



---

# BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY

---

Department of Computer Science and Engineering

## LABORATORY REPORT

### Abstract

This mid-term report consolidates Linux command-line skills from Weeks 1–3 of CSE 210: file/directory operations, text processing, system administration, permissions, user/group management, process control, archival tasks, networking utilities, and introductory shell scripting. All visuals are print-optimized for crisp black & white output.

### Submitted By

---

**Name:** MD. Khairul Haque  
**Student ID:** 20245103344  
**Section:** 09

---

### Submitted To

---

**Instructor:** Wahida Mahjabin  
**Designation:** Lecturer  
**Department:** Computer Science and Engineering

---

---

Instructor Signature

Date

---

ADVANCING TECHNICAL EDUCATION  
THROUGH RIGOROUS PRACTICUM

## 1 Objective

To compile and demonstrate practical knowledge of Linux Bash commands covered in Weeks 1–3, including file/directory handling, text processing, system administration, file permissions, user/group management, process management, archival operations, and basic network commands. This report also includes introductory shell scripting concepts for foundational programming in Bash.

## 2 Introduction

This mid-term lab report consolidates content from the CSE 310 Lab Manual and lecture slides (Weeks 1–3). Commands are organized into sections with tables for clarity (syntax, description, and Jimmy-themed examples). Exercises were simulated in a Linux environment (e.g., Ubuntu). The focus is on building operating system skills through hands-on command usage and basic scripting.

## 3 File and Directory Commands (Week 1)

Command	Description	Example
<code>mkdir &lt;directory&gt;</code>	Create a directory	<code>mkdir jimmy_adventures</code>
<code>cd &lt;directory&gt;</code>	Change directory	<code>cd jimmy_adventures</code>
<code>touch &lt;file&gt;</code>	Create an empty file	<code>touch jimmy_saving_innocents.txt</code>
<code>mv &lt;src&gt; &lt;dst&gt;</code>	Move or rename a file	<code>mv jimmy_saving_innocents.txt heroic_stories/</code>
<code>cp &lt;src&gt; &lt;dst&gt;</code>	Copy file/directory	<code>cp jimmy_saving_innocents.txt heroic_stories/</code>
<code>rm &lt;file&gt;</code>	Remove a file	<code>rm jimmy_saving_innocents.txt</code>
<code>rmdir &lt;dir&gt;</code>	Remove empty directory	<code>rmdir jimmy_adventures</code>
<code>ls [options]</code>	List files/directories	<code>ls -l</code>
<code>history</code>	Show command history	<code>history</code>

**Observation.** These form the basis of file system navigation, essential for organizing heroic tales like Jimmy’s.

## 4 `ls` — Extended Options

Command	Description	Example
<code>ls</code>	List current directory	<code>ls</code>
<code>ls &lt;dir&gt;</code>	List a directory	<code>ls /home/jimmy_adventures</code>
<code>ls -l</code>	Long listing (meta)	<code>ls -l</code>
<code>ls -a</code>	Include hidden files	<code>ls -a</code>
<code>ls -la</code>	Long + hidden	<code>ls -la</code>
<code>ls -lh</code>	Human-readable sizes	<code>ls -lh</code>
<code>ls -ltr</code>	Oldest first by time	<code>ls -ltr</code>

Command	Description	Example
<code>ls -lt</code>	Newest first by time	<code>ls -lt</code>
<code>ls -R</code>	Recursive listing	<code>ls -R</code>
<code>ls -d */</code>	Only directories	<code>ls -d */</code>
<code>ls -S</code>	Sort by size (desc)	<code>ls -S</code>
<code>ls -r</code>	Reverse order	<code>ls -lr</code>
<code>ls -i</code>	Inode numbers	<code>ls -i</code>
<code>ls -F</code>	Mark types (/,*)	<code>ls -F</code>
<code>ls -color</code>	Colorize output	<code>ls -color</code>
<code>ls -1</code>	One per line	<code>ls -1</code>
<code>ls -X</code>	Sort by extension	<code>ls -X</code>
<code>ls -t</code>	Sort by mtime	<code>ls -t</code>
<code>ls -p</code>	Slash directories	<code>ls -p</code>

