

## Week-3 Assignment

Omprakash Nara

### Introduction

In Week 3 of my Linux Fundamentals assignment, I explored how Linux serves as the backbone for enterprise systems, servers, and cloud environments. Using Ubuntu on WSL, I practiced navigating the Linux file system, creating and managing files, working with permissions, and running processes. I also explored package management, searching, and system information commands. This practice helped me build confidence in using Linux command line tools effectively.

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ompra> wsl --list --verbose
  NAME      STATE            VERSION
* Ubuntu    Stopped          2
PS C:\Users\ompra> wsl -d Ubuntu
root@OmprakashNara:/mnt/c/Users/ompra# sudo apt update && sudo apt upgrade -y
Hit:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease [126 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease [126 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease [126 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports noble/universe arm64 Packages [15.3 MB]
Get:6 http://ports.ubuntu.com/ubuntu-ports noble/universe Translation-en [5982 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports noble/universe arm64 Components [3573 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports noble/universe arm64 c-n-f Metadata [295 kB]
Get:9 http://ports.ubuntu.com/ubuntu-ports noble/multiverse arm64 Packages [223 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports noble/multiverse Translation-en [118 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports noble/multiverse arm64 Components [31.6 kB]
Get:12 http://ports.ubuntu.com/ubuntu-ports noble/multiverse arm64 c-n-f Metadata [7152 B]
Get:13 http://ports.ubuntu.com/ubuntu-ports noble-updates/main arm64 Packages [1550 kB]
Get:14 http://ports.ubuntu.com/ubuntu-ports noble-updates/main Translation-en [285 kB]
Get:15 http://ports.ubuntu.com/ubuntu-ports noble-updates/main arm64 Components [172 kB]
Get:16 http://ports.ubuntu.com/ubuntu-ports noble-updates/main arm64 c-n-f Metadata [15.0 kB]
Get:17 http://ports.ubuntu.com/ubuntu-ports noble-updates/universe arm64 Packages [1427 kB]
Get:18 http://ports.ubuntu.com/ubuntu-ports noble-updates/universe Translation-en [300 kB]
Get:19 http://ports.ubuntu.com/ubuntu-ports noble-updates/universe arm64 Components [376 kB]
Get:20 http://ports.ubuntu.com/ubuntu-ports noble-updates/universe arm64 c-n-f Metadata [29.9 kB]
Get:21 http://ports.ubuntu.com/ubuntu-ports noble-updates/restricted arm64 Packages [2809 kB]
Get:22 http://ports.ubuntu.com/ubuntu-ports noble-updates/restricted Translation-en [458 kB]
Get:23 http://ports.ubuntu.com/ubuntu-ports noble-updates/restricted arm64 Components [212 B]
Get:24 http://ports.ubuntu.com/ubuntu-ports noble-updates/restricted arm64 c-n-f Metadata [492 B]
Get:25 http://ports.ubuntu.com/ubuntu-ports noble-updates/multiverse arm64 Packages [30.3 kB]
Get:26 http://ports.ubuntu.com/ubuntu-ports noble-updates/multiverse Translation-en [6116 B]
Get:27 http://ports.ubuntu.com/ubuntu-ports noble-updates/multiverse arm64 Components [212 B]
```

```

-v          natural sort of (version) numbers within text
-w, --width=COLS  set output width to COLS. 0 means no limit
-x          list entries by lines instead of by columns
-X          sort alphabetically by entry extension
-Z, --context  print any security context of each file
--zero      end each output line with NUL, not newline
-l         list one file per line
--help      display this help and exit
--version   output version information and exit

```

The SIZE argument is an integer and optional unit (example: 10K is 10\*1024). Units are K,M,G,T,P,E,Z,Y,R,Q (powers of 1024) or KB,MB,... (powers of 1000). Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

The TIME\_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT. FORMAT is interpreted like in date(1). If FORMAT is FORMAT1<newline>FORMAT2, then FORMAT1 applies to non-recent files and FORMAT2 to recent files. TIME\_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale. Also the TIME\_STYLE environment variable sets the default style to use.

The WHEN argument defaults to 'always' and can also be 'auto' or 'never'.

Using color to distinguish file types is disabled both by default and with --color=never. With --color=auto, ls emits color codes only when standard output is connected to a terminal. The LS\_COLORS environment variable can change the settings. Use the dircolors(1) command to set it.

Exit status:

- 0 if OK,
- 1 if minor problems (e.g., cannot access subdirectory),
- 2 if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>  
 Report any translation bugs to <<https://translationproject.org/team/>>  
 Full documentation <<https://www.gnu.org/software/coreutils/ls/>>  
 or available locally via: info '(coreutils) ls invocation'  
 root@OmpakashNara:/mnt/c/Users/ompra/test\_folder#

## File System Navigation

I began by learning how to move through the Linux file system using commands such as pwd, ls, ls -l, and cd. The pwd command showed my current working directory, while ls displayed the files inside a directory. Adding the -l flag gave detailed file permissions, ownership, and size information. Using cd and cd ~, I was able to switch between directories, including my home directory and system folders like /etc. This taught me how Linux organizes system files (/etc), user directories (/home), and logs (/var).

