flag]**

free -b : Shows free disk space in bytes.

free --giga : Shows free disk space in gigabytes.

```

akhil@linux:~$ free
              total        used        free      shared  buff/cache   available
Mem:      8048088    1559248    4504420       96232    1984420    6082500
Swap:     4459516           0    4459516
akhil@linux:~$ free --giga
              total        used        free      shared  buff/cache   available
Mem:           8           1           4           0           2           6
Swap:          4           0           4
akhil@linux:~$ free -b
              total        used        free      shared  buff/cache   available
Mem:    8241242112    1618501632    4594032640    96985088    2028707840    6208188416
Swap:    4566544384           0    4566544384

```

4. ping - Used to detect network connectivity with a server. It sends an ICMP ECHO_REQUEST to the network host and collects an ICMP ECHO_RESPONSE from a host or gateway.

options

- 4 Use IPv4
- 6 Use IPv6
- c count (stop after sending c amount of requests)
- i interval (send pings at given interval)

```

akhil@linux:~$ ping -c 4 google.com
PING google.com (172.217.166.110) 56(84) bytes of data.
64 bytes from maa05s09-in-f14.1e100.net (172.217.166.110): icmp_seq=1 ttl=120 time=11.5 ms
64 bytes from maa05s09-in-f14.1e100.net (172.217.166.110): icmp_seq=2 ttl=120 time=12.3 ms
64 bytes from maa05s09-in-f14.1e100.net (172.217.166.110): icmp_seq=3 ttl=120 time=12.0 ms
64 bytes from maa05s09-in-f14.1e100.net (172.217.166.110): icmp_seq=4 ttl=120 time=13.8 ms

```

5. ps : Reports a snapshot of the current running processes.

syntax : ps[options]

options : -u <user> : processes running for a particular user

-e : shows **every** process running on the system

without any flag it shows the processes running in the current terminal.


```
akhil@linux:~$ ps
```

PID	TTY	TIME	CMD
8660	pts/0	00:00:00	bash
10072	pts/0	00:00:00	ps

```
akhil@linux:~$ ps -u akhil
```

PID	TTY	TIME	CMD
5819	?	00:00:00	systemd
5822	?	00:00:00	(sd-pam)
5832	?	00:00:00	pulseaudio
5835	?	00:00:00	gnome-keyring-d
5844	?	00:00:00	dbus-daemon
5848	tty2	00:00:00	gdm-x-session
5902	tty2	00:00:00	cinnamon-sessio
5976	?	00:00:00	ssh-agent
5988	?	00:00:00	at-spi-bus-laun
5993	?	00:00:00	dbus-daemon
5996	?	00:00:01	at-spi2-registr
6007	tty2	00:00:00	csd-clipboard
6009	tty2	00:00:00	csd-print-notif
6011	tty2	00:00:00	csd-media-keys
6015	tty2	00:00:00	csd-cursor
6016	tty2	00:00:00	csd-background
6017	tty2	00:00:00	csd-color
6018	tty2	00:00:00	csd-ally-keyboa
6019	tty2	00:00:00	csd-xrandr
6022	tty2	00:00:00	csd-ally-settin
6023	tty2	00:00:00	csd-screensaver
6026	tty2	00:00:00	csd-housekeepin
6027	tty2	00:00:00	csd-automount
6028	tty2	00:00:00	csd-orientation
6030	tty2	00:00:00	csd-wacom
6031	tty2	00:00:00	csd-xsettings
6032	tty2	00:00:00	csd-power
6033	tty2	00:00:00	csd-sound
6036	tty2	00:00:01	csd-keyboard
6037	tty2	00:00:00	csd-mouse
6080	?	00:00:00	dconf-service
6082	?	00:00:00	gvfsd
6092	?	00:00:00	gvfsd-fuse
6094	?	00:00:00	gvfs-udisks2-vo
6104	?	00:00:00	gvfs-mtp-volume
6108	?	00:00:00	gvfs-goa-volume
6112	?	00:00:00	goa-daemon
6119	?	00:00:00	goa-identity-se
6124	?	00:00:00	gvfs-gphoto2-vo
6129	?	00:00:00	gvfs-afc-volume
6136	tty2	00:00:00	csd-printer
6161	tty2	00:00:00	cinnamon-launch
6166	tty2	00:01:00	cinnamon
6183	tty2	00:00:00	xapp-sn-watcher
6213	tty2	00:00:00	blueberry-obex-

```
akhil@linux:~$ ps -e
```