**Observation.** Useful for inspecting directories containing Jimmy's heroic documents.

## 5 System Administration Commands (Week 1)

Command	Description	Example
<code>man &lt;cmd&gt;</code>	Manual page	<code>man ls</code>
<code>pkill &lt;name&gt;</code>	Kill by process name	<code>pkill jimmy_story_editor</code>
<code>wget &lt;url&gt;</code>	Download from web	<code>wget https://heroicstories.com/jimmy.txt</code>
<code>free -h</code>	Memory usage	<code>free -h</code>
<code>top</code>	Live process view	<code>top</code>
<code>last</code>	Login history	<code>last</code>
<code>ps aux</code>	Processes (detailed)	<code>ps aux</code>
<code>nano &lt;file&gt;</code>	Terminal editor	<code>nano jimmy.txt</code>
<code>netstat -tuln</code>	Network sockets	<code>netstat -tuln</code>
<code>ifconfig</code>	Network config	<code>ifconfig</code>
<code>echo "..."</code>	Print text	<code>echo "Jimmy saves the day!"</code>
<code>sudo &lt;cmd&gt;</code>	Run as root	<code>sudo apt update</code>
<code>ping &lt;host&gt;</code>	Test connectivity	<code>ping heroicstories.com</code>
<code>su - &lt;user&gt;</code>	Switch user	<code>su - jimmy</code>

*Note: Prefer modern `ss` and `ip` (`iproute2`) over legacy `netstat`/`ifconfig`.*

## 6 Text Processing Commands (Week 1)

Command	Description	Example
cat <file>	Display/concatenate	cat jimmy_saving_innocents.txt
sort <file>	Sort lines	sort heroicJimmy.txt
uniq <file>	Remove duplicates	uniq jimmy_saving_innocents.txt
cut	Extract fields	cut -d',' -f1 heroicJimmy.txt
paste	Merge lines	paste jimmy.txt heroicJimmy.txt
join	Join on key	join fileA fileB
comm	Compare sorted files	comm A.txt B.txt

**Observation.** Ideal for editing and analyzing Jimmy's heroic narratives.

## 7 File Permission Commands (Week 2)

Command	Description	Example
chmod u+rwx <file>	Change permissions (symbolic)	chmod u+rwx jimmy_saving_innocents.txt
chmod 755 <file>	Change permissions (octal)	chmod 755 heroicJimmy.txt
chown <user> <file>	Change ownership	chown jimmy jimmy_saving_innocents.txt
chgrp <group> <file>	Change group	chgrp heroes heroicJimmy.txt

**Observation.** Ensures Jimmy's stories are securely shared with his group of heroes.

## 8 User Management Commands (Week 2)

Command	Description	Example
sudo useradd -m <user>	Add user with home	sudo useradd -m jimmy
sudo passwd <user>	Set user password	sudo passwd jimmy
passwd	Change own password	passwd
sudo passwd -l <user>	Lock account	sudo passwd -l jimmy
sudo usermod -a -G <grp> <user>	Add to group	sudo usermod -a -G heroes jimmy
sudo usermod -g <grp> <user>	Change primary group	sudo usermod -g saviors jimmy
sudo userdel <user>	Delete (keep home)	sudo userdel jimmy
sudo userdel -r <user>	Delete + remove home	sudo userdel -r jimmy

## 9 Group Management Commands (Week 2)

Command	Description	Example
<code>sudo groupadd &lt;grp&gt;</code>	Create group	<code>sudo groupadd heroes</code>
<code>sudo gpasswd &lt;grp&gt;</code>	Set group password	<code>sudo gpasswd heroes</code>
<code>id</code>	Show own IDs/groups	<code>id</code>
<code>id &lt;user&gt;</code>	Show user's IDs/groups	<code>id jimmy</code>
<code>groups &lt;user&gt;</code>	List user's groups	<code>groups jimmy</code>
<code>sudo groupmod -n &lt;new&gt; &lt;old&gt;</code>	Rename group	<code>sudo groupmod -n saviors heroes</code>
<code>sudo groupdel &lt;grp&gt;</code>	Delete group	<code>sudo groupdel heroes</code>

