🐧 LINUX FUNDAMENTALS & CLI BASICS

1. WHAT IS LINUX?

- **Family** of open-source, Unix-like operating systems built on the **Linux kernel**
- An **Operating System (OS)** acts as the interface between user and hardware components

2. CLI = COMMAND-LINE INTERFACE

- **Text-based interface** where you type commands → computer executes them
- **Essential for DevOps** for speed, automation, and control

3. SEVEN COMMAND CATEGORIES

Category	Purpose	Examples
System	System management & control	`reboot`, `shutdown`, `systemctl`
Hardware	Hardware information & monitoring	`lscpu`, `lsblk`, `free`, `df`
File	File and directory operations	`ls`, `cp`, `mv`, `rm`, `mkdir`
Permission	Access control & security	`chmod`, `chown`, `umask`
User	User and group management	`useradd`, `usermod`, `groupadd`
Search	Finding files and content	`grep`, `find`, `locate`, `which`
Software	Package management	`apt`, `yum`, `dnf`, `pip`

🧠 Memory Hook: S-H-F-P-U-S-S

- **S**ystem → **H**ardware → **F**ile → **P**ermission → **U**ser → **S**earch → **S**oftware

4. ESSENTIAL SHORTCUTS & PRIVILEGE MANAGEMENT

Privilege Escalation

Task	Command
Become **root user**	`sudo -i` or `sudo su -`

Task	Command
Run single command as root	`sudo <command>`
Return to normal user	`exit`
Switch to another user	`su - username`

Terminal Shortcuts

Shortcut	Action
`Ctrl + L` or `clear`	Clear terminal screen
`Ctrl + C`	Terminate current process
`Ctrl + D`	Logout/exit shell
`Ctrl + A`	Move to beginning of line
`Ctrl + E`	Move to end of line
`Ctrl + U`	Clear line before cursor
`Ctrl + K`	Clear line after cursor
`Tab`	Auto-complete commands/files

5. BASIC COMMAND STRUCTURE

```
command [options] [arguments]
```

- **Command**: The program to execute
- **Options**: Modify behavior (usually `-` or `--`)
- **Arguments**: Targets or parameters for the command

6. QUICK START GUIDE

Immediate Practice Commands

```
# Check current user
whoami

# See current directory
pwd

# List files
```

```
ls

# Check system information
uname -a

# View disk usage
df -h

# Check memory
free -h
```

Navigation Basics

```
# Change directory
cd /path/to/directory

# Go to home directory
cd ~  OR  cd

# Go up one level
cd ..

# Previous directory
cd -
```

7. STUDY TIPS

For Beginners:

1. **Practice daily** - Use the terminal for common tasks
2. **Learn help system** - Use `man command` or `command --help`
3. **Start with file operations** - Master `ls`, `cd`, `cp`, `mv`, `rm`
4. **Understand permissions early** - Critical for security

For DevOps:

1. **Automate everything** - Script repetitive tasks
2. **Master text processing** - `grep`, `awk`, `sed`
3. **Learn process management** - `ps`, `top`, `kill`
4. **Practice privilege escalation** - Safe `sudo` usage

🐧 LINUX SYSTEM COMMANDS - COMPREHENSIVE CHEAT SHEET

1. OS & KERNEL INFORMATION

Task	Command	Sample Output
OS Name	`uname`	Linux
Kernel Version	`uname -r`	6.5.0-28-generic
System Architecture	`uname -m`	x86_64
Full System Details	`uname -a`	Linux host 6.5... x86_64 GNU/Linux

2. HOSTNAME MANAGEMENT

Action	Command
Show Current Hostname	`hostname`
Change Hostname	`sudo hostnamectl set-hostname "my-server"`
Note: Logout → login to see changes	

3. NETWORK CONFIGURATION

Information	Command
All Network Addresses	`ip addr` or `ip a`
Routing Table	`ip route`
Network Interfaces	`ip link show`
Legacy Tool (if installed)	`ifconfig`

4. CALENDAR & DATE COMMANDS

Calendar Views

Task	Command	Description
Current Month	`cal`	Grid calendar for current month
Specific Year	`cal 2024`	Entire year calendar
Specific Month/Year	`cal 8 2012`	August 2012 calendar

Date Formatting

Format	Command	Example Output
Day of Month	`date +'%d'`	05
Month Number	`date +'%m'`	12
Year (4-digit)	`date +'%Y'`	2024
Hours (24h)	`date +'%H'`	19
Minutes	`date +'%M'`	34
Seconds	`date +'%S'`	07
Full Date (Slash)	`date +'%d/%m/%y'`	12/05/24
ISO-8601 Format	`date +'%F'`	2024-12-05
Day Name	`date +'%A'`	Thursday
Month Name	`date +'%B'`	December
Custom Format	`date +'%A, %B %d, %Y'`	Thursday, December 05, 2024

5. TIME & TIMEZONE MANAGEMENT

Action	Command
Show Current Time + Zone	`timedatectl`
List Available Timezones	`timedatectl list-timezones`
Change Timezone	`sudo timedatectl set-timezone Asia/Kolkata`
Set Hardware Clock	`sudo timedatectl set-local-rtc 1`

