10 Essential Linux Commands - File Management & Permissions

Yagyansh Singh Ahlawat (2501010120)

ETCCCP105 Assignment - Linux Practice (CO2)

Command 1: pwd (Print Working Directory)

Purpose: Display the current directory path

```bash
pwd
```

Expected Output:

```
/home/yagyansh/CareerDevelopment
```

Real-world Use: Know your current location before performing file operations

Screenshot Instructions: Run `pwd` in terminal and capture output

---

Command 2: mkdir (Make Directory)

Purpose: Create new directories for organizing career files

```bash
mkdir -p ~/CareerDevelopment/Certifications/CEH
```

Explanation:

- `-p`: Creates parent directories if they don't exist
- `~`: Home directory shortcut

Screenshot Instructions: Run command, then use `ls -R` to show created structure

---

Command 3: ls (List Directory Contents)

Purpose: View files and directories with detailed information

```bash
ls -lah ~/CareerDevelopment
```

Options Explained:

- `-l`: Long format (permissions, size, date)
- `-a`: Show hidden files (starting with .)
- `-h`: Human-readable file sizes (KB, MB, GB)

Expected Output:

```
total 24K
drwxr-xr-x  6 yagyansh yagyansh 4.0K Nov 29 10:30 .
drwxr-xr-x 25 yagyansh yagyansh 4.0K Nov 29 10:25 ..
drwxr-xr-x  3 yagyansh yagyansh 4.0K Nov 29 10:30 Certifications
drwxr-xr-x  2 yagyansh yagyansh 4.0K Nov 29 10:30 Projects
-rw-r--r--  1 yagyansh yagyansh  523 Nov 29 10:30 README.md
```

Screenshot Instructions: Capture full ls output showing permissions and details

---

Command 4: touch (Create Empty Files)

Purpose: Create placeholder files or update timestamps

```bash
touch ~/CareerDevelopment/Resume_CVs/Yagyansh_CV_v1.pdf
touch ~/CareerDevelopment/JobApplications/Applied/company_{1..5}.txt
```

Advanced Use: Second command creates 5 files at once using brace expansion

Screenshot Instructions: Run touch commands, then `ls -l` to verify creation

---

Command 5: cp (Copy Files)

Purpose: Duplicate files and directories for backup

```bash
cp -r ~/CareerDevelopment/Resume_CVs ~/CareerBackup/
```

Options:

- `-r`: Recursive (copy directories and contents)
- `-v`: Verbose (show what's being copied)
- `-i`: Interactive (ask before overwriting)

Use Case: Backup resume versions before major changes

Screenshot Instructions: Show before and after directory listings

---

Command 6: mv (Move/Rename Files)

Purpose: Relocate files or rename them

```bash
# Rename file
mv ~/CareerDevelopment/resume.pdf ~/CareerDevelopment/Yagyansh_Resume_2025.pdf

# Move to different directory
mv ~/Downloads/certificate.pdf ~/CareerDevelopment/Certifications/
```

Real-world Scenario: Organize downloaded certificates into proper folders

Screenshot Instructions: Show file before and after mv operation

---

Command 7: chmod (Change File Permissions)

Purpose: Control read, write, execute permissions

```bash
# Make script executable
chmod +x ~/CareerDevelopment/scripts/automation.sh

# Set specific permissions (owner: rwx, group: rx, others: r)
chmod 754 ~/CareerDevelopment/scripts/backup.sh
```

Permission Numbers:

- 7 = rwx (read, write, execute)
- 6 = rw- (read, write)
- 5 = r-x (read, execute)
- 4 = r-- (read only)

Security Use: Protect sensitive career documents from unauthorized access

Screenshot Instructions: Show `ls -l` before and after chmod

Command 8: chown (Change Ownership)

Purpose: Transfer file ownership (requires sudo for system files)

```bash
# Change owner
sudo chown yagyansh:yagyansh ~/CareerDevelopment/projects/*

# Change owner recursively
sudo chown -R yagyansh:students ~/CareerDevelopment/shared/
```

Format: `chown user:group filename`

Use Case: Manage shared project files with proper ownership

Screenshot Instructions: Capture ownership change with `ls -l`

Command 9: find (Search for Files)