## 10 Process Management Commands (Week 3)

Command	Description	Example
<code>top</code>	Real-time processes	<code>top</code>
<code>ps</code>	Show processes	<code>ps</code>
<code>ps -u &lt;user&gt;</code>	Processes for user	<code>ps -u jimmy</code>
<code>ps -A</code>	All processes	<code>ps -A</code>
<code>kill &lt;PID&gt;</code>	Terminate process	<code>kill 1234</code>
<code>kill -9 &lt;PID&gt;</code>	Force kill (SIGKILL)	<code>kill -9 1234</code>
<code>kill -l</code>	List signals	<code>kill -l</code>
<code>nice -n &lt;prio&gt; &lt;cmd&gt;</code>	Start with priority	<code>nice -n 5 jimmy_story_editor</code>
<code>renice &lt;prio&gt; -p &lt;PID&gt;</code>	Reprioritize by PID	<code>renice 10 -p 1234</code>

## 11 Archival Commands: `tar` (Week 3)

Command	Description	Example
<code>tar -cvf &lt;arc&gt; &lt;files&gt;</code>	Create tar	<code>tar -cvf jimmy_stories.tar jimmy.txt heroic.txt</code>
<code>tar -xvf &lt;arc&gt;</code>	Extract tar	<code>tar -xvf jimmy_stories.tar</code>
<code>tar -cvzf &lt;arc.tar.gz&gt; &lt;files&gt;</code>	Create gzip tar	<code>tar -cvzf jimmy.tar.gz jimmy.txt</code>
<code>tar -xvzf &lt;arc.tar.gz&gt;</code>	Extract gzip tar	<code>tar -xvzf jimmy.tar.gz</code>
<code>tar -cvjf &lt;arc.tar.bz2&gt; &lt;files&gt;</code>	Create bzip2 tar	<code>tar -cvjf jimmy.tar.bz2 heroic.txt</code>
<code>tar -xvjf &lt;arc.tar.bz2&gt;</code>	Extract bzip2 tar	<code>tar -xvjf jimmy.tar.bz2</code>
<code>tar -C &lt;dir&gt; -xvf &lt;arc&gt;</code>	Extract to directory	<code>tar -C /home/jimmy -xvf jimmy_stories.tar</code>

**Observation.** Archives Jimmy's stories for backup.

## 12 Archival Commands: `zip` (Week 3)

Command	Description	Example
<code>zip &lt;zipfile.zip&gt; &lt;files&gt;</code>	Create zip	<code>zip jimmy_stories.zip jimmy_saving_innocents.txt</code>
<code>zip -r &lt;zipfile.zip&gt; &lt;dir&gt;</code>	Zip folder recursively	<code>zip -r jimmy_adventures.zip jimmy_adventures/</code>

## 13 Basic File/Folder Commands (Related to Tasks)

Command	Description	Example
<code>touch &lt;file&gt;</code>	Create empty file	<code>touch heroicJimmy.txt</code>
<code>mkdir &lt;dir&gt;</code>	Create directory	<code>mkdir jimmy_adventures</code>
<code>mv &lt;src&gt; &lt;dst&gt;</code>	Move/rename	<code>mv jimmy_saving_innocents.txt jimmy_adventures/</code>

