Experiment No 1

Title:

Implementation of DevOps-Related Linux Commands

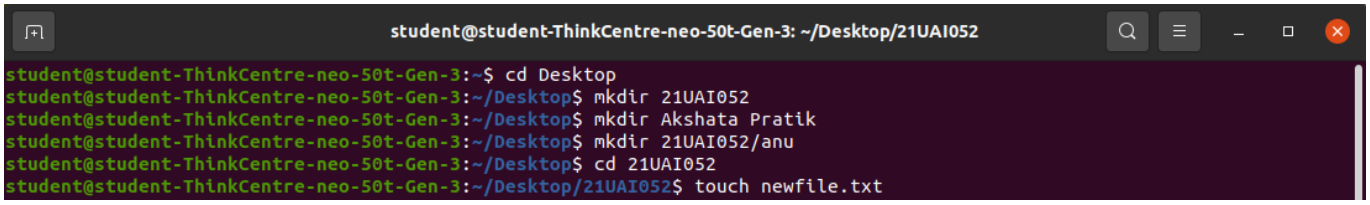
Objective:

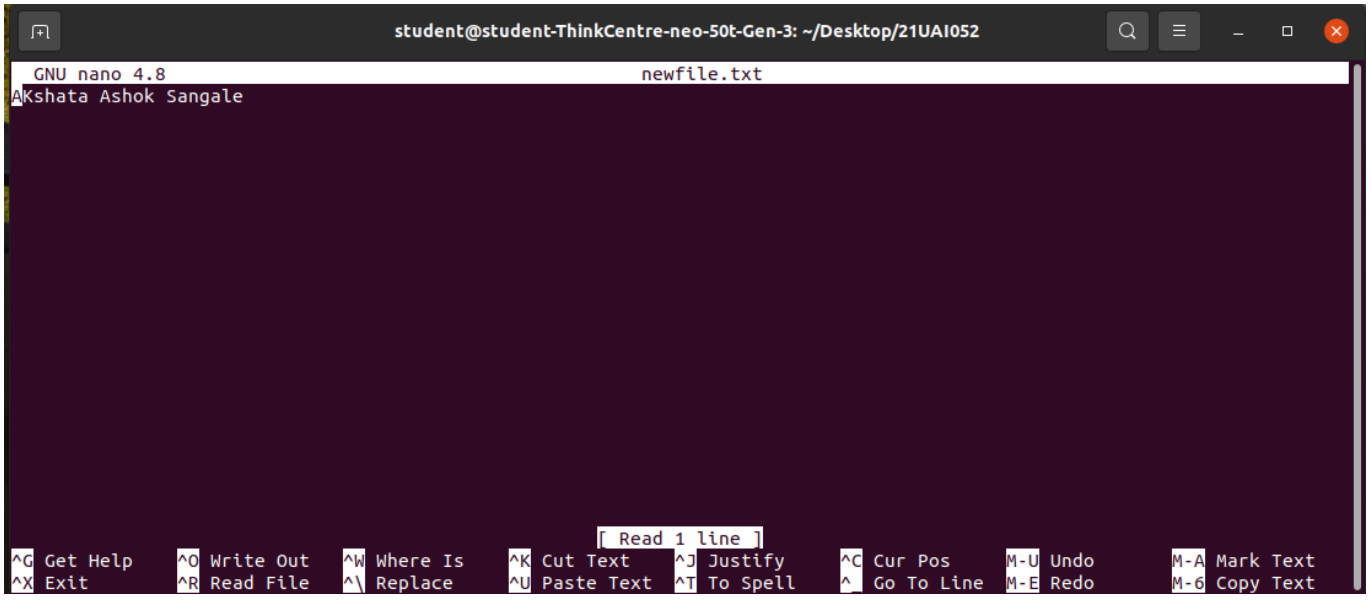
To learn and execute essential Linux commands used in DevOps for file handling, process management, package installation, and text processing.