```

Setting up ubuntu-wsl (1.539.2) ...
root@OmpakashNara:/mnt/c/Users/ompra# pwd
/mnt/c/Users/ompra
root@OmpakashNara:/mnt/c/Users/ompra# ls
AppData
'Application Data'
Contacts
Cookies
Documents
Downloads
Favorites
IdeaProjects
links
'Local Settings'
Music
'My Documents'
NTUSER.DAT
NTUSER.DAT{90e78c37-2698-11f0-af42-8c3b4ab6e24c}.TM.blf
NTUSER.DAT{90e78c37-2698-11f0-af42-8c3b4ab6e24c}.TMContainer000000000000000001.regtrans-ms
NTUSER.DAT{90e78c37-2698-11f0-af42-8c3b4ab6e24c}.TMContainer000000000000000002.regtrans-ms
root@OmpakashNara:/mnt/c/Users/ompra# cd
root@OmpakashNara:~# ls -l
root@OmpakashNara:~# cd ~

```

NetHood  
 OneDrive  
 PrintHood  
 Recent  
 'Saved Games'  
 Searches  
 SendTo  
 'Start Menu'  
 Templates  
 Windows  
 week1-README.md  
 ntuser.dat.LOG1  
 ntuser.dat.LOG2  
 ntuser.ini  
 se-training  
 week2\_computersystems

## File and Directory Management

Next, I worked with file and directory management commands. I created directories with `mkdir` and empty files with `touch`. I then copied (`cp`), renamed/moved (`mv`), and removed (`rm`) files to see how Linux handles file operations. To prove removal, I listed files before and after using `rm`, which confirmed that the file was successfully deleted. These steps gave me practical experience in managing a Linux environment where file operations are routine.

```
root@OmpakashNara:~# mkdir test_folder
root@OmpakashNara:~# cd test_folder
root@OmpakashNara:~/test_folder# pwd
/root/test_folder
root@OmpakashNara:~/test_folder# cd
root@OmpakashNara:~# cd /mnt/c/Users/ompra#
-bash: cd: /mnt/c/Users/ompra: No such file or directory
root@OmpakashNara:~# cd /mnt/c/Users/ompra
root@OmpakashNara:/mnt/c/Users/ompra# mkdir test_folder
root@OmpakashNara:/mnt/c/Users/ompra# cd test_folder
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# echo "Hello Linux World" > file1.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# ls
cat hello.txt
file1.txt
cat: hello.txt: No such file or directory
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# echo "This is line 2" >> hello.txt
echo "This is line 3" >> hello.txt
echo "This is line 4" >> hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# echo "This is line 4" >> file1.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# cat file1.txt
Hello Linux World
This is line 4
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# echo "Creation of File2" > file2.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# echo "This is line 2" >> hello.txt
echo "This is line 2" >> hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# cat file2.txt
Creation of File2
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# echo "This is line 2" >> file2.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# cat file2.txt
Creation of File2
This is line 2
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# touch file1.txt file2.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# ls
```

## Viewing and Editing Files

I used commands such as `cat`, `head`, `tail`, and `less` to view file contents. By creating a `hello.txt` file with sample text, I practiced displaying the entire file (`cat`), only the beginning lines (`head`), the last lines (`tail`), and scrolling through larger files (`less`). This helped me understand the different ways to read and review log files, which is especially important for system administration and troubleshooting.

```
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# echo "Hello Linux" > hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# ^C
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# cat hello.txt
Hello Linux
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# head -n 1 hello.txt
Hello Linux
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# tail -n 1 hello.txt
Hello Linux
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# less hello.txt
```

## Permissions and Ownership

I practiced modifying file permissions using the `chmod` command and changing ownership with `chown`. Running `ls -l` before and after these changes clearly showed the effect on the file's permission bits and ownership. This exercise demonstrated how Linux enforces security at the file system level, which is critical for protecting sensitive files on multi-user systems.