## 14 Basic Shell Scripting (Introduction)

### 1. Basic Shell Commands

Command	Description	Example
<code>echo</code>	Print text	<code>echo "Jimmy saves innocents!"</code>
<code>pwd</code>	Working directory	<code>pwd</code>
<code>ls</code>	List files	<code>ls -l</code>
<code>cd</code>	Change directory	<code>cd /home/jimmy_adventures</code>
<code>mkdir</code>	Make directory	<code>mkdir jimmy_stories</code>
<code>rmdir</code>	Remove empty dir	<code>rmdir jimmy_stories</code>
<code>touch</code>	Create file	<code>touch jimmy_saving_innocents.txt</code>
<code>rm</code>	Remove file	<code>rm jimmy_saving_innocents.txt</code>
<code>cp</code>	Copy file	<code>cp jimmy_saving_innocents.txt heroicJimmy.txt</code>
<code>mv</code>	Move/rename	<code>mv heroicJimmy.txt jimmy_adventures/</code>

### 2. Variables in Shell

```

1 name="JIMMY"
2 age=20
3 echo "Name: $name"
4 echo "Age: $age"

```

*Notes:* Variables are case-sensitive; no spaces around `=`.

### 3. Reading Input

```
1 echo "Enter your hero's name: "  
2 read name  
3 echo "Hello $name, the savior!"
```

### 4. Simple if Statement

```
1 num=10  
2 if [ $num -gt 5 ]; then  
3     echo "Jimmy's bravery level is greater than 5"  
4 fi
```

*Comparators: -gt, -lt, -eq.*

### 5. if-else Statement

```
1 num=3  
2 if [ $num -gt 5 ]; then  
3     echo "Jimmy's bravery is greater than 5"  
4 else  
5     echo "Jimmy's bravery is 5 or less"  
6 fi
```

### 6. if-elif-else Statement

```
1 num=7  
2 if [ $num -gt 10 ]; then  
3     echo "Jimmy's bravery is greater than 10"  
4 elif [ $num -eq 7 ]; then  
5     echo "Jimmy's bravery is 7"  
6 else  
7     echo "Jimmy's bravery is less than 10 and not 7"  
8 fi
```

### 7. Checking Strings in if

```
1 name="Jimmy"  
2 if [ "$name" = "Jimmy" ]; then  
3     echo "Hello Jimmy, the hero!"  
4 else  
5     echo "You are not Jimmy"  
6 fi
```

*Use = for equality and != for inequality.*

### 8. Checking File Conditions

```
1 file="jimmy_saving_innocents.txt"  
2 if [ -f "$file" ]; then  
3     echo "$file exists"  
4 else  
5     echo "$file does not exist"  
6 fi  
7 # -f: regular file, -d: directory, -e: exists (file or dir)
```

## 9. Combining Conditions

```

1 num=15
2 if [ $num -gt 10 ] && [ $num -lt 20 ]; then
3     echo "Jimmy's bravery is between 10 and 20"
4 fi
5 if [ $num -lt 10 ] || [ $num -eq 15 ]; then
6     echo "Either less than 10 or exactly 15"
7 fi

```

**Observation.** Shell scripting automates repetitive tasks like managing Jimmy's story files.

## 15 Additional Essential Linux Commands (Extended Coverage)

### 15.1 File and Directory Inspection / Manipulation

Command	Description	Jimmy-Themed Example
<code>file &lt;path&gt;</code>	Detect file type	<code>file jimmy_saving_innocents.txt</code>
<code>unzip &lt;zipfile&gt;</code>	Extract ZIP archive	<code>unzip jimmy_adventures.zip -d ./extracted/</code>
<code>locate &lt;name&gt;</code>	Find by name (DB)	<code>locate jimmy_saving_innocents.txt</code>
<code>find &lt;dir&gt; -name "&lt;pat&gt;"</code>	Real-time file search	<code>find . -type f -name "jimmy*"</code>
<code>du -sh &lt;dir&gt;</code>	Disk usage summary	<code>du -sh jimmy_adventures/</code>
<code>ln &lt;src&gt; &lt;dst&gt;</code>	Hard link	<code>ln heroic.txt heroic_link</code>
<code>ln -s &lt;src&gt; &lt;dst&gt;</code>	Symbolic link	<code>ln -s /stories/jimmy.txt latest_hero.txt</code>

**Observation.** `locate` uses a database (`updatedb`); `find` searches live directories and supports predicates like `-size`, `-mtime`, `-exec`.