PID	TTY	TIME	CMD
1	?	00:00:03	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	rcu_gp
4	?	00:00:00	rcu_par_gp
6	?	00:00:00	kworker/0:0H-kblockd
9	?	00:00:00	mm_percpu_wq
10	?	00:00:00	ksoftirqd/0
11	?	00:00:01	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-kblockd
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
26	?	00:00:00	kworker/2:0H-kblockd
27	?	00:00:00	cpuhp/3
28	?	00:00:00	idle_inject/3
29	?	00:00:00	migration/3
30	?	00:00:00	ksoftirqd/3
32	?	00:00:00	kworker/3:0H-kblockd
33	?	00:00:00	cpuhp/4
34	?	00:00:00	idle_inject/4
35	?	00:00:00	migration/4
36	?	00:00:00	ksoftirqd/4
38	?	00:00:00	kworker/4:0H-kblockd
39	?	00:00:00	cpuhp/5
40	?	00:00:00	idle_inject/5
41	?	00:00:00	migration/5
42	?	00:00:00	ksoftirqd/5
44	?	00:00:00	kworker/5:0H-kblockd

2. C Program to reverse an array

Main File:

```
#include<stdio.h>
#include "array.h"
int main() {

    int n;
    printf("Enter size of array\n");
    scanf("%d",&n);
    int arr[n];
    printf("Enter elements of array\n");
    for(int i = 0; i<n; i++)
        scanf("%d",&arr[i]);
    printReverse(arr, n);
    return 0;
}
```

Utility Functions File:

```
#include<stdio.h>
#include "array.h"

void printReverse(int arr[], int n) {
    for (int i = n - 1; i ≥ 0 ; i--)
        printf("%d ", arr[i]);
}
```

Header File:

```

    #ifndef ARRAY_H
    #define ARRAY_H    // header gaurding

    void printReverse(int arr[] , int n);

    #endif

```

Makefile:

```

# target: dependancies
#   action
output:main.o array.o
    gcc main.o array.o -o output
main.o:main.c
    gcc -c main.c
array.o:array.c array.h
    gcc -c array.c
clean:
    rm *.o output

```

Output:

```

akhil@linux:/media/akhil/Work/SEM5/OS-Lab/makeFileDemo$ make
gcc -c array.c
gcc main.o array.o -o output
akhil@linux:/media/akhil/Work/SEM5/OS-Lab/makeFileDemo$ ./output
Enter size of array
4
Enter elements of array
1
2
3
4
4 3 2 1 akhil@linux:/media/akhil/Work/SEM5/OS-Lab/makeFileDemo$ 

```

Questions

1. Why do we use Makefile?

Makefiles are special format files that help build and manage the projects automatically. If we have a large amount of files compiling every single file by explicitly specifying them every time some change is made can be an exhaustive task. But with makefiles If certain files are updated, then the makefile rebuilds only those files, thus saving processing power. If none of the files are built, then makefile links and compiles all of them.

2. Is Makefile a shell script?

No. Makefile is a program building tool which runs on Unix, Linux, and their flavors. It aids in simplifying building program executables that may need various modules.

3. What does "clean" do in Makefile?

When you enter 'make clean' in the command line to delete incorrect /inconsistent object and executable files. The compiler can link or compile files incorrectly. To fix this issue, existing .obj files have to be deleted and re-written. Clean is used to complete this task.

4. How does make learn about the last modified files to be compiled?

Make works by checking information about files. Make analyzes relationships between targets and their dependencies, and then checks if the files exist. If they do, it asks the OS for the timestamp when the file was last modified. If a file either does not exist, or exists and is earlier than its corresponding dependent file, then Make rebuilds the targets from the dependents.

5. What does Cflags in Makefile mean?

Many target and prerequisite values have to be replaced with variables and/or certain client dependent patterns. CFLAGS is used to give extra flags/options to the compiler. Eg: -c, -Wall, -o etc.

6. Why do we use -f option with make command?

-f specifies the particular file to be used as the makefile. If unspecified, it scans the entire folder for makefiles .