```
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# ls -l hello.txt
-rwxrwxrwx 1 root root 12 Oct  1 21:11 hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# chmod 755 hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# ls -l hello.txt
-rwxrwxrwx 1 root root 12 Oct  1 21:11 hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# sudo chown root:root hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# ls -l hello.txt
-rwxrwxrwx 1 root root 12 Oct  1 21:11 hello.txt
root@OmpakashNara:/mnt/c/Users/ompra/test_folder#
```

## Processes and System Information

To monitor the system, I ran commands such as `ps aux` to view running processes and `top` to interactively monitor CPU and memory usage. I also learned to identify the logged-in user with `whoami` and gather system details with `uname -a`. These commands highlighted how administrators keep track of system health and active processes in real time.

```
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# whoami
root
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# uname -a
Linux OmpakashNara 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:31:42 UTC 2025 aarch64 aarch64 aarch64 GNU/
Linux
root@OmpakashNara:/mnt/c/Users/ompra/test_folder#
```

```

root@OmpakashNara:/mnt/c/Users/ompra/test_folder# ps aux | head -5
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.2  0.1 22196 12672 ?        Ss   20:48   0:04 /usr/lib/systemd/systemd --system --deserialize=36
root         2  0.0  0.0   3008  1792 ?        SL   20:48   0:00 /init
root         6  0.0  0.0   3024  2144 ?        SL   20:48   0:00 plan9 --control-socket 7 --log-level 4 --server-fd 8 --pipe-fd 10
--log-truncate
root      307  0.0  0.0   8264  3456 pts/1    Ss   20:48   0:00 /bin/login -f
root@OmpakashNara:/mnt/c/Users/ompra/test_folder# top
top - 21:19:07 up 30 min, 0 user, load average: 0.00, 0.00, 0.00
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7728.6 total, 7126.2 free, 554.5 used, 215.6 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7174.1 avail Mem

```

```

top - 21:18:52 up 30 min, 0 user, load average: 0.00, 0.00, 0.00
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7728.6 total, 7127.1 free, 553.6 used, 215.6 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7175.0 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	22196	12672	8960	S	0.0	0.2	0:04.34	systemd
2	root	20	0	3008	1792	1792	S	0.0	0.0	0:00.03	init-systemd(Ub
6	root	20	0	3024	2144	2048	S	0.0	0.0	0:00.00	init
307	root	20	0	8264	3456	2944	S	0.0	0.0	0:00.00	login
351	root	20	0	5780	4608	3200	S	0.0	0.1	0:00.00	bash
532	root	20	0	3016	1032	896	S	0.0	0.0	0:00.00	SessionLeader
533	root	20	0	3032	1036	896	S	0.0	0.0	0:00.12	Relay(535)
535	root	20	0	5780	4736	3200	S	0.0	0.1	0:00.09	bash
1008	root	20	0	4160	2304	2176	S	0.0	0.0	0:00.01	cron
1009	message+	20	0	9900	4608	3840	S	0.0	0.1	0:00.51	dbus-daemon
1040	root	20	0	17268	7040	6144	S	0.0	0.1	0:00.16	systemd-logind
1044	root	20	0	1755592	11776	9728	S	0.0	0.1	0:00.06	wsl-pro-service
1060	root	20	0	2732	1664	1536	S	0.0	0.0	0:00.00	agetty
1085	root	20	0	2688	1536	1536	S	0.0	0.0	0:00.00	agetty
1101	root	20	0	107904	22016	12672	S	0.0	0.3	0:00.05	unattended-upgr
1716	systemd+	20	0	18700	8448	7296	S	0.0	0.1	0:00.02	systemd-network
1721	root	19	-1	50120	14592	13696	S	0.0	0.2	0:00.18	systemd-journal
5104	syslog	20	0	222812	4224	3712	S	0.0	0.1	0:00.03	rsyslogd
6139	systemd+	20	0	90760	6912	6144	S	0.0	0.1	0:00.08	systemd-timesyn
6194	root	20	0	25096	5696	4544	S	0.0	0.1	0:00.16	systemd-udev
6491	systemd+	20	0	21364	12416	10240	S	0.0	0.2	0:00.05	systemd-resolve
7709	polkitd	20	0	309068	7296	6528	S	0.0	0.1	0:00.10	polkitd
7998	root	20	0	9056	5120	3072	R	0.0	0.1	0:00.01	top

