EXPERIMENT 5

Command	**Scenario**
free -h	Monitor available RAM in real-time.
cat /proc/meminfo	Check detailed memory information.
top	Real-time process monitoring (find memory-heavy processes).
htop	Interactive real-time process monitoring (alternative to top).
df -h	Check disk space usage to prevent swap/memory issues.
du -sh /var/*	Check directory size to clear space.
vmstat 5	View memory statistics in real-time (observe patterns).
sudo dmidecode -t memory	Check hardware details (RAM modules, slots).
getconf PAGE_SIZE	Check system's memory page size.
sar -r 1 5	Historical resource usage report (memory over time).

EXPERIMENT 4

Command	Scenario
ps aux --sort=-%cpu	Monitor active processes sorted by CPU usage.
pstree -p	View process hierarchy (parent-child relationships).
nice -n 10 python3 heavy_script.py	Run a heavy script with lower CPU priority.
pgrep -f heavy_script.py	Find the PID of a resource-hungry process.
renice -n 15 -p 4567	Change the priority (niceness) of a running process.
pkill firefox	Kill all Firefox browser processes.
xkill	Forcefully close a frozen window by clicking on it.
./long_task.sh & jobs fg %1 bg %1	Handle background and foreground processes (job control).

Command	Scenario	Usage
ps	You want to see all currently running processes for your user.	ps aux or ps -ef
pstree	You want to view the parent-child relationship of processes.	pstree
nice	You want to start a process with lower priority so it doesn't consume much CPU.	nice -n 10 myscript.sh
renice	You want to change the priority of an already running process.	renice -n 5 -p <PID>
kill	You want to terminate a process using its PID.	kill 1234
pkill	You want to kill a process by its name (not PID).	pkill firefox
killall	You want to kill all instances of a process.	killall chrome
xkill	A graphical app is frozen and you want to forcefully kill it by clicking on its window.	xkill (then click on the window)
fg	A process running in background needs to be brought to foreground.	fg %1
bg	A stopped process needs to be resumed in the background.	bg %1

pgrep	You want to find PID of a running process by name.	pgrep python

EXPERIMENT 3

Step	**Command**	**Scenario/Usage**
Creating Student and Faculty Accounts	useradd student1 useradd faculty1 passwd student1	Creates student and faculty user accounts and sets a password for student1.
Assigning Users to Groups	groupadd students groupadd faculty usermod -aG students student1 usermod -aG faculty faculty1	Creates groups and adds users to the appropriate group.
Checking Logged-In Users	who id student1	Displays active users and shows UID, GID for student1.
Deleting a User Who Graduated	userdel -r student1	Removes student1 and their home directory.
Checking Password Expiration Policy	chage -l faculty1	Checks the password expiry settings for faculty1.

EXPERIMENT 2

Step	**Command**	**Scenario/Usage**
Creating a Project Directory and Files	mkdir ProjectX cd ProjectX touch main.py README.md	Creates a new project directory, navigates into it, and creates initial files.
Checking and Managing Files	ls -l cat README.md nano README.md	Lists files with details, displays file content, and edits the README.
Copying and Moving Files	cp main.py backup.py mv backup.py old_version.py	Creates a backup of the script and renames the backup file.
Archiving and Deleting Files	rm old_version.py rm -r ProjectX	Deletes the old backup file and removes the entire project directory if needed.
Checking File Content	head main.py tail -n 5 main.py	Displays the first 10 lines and last 5 lines of the script file.

EXPERIMENT 1

Scenario	**Command**
Display a custom message when a build or script starts	echo "Build started at $(date)"
Clean up the terminal before starting a new debugging session	clear
Safely exit a terminal session or shell script after completion	exit
Log the current system date and time before starting a process	date
Measure the performance or execution time of a build or script	time ./build.sh
Check how long the server has been up (for troubleshooting)	uptime
View the calendar to plan meeting times or script executions	cal
Read the contents of configuration files or logs	cat config.txt or cat logs/error.log
Identify which terminal device a user session is connected to	tty
Learn how to use an unfamiliar command or check syntax	man grep, man find
Locate the path of installed programs or interpreters	which python, which node
Review past commands for reuse or debugging	history
Check your user ID and group for permission checks	id
Confirm your current working directory	pwd
Confirm the current logged-in user (especially in multi-user environments)	whoami