Important Steps

1.Creating and Navigating Directories:

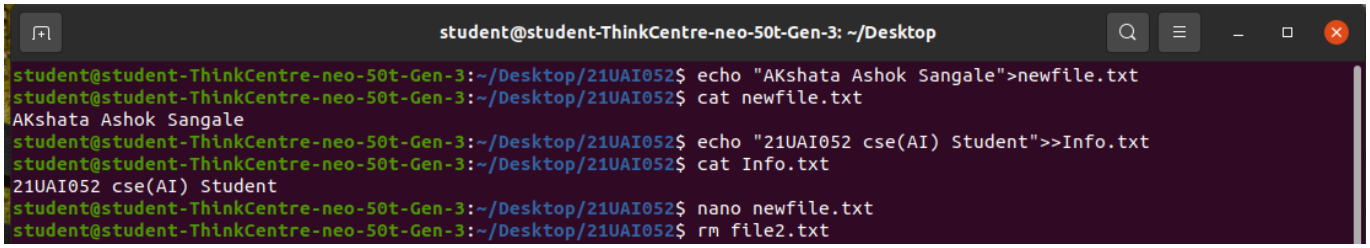
- Used mkdir to create directories.
- Used ls to list files and directories.
- Used cd to change directories and navigate.





2.File Management:

- Used touch to create empty files.
- Used echo to add content to files.
- Used cat to view file contents.
- Used nano to edit files.



4.Copying, Moving, and Deleting Files:

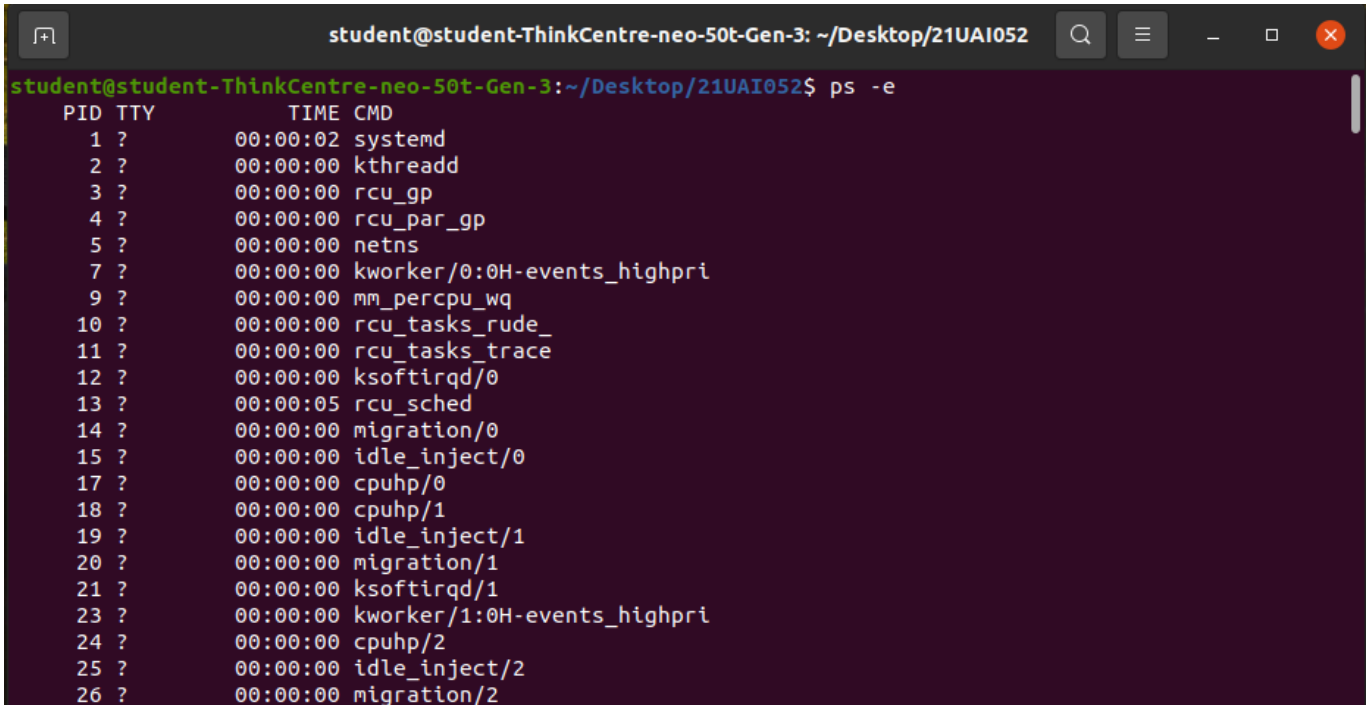
- Used cp to copy files and directories.
- Used mv to move or rename files and directories.
- Used rm and rmdir to delete files and directories.