## Searching, Finding Files and Package Management

I used grep to search inside text files and find to locate files by name. For example, searching for the word “Hello” inside hello.txt quickly filtered matching lines. Using find . -name “\*.txt”, I was able to locate all text files under the current directory. These commands are powerful tools for working with large file systems and locating specific content efficiently. I practiced package management with apt. I installed the tree utility to visualize directory structures in a hierarchical way.



```

root@OmrakashNara:/mnt/c/Users/ompra/test_folder# whoami
root
root@OmrakashNara:/mnt/c/Users/ompra/test_folder# uname -a
Linux OmprakashNara 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:31:42 UTC 2025 aarch64 aarch64 aarch64 GNU/Linux
root@OmrakashNara:/mnt/c/Users/ompra/test_folder# grep "Hello" hello.txt
Hello Linux
root@OmrakashNara:/mnt/c/Users/ompra/test_folder# find ~ -name "*.txt"
root@OmrakashNara:/mnt/c/Users/ompra/test_folder# sudo apt install tree -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgll-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 46.0 kB of archives.
After this operation, 160 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports noble/universe arm64 tree arm64 2.1.1-2ubuntu3 [46.0 kB]
Fetched 46.0 kB in 0s (111 kB/s)
Selecting previously unselected package tree.
(Reading database ... 40993 files and directories currently installed.)
Preparing to unpack .../tree_2.1.1-2ubuntu3_arm64.deb ...
Unpacking tree (2.1.1-2ubuntu3) ...
Setting up tree (2.1.1-2ubuntu3) ...
Processing triggers for man-db (2.12.0-4build2) ...
root@OmrakashNara:/mnt/c/Users/ompra/test_folder# tree ~
/root
├── test_folder

2 directories, 0 files
root@OmrakashNara:/mnt/c/Users/ompra/test_folder#

```

## Manual and Help Commands

Finally, I also explored Linux documentation using `man` and `--help` flags, which provide detailed usage instructions for any command. These tools are essential for self-sufficiency in Linux, allowing users to learn and troubleshoot directly within the system.

```

2 directories, 0 files
root@OmrakashNara:/mnt/c/Users/ompra/test_folder# man ls
root@OmrakashNara:/mnt/c/Users/ompra/test_folder# ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                do not ignore entries starting with .
-A, --almost-all        do not list implied . and ..
--author                with -l, print the author of each file
-b, --escape             print C-style escapes for nongraphic characters
--block-size=SIZE        with -l, scale sizes by SIZE when printing them;
                        e.g., '--block-size=M'; see SIZE format below

-B, --ignore-backups     do not list implied entries ending with ~
-c                        with -lt: sort by, and show, ctime (time of last
                        change of file status information);
                        with -l: show ctime and sort by name;
                        otherwise: sort by ctime, newest first

-C                        list entries by columns
--color[=WHEN]           color the output WHEN; more info below
-d, --directory          list directories themselves, not their contents
-D, --dired              generate output designed for Emacs' dired mode
-f                        list all entries in directory order
-F, --classify[=WHEN]   append indicator (one of */=>@|) to entries WHEN

```

```

LS(1)                                     User Commands                                     LS(1)

NAME
    ls - list directory contents

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

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

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters

    --block-size=SIZE
        with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

    -B, --ignore-backups
        do not list implied entries ending with ~

    -c
        with -lt: sort by, and show, ctime (time of last change of file status information); with -l: show ctime and sort
        by name; otherwise: sort by ctime, newest first

    -C
        list entries by columns

Manual page ls(1) line 1 (press h for help or q to quit)

```

```

-v
-w, --width=COLS    natural sort of (version) numbers within text
                    set output width to COLS.  0 means no limit
-x
                    list entries by lines instead of by columns
-X
                    sort alphabetically by entry extension
-Z, --context
--zero
                    print any security context of each file
                    end each output line with NUL, not newline
-l
                    list one file per line
--help
                    display this help and exit
--version
                    output version information and exit

```

The SIZE argument is an integer and optional unit (example: 10K is 10\*1024). Units are K,M,G,T,P,E,Z,Y,R,Q (powers of 1024) or KB,MB,... (powers of 1000). Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

The TIME\_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT. FORMAT is interpreted like in date(1). If FORMAT is FORMAT1<newline>FORMAT2, then FORMAT1 applies to non-recent files and FORMAT2 to recent files. TIME\_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale. Also the TIME\_STYLE environment variable sets the default style to use.

The WHEN argument defaults to 'always' and can also be 'auto' or 'never'.

Using color to distinguish file types is disabled both by default and with --color=never. With --color=auto, ls emits color codes only when standard output is connected to a terminal. The LS\_COLORS environment variable can change the settings. Use the dircolors(1) command to set it.

Exit status:

- 0 if OK,
- 1 if minor problems (e.g., cannot access subdirectory),
- 2 if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>  
 Report any translation bugs to <<https://translationproject.org/team/>>  
 Full documentation <<https://www.gnu.org/software/coreutils/ls>>  
 or available locally via: info '(coreutils) ls invocation'  
 root@OmpakashNara:/mnt/c/Users/ompra/test\_folder#