Purpose: Locate files based on various criteria

```bash
```

```
# Find all PDF files in career directory
find ~/CareerDevelopment -name "*.pdf" -type f

# Find files modified in last 7 days
find ~/CareerDevelopment -mtime -7

# Find files larger than 1MB
find ~/CareerDevelopment -size +1M

# Find and execute command on results
find ~/CareerDevelopment/Projects -name "*.py" -exec wc -l {} \;
```

Advanced Options:

- `-name`: Search by filename pattern
- `-type f`: Files only (d for directories)
- `-mtime`: Modified time
- `-size`: File size filter
- `-exec`: Execute command on results

Cybersecurity Application: Find all bash scripts for security review

Screenshot Instructions: Run multiple find variants and show results

---

Command 10: tar (Archive and Compress)

Purpose: Create compressed backups of directories

```bash
# Create compressed archive
tar -czvf career_backup_$(date +%Y%m%d).tar.gz ~/CareerDevelopment/

# Extract archive
tar -xzvf career_backup_20251129.tar.gz

# List archive contents without extracting
tar -tzvf career_backup_20251129.tar.gz
```

Options Explained:

- `-c`: Create archive
- `-x`: Extract archive
- `-t`: List contents
- `-z`: Compress with gzip
- `-v`: Verbose output
- `-f`: Specify filename

Professional Use: Regular backups before major resume updates

Screenshot Instructions: Show tar creation process and file size comparison

---

Bonus Commands for Cybersecurity Career

Command 11: grep (Pattern Matching)

```bash
# Search for skill keywords in resume
grep -i "python\|cybersecurity\|penetration" ~/CareerDevelopment/Resume_CVs/*.txt

# Count occurrences
grep -c "security" ~/CareerDevelopment/JobApplications/Applied/*.txt
```

Command 12: df (Disk Usage)

```bash
# Check available disk space
df -h
```

Command 13: ps (Process Status)

```bash
# View running processes
ps aux | grep python
```

Summary Table

Command	Primary Use	Cybersecurity Relevance
pwd	Navigation	Know script execution context
mkdir	Organization	Create structured project directories
ls	Inspection	Audit file permissions
touch	File creation	Initialize log files
cp	Backup	Preserve original files before changes
mv	Organization	Rename files to naming conventions
chmod	Security	Restrict access to sensitive data
chown	Management	Control file ownership
find	Search	Locate configuration files quickly
tar	Backup	Compress and archive project data

Command Execution Checklist

For each command, ensure you capture:

□ Command syntax typed in terminal
□ Full output displayed
□ File/directory state before operation
□ File/directory state after operation
□ Any error messages (if applicable)
□ Timestamp visible in terminal

Screenshot Organization:

```
linux_scripts/command_screenshots/
├── 01_pwd_command.png
├── 02_mkdir_command.png
├── 03_ls_command.png
├── 04_touch_command.png
├── 05_cp_command.png
├── 06_mv_command.png
├── 07_chmod_command.png
├── 08_chown_command.png
├── 09_find_command.png
└── 10_tar_command.png
```

Real-World Scenario: Complete Workflow

```bash
# 1. Create project structure
mkdir -p ~/CyberProject/{src,docs,tests}

# 2. Navigate to project
cd ~/CyberProject

# 3. Create files
touch src/scanner.py docs/README.md

# 4. Set permissions
chmod 750 src/scanner.py

# 5. Verify structure
ls -lR

# 6. Backup project
tar -czvf ~/Backups/cyber_project_$(date +%Y%m%d).tar.gz ~/CyberProject

# 7. Find Python files
find ~/CyberProject -name "*.py"

# 8. Change ownership (if needed)
sudo chown -R $USER:$USER ~/CyberProject

# 9. Copy to shared location
cp -r ~/CyberProject ~/SharedProjects/

# 10. Check disk usage
df -h ~
```

Assignment Completion Notes:

- All commands executed on: [Your Linux Distribution]
- Terminal: [bash/zsh]
- Date: November 29, 2025

- Student: Yagyansh Singh Ahlawat (2501010120)