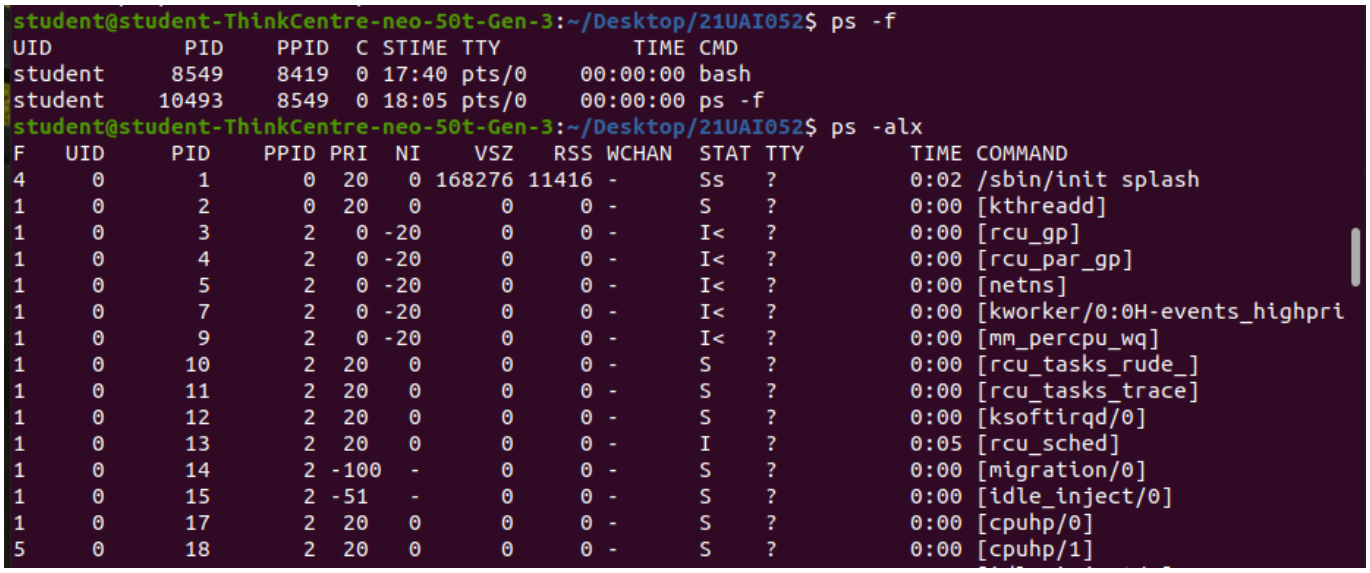
5.Package Management:

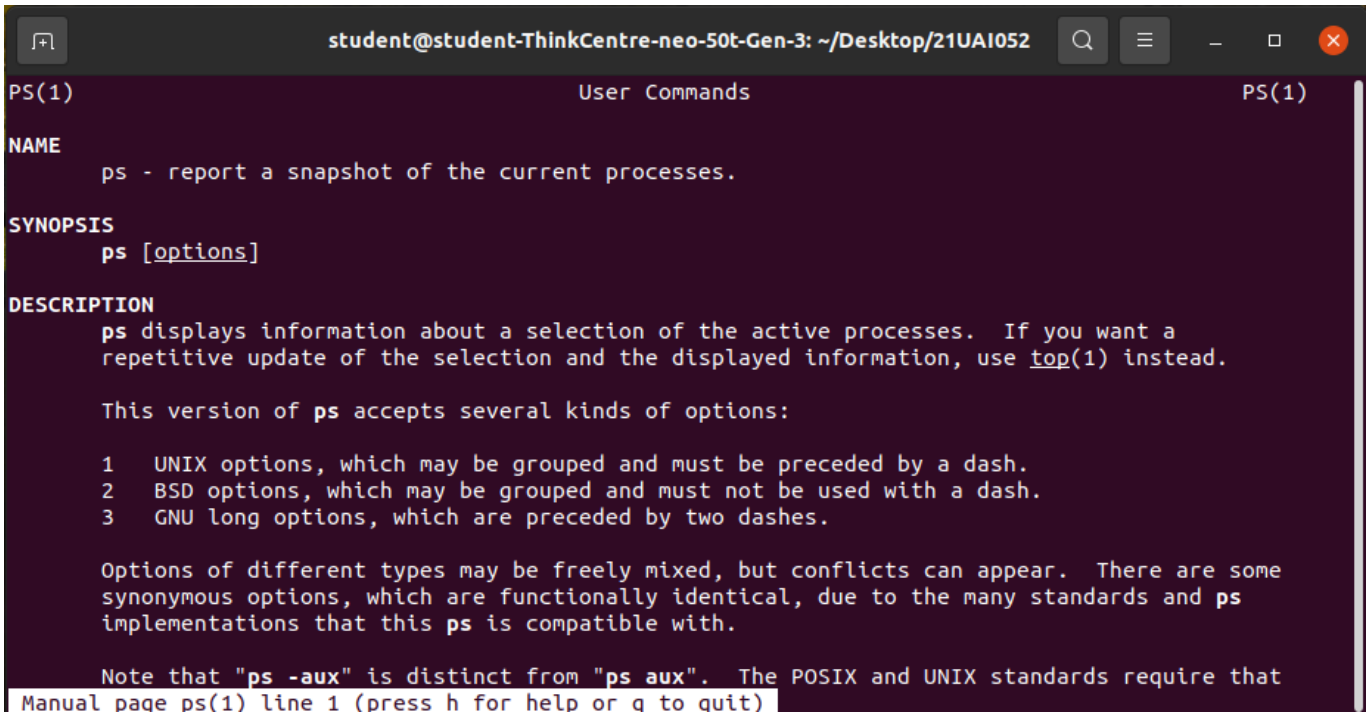
- Used sudo apt-get install to install software packages.