### 15.2 Text Editors (Partial)

Editor	Description	Example
<code>vi &lt;file&gt;</code>	Modal editor (Vim/Vi)	<code>vi jimmy_notes.md</code>
<code>jed &lt;file&gt;</code>	Lightweight editor	<code>jed jimmy_manifesto.txt</code>

### 15.3 Text Processing / Manipulation

Command	Description	Example
<code>sed 's/A/B/g' &lt;file&gt;</code>	Stream edit (replace)	<code>sed -i 's/Jimmy/Hero Jimmy/g' story.txt</code>
<code>awk 'prog' &lt;file&gt;</code>	Pattern scanning	<code>awk -F, 'print \$1' heroes.csv</code>
<code>diff &lt;A&gt; &lt;B&gt;</code>	Line diff (unified)	<code>diff -u v1.txt v2.txt</code>



Command	Description	Example
<code>tee &lt;file&gt;</code>	TTY + file output	<code>echo "legend"   tee title.txt</code>

## 15.4 User and System Info

Command	Description	Example
<code>whoami</code>	Current user	<code>whoami</code>
<code>uname -a</code>	Kernel/system info	<code>uname -a</code>
<code>hostname</code>	Get/set hostname	<code>hostname hero-station</code>

## 15.5 Process and System Management (Advanced)

Command	Description	Example
<code>htop</code>	Interactive process viewer	<code>htop</code>
<code>time &lt;cmd&gt;</code>	Measure runtime	<code>time ./compile_jimmy.sh</code>
<code>systemctl &lt;op&gt; &lt;svc&gt;</code>	Manage systemd services	<code>sudo systemctl restart ssh</code>
<code>watch &lt;cmd&gt;</code>	Repeat command	<code>watch -n 2 'ls -lh stories/'</code>
<code>jobs</code>	List shell jobs	<code>jobs</code>
<code>shutdown -r now</code>	Reboot/shutdown	<code>sudo shutdown -h +10</code>

## 15.6 Network and Transfer (Advanced)

Command	Description	Example
<code>curl -O &lt;url&gt;</code>	Transfer via URL	<code>curl -O https://example.com/jimmy.txt</code>
<code>scp &lt;src&gt; &lt;user@host:dst&gt;</code>	Secure copy (SSH)	<code>scp jimmy.txt ubuntu@server:/srv/heroes/</code>
<code>rsync -av &lt;src&gt; &lt;dst&gt;</code>	Efficient sync	<code>rsync -av stories/ backup:/data/stories/</code>
<code>ip a</code>	Network interfaces	<code>ip a</code>
<code>traceroute &lt;host&gt;</code>	Packet path	<code>traceroute heroicstories.com</code>
<code>nslookup &lt;name&gt;</code>	DNS query (simple)	<code>nslookup heroicstories.com</code>
<code>dig &lt;name&gt; ANY</code>	DNS query (detailed)	<code>dig heroicstories.com A +short</code>

**Observation.** Prefer `ip` over `ifconfig`. For DNS, `dig` provides structured output suitable for scripting.

## 15.7 Miscellaneous

---

Command	Description	Example
<code>alias ll='ls -lh'</code>	Create alias	<code>alias gs='git status'</code>
<code>unalias &lt;name&gt;</code>	Remove alias	<code>unalias ll</code>
<code>cal [m] [y]</code>	Calendar	<code>cal 9 2025</code>
<code>apt / dnf</code>	Package managers	<code>sudo apt install htop</code>

---

## 16 Conclusion

This report covers essential Linux commands and introductory scripting for the mid-term lab, with Jimmy-themed examples for engagement. The B&W design ensures maximum legibility in print and consistent results on any grayscale printer. Future weeks will cover advanced scripting and algorithms.

## 17 References

- CSE 310 Lab Manual: Linux Bash Commands
- Week 1–3 Lecture Slides
- The Linux Documentation Project: <https://www.tldp.org>