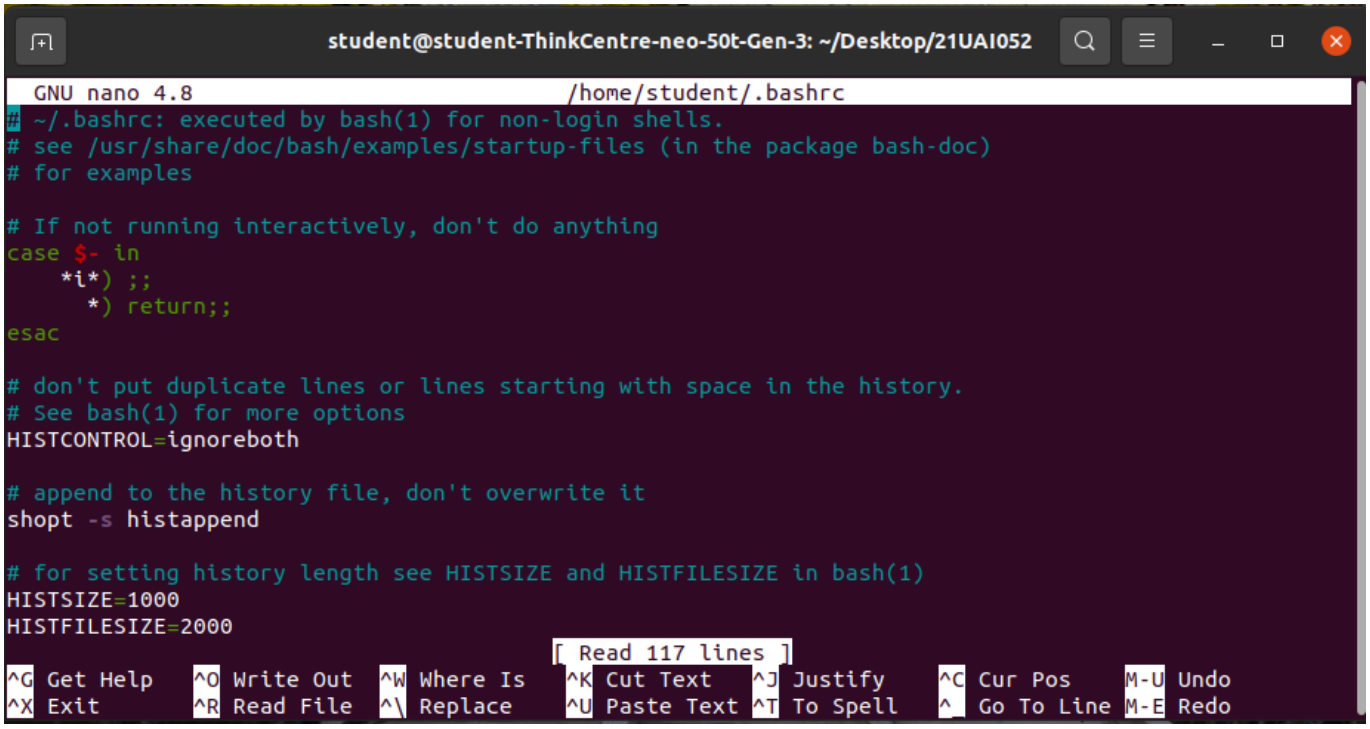
5.**Process and System Management:**

- Used ps -e to list running processes.
- Used grep to filter process information.
- Used ping to check network connectivity.


student@student-ThinkCentre-neo-50t-Gen-3: ~/Desktop/21UAI052
student@student-ThinkCentre-neo-50t-Gen-3:~/Desktop/21UAI052$ ps -e
PID TTY TIME CMD
1 ? 00:00:02 systemd
2 ? 00:00:00 kthreadd
3 ? 00:00:00 rcu_gp
4 ? 00:00:00 rcu_par_gp
5 ? 00:00:00 netns
7 ? 00:00:00 kworker/0:0H-events_highpri
9 ? 00:00:00 mm_percpu_wq
10 ? 00:00:00 rcu_tasks_rude_
11 ? 00:00:00 rcu_tasks_trace
12 ? 00:00:00 ksoftirqd/0
13 ? 00:00:05 rcu_sched
14 ? 00:00:00 migration/0
15 ? 00:00:00 idle_inject/0
17 ? 00:00:00 cpuhp/0
18 ? 00:00:00 cpuhp/1
19 ? 00:00:00 idle_inject/1
20 ? 00:00:00 migration/1
21 ? 00:00:00 ksoftirqd/1
23 ? 00:00:00 kworker/1:0H-events_highpri
24 ? 00:00:00 cpuhp/2
25 ? 00:00:00 idle_inject/2
26 ? 00:00:00 migration/2

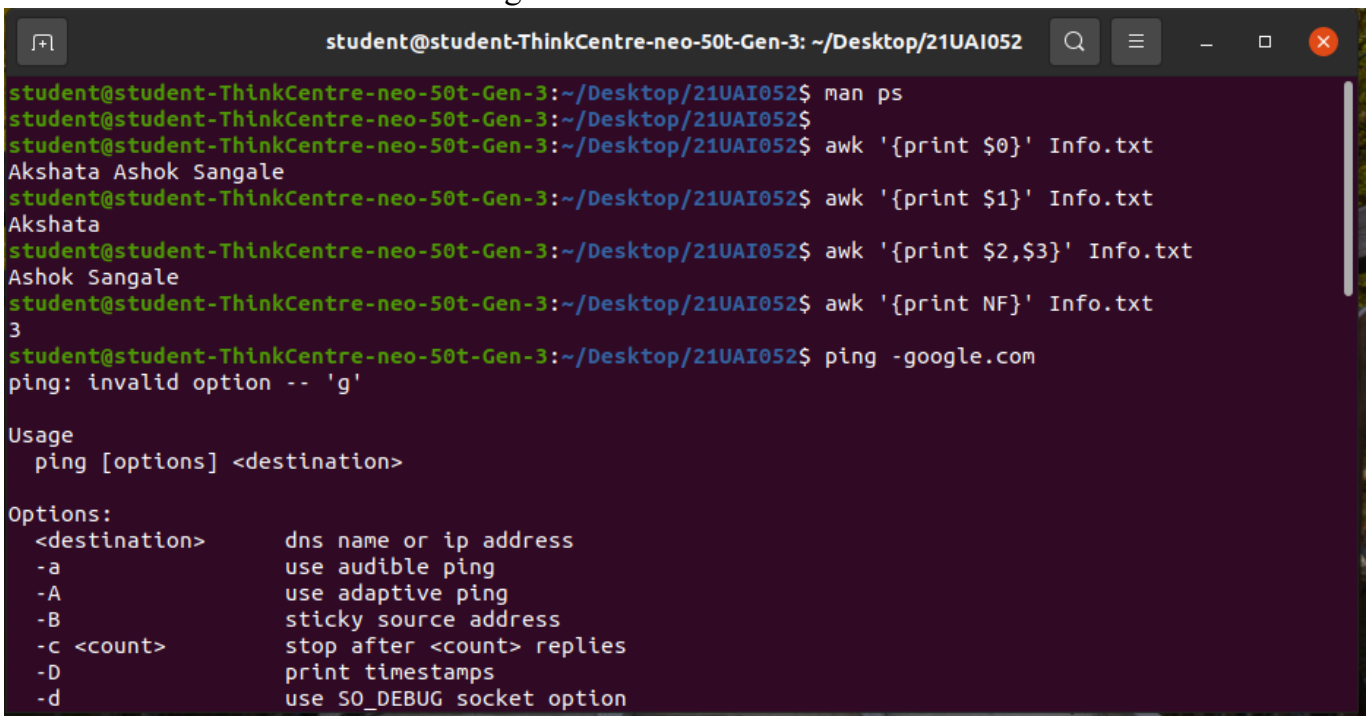

student@student-ThinkCentre-neo-50t-Gen-3:~/Desktop/21UAI052$ ps -f
UID PID PPID C STIME TTY TIME CMD
student 8549 8419 0 17:40 pts/0 00:00:00 bash
student 10493 8549 0 18:05 pts/0 00:00:00 ps -f
student@student-ThinkCentre-neo-50t-Gen-3:~/Desktop/21UAI052$ ps -alx
F UID PID PPID PRI NI VSZ RSS WCHAN STAT TTY TIME COMMAND
4 0 1 0 20 0 168276 11416 - Ss ? 0:02 /sbin/init splash
1 0 2 0 20 0 0 0 - S ? 0:00 [kthreadd]
1 0 3 2 0 -20 0 0 - I< ? 0:00 [rcu_gp]
1 0 4 2 0 -20 0 0 - I< ? 0:00 [rcu_par_gp]
1 0 5 2 0 -20 0 0 - I< ? 0:00 [netns]
1 0 7 2 0 -20 0 0 - I< ? 0:00 [kworker/0:0H-events_highpri
1 0 9 2 0 -20 0 0 - I< ? 0:00 [mm_percpu_wq]
1 0 10 2 20 0 0 0 - S ? 0:00 [rcu_tasks_rude_]
1 0 11 2 20 0 0 0 - S ? 0:00 [rcu_tasks_trace]
1 0 12 2 20 0 0 0 - S ? 0:00 [ksoftirqd/0]
1 0 13 2 20 0 0 0 - I ? 0:05 [rcu_sched]
1 0 14 2 -100 - 0 0 - S ? 0:00 [migration/0]
1 0 15 2 -51 - 0 0 - S ? 0:00 [idle_inject/0]
1 0 17 2 20 0 0 0 - S ? 0:00 [cpuhp/0]
5 0 18 2 20 0 0 0 - S ? 0:00 [cpuhp/1]


student@student-ThinkCentre-neo-50t-Gen-3: ~/Desktop/21UAI052
PS(1) User Commands PS(1)
NAME
ps - report a snapshot of the current processes.
SYNOPSIS
ps [options]
DESCRIPTION
ps displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use top(1) instead.
This version of ps accepts several kinds of options:
1 UNIX options, which may be grouped and must be preceded by a dash.
2 BSD options, which may be grouped and must not be used with a dash.
3 GNU long options, which are preceded by two dashes.
Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options, which are functionally identical, due to the many standards and ps implementations that this ps is compatible with.
Note that "ps -aux" is distinct from "ps aux". The POSIX and UNIX standards require that
Manual page ps(1) line 1 (press h for help or q to quit)



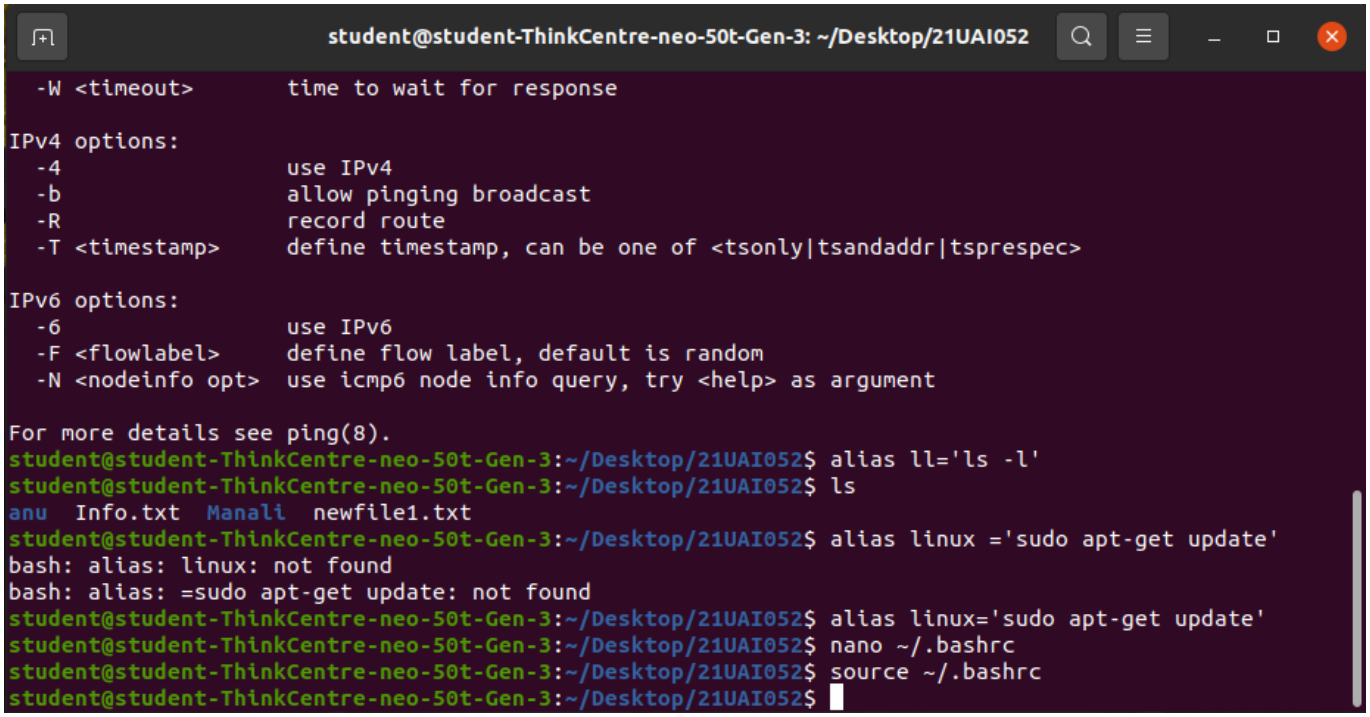
6. Text Processing with awk:

- Extracted columns and rows from structured files.
- Counted fields and rows using NF and NR.



7.Alias and Permissions:

- Created command shortcuts using alias.
- Used ls -l to check file permissions.



Conclusion:

The experiment helped in understanding and executing fundamental Linux commands essential for DevOps tasks. It covered directory handling, file operations, package management, process monitoring, text processing, and system administration, all of which are crucial for DevOps workflows.