6. USERS & PROCESSES

Need	Command
Current Username	`whoami`
All Logged-in Users	`who` or `w`
Running Processes (Simple)	`ps`

Need	Command
All Processes	`ps aux`
Background Jobs	`jobs`
Kill Process	`kill -9 4430`
Kill by Name	`pkill process-name`

7. SYSTEM UPTIME & LOAD

Information	Command	Sample Output
Uptime with Load	`uptime`	14:20 up 5 days, load 0.20
Boot Time	`uptime -s`	2024-05-01 09:16:40
Pretty Format	`uptime -p`	up 5 days, 3 hours
System Load	`cat /proc/loadavg`	0.20 0.18 0.15 1/500 3412

8. PRIVILEGE & TERMINAL MANAGEMENT

Privilege Escalation

Task	Command
Become Root	`sudo -i` or `sudo su -`
Run Single Command as Root	`sudo command`
Return to Normal User	`exit`

Terminal Shortcuts

Shortcut	Action
`Ctrl + L` or `clear`	Clear Screen
`Ctrl + C`	Cancel/Interrupt
`Ctrl + D`	Exit Shell
`Ctrl + A`	Move to Line Start

Shortcut	Action
`Ctrl + E`	Move to Line End
`Ctrl + U`	Clear to Line Start
`Ctrl + K`	Clear to Line End
`Tab`	Auto-complete

9. QUICK SYSTEM CHECKS

One-Line System Info

```
# Comprehensive system overview
uname -a && hostname && date && uptime

# Memory and CPU info
free -h && lscpu | grep "Model name"

# Disk space check
df -h | grep -v tmpfs
```

10. PRACTICAL USAGE SCENARIOS

For System Administration:

- Use `timedatectl` for timezone management in cloud deployments
- Monitor system load with `uptime` during performance checks
- Use `ip addr` for network troubleshooting

For Scripting:

```
# Use date in backup scripts
backup_file="backup_$(date +'%Y%m%d_%H%M%S').tar.gz"

# Check system uptime in monitoring scripts
if uptime -p | grep -q "day"; then
    echo "System has been up for multiple days"
fi
```

🐧 LINUX HARDWARE & FILE COMMANDS - COMPREHENSIVE CHEAT SHEET

1. HARDWARE & RESOURCE INSPECTION 🖥

Resource	Command	Purpose & Output
CPU Information	`lscpu`	CPU model, cores, architecture, flags
Live System Monitor	`top`	Real-time CPU/MEM usage (press `q` to quit)
Memory Summary	`free`	RAM in KB
Memory in MB	`free -m`	RAM in megabytes
Memory in GB	`free -g`	RAM in gigabytes
Disk Space	`df -h`	Human-readable disk usage
Block Devices	`lsblk`	List storage devices (EBS, partitions, disks)
Hardware Info	`lshw`	Detailed hardware configuration

2. FILE TYPE LEGEND (`ls -l` FIRST CHARACTER)

Symbol	File Type	Examples
`-`	Regular file	Documents, scripts, images
`d`	Directory	Folders
`b`	Block device	Hard drives, EBS volumes
`c`	Character device	Terminals, serial ports
`l`	Symbolic link	Shortcuts to files/directories
`p`	Named pipe	Inter-process communication
`s`	Socket	Network communication

3. CREATING FILES & DIRECTORIES 📂

File Creation

Need	Command	Result
Single file	`touch file1`	Creates empty file1
Multiple files	`touch file1 file2 file3`	Creates three files

Need	Command	Result
Sequential files	`touch file{1..7}`	Creates file1 through file7
Pattern files	`touch app_{1..3}.txt`	Creates app_1.txt, app_2.txt, app_3.txt

Directory Creation

Need	Command	Result
Single directory	`mkdir aws`	Creates aws folder
Multiple directories	`mkdir dir1 dir2 dir3`	Creates three directories
Sequential directories	`mkdir aws{1..7}`	Creates aws1 through aws7
Nested directories	`mkdir -p parent/child/grandchild`	Creates full path

4. LISTING ITEMS

Task	Command	Output
Simple list	`ls`	Basic file/directory names
Long format	`ls -l`	Detailed list with permissions
All files	`ls -a`	Includes hidden files (starting with .)
Human readable	`ls -lh`	File sizes in KB, MB, GB
Reverse order	`ls -lr`	Reverse alphabetical order
Sort by time	`ls -lt`	Newest files first
Recursive	`ls -R`	List subdirectories recursively
Common alias	`ll`	Typically `ls -l`

5. REMOVING FILES & DIRECTORIES 🗑️

File Removal

Scope	Command	Behavior
Single file	`rm file1`	Asks for confirmation

Scope	Command	Behavior
Multiple files	`rm file1 file2 file3`	Asks for each file
Force delete	`rm -f file1`	No prompts, force removal
Recursive force	`rm -rf dir1`	Remove directory and contents
Range deletion	`rm -rf aws{1..7}`	Removes aws1 through aws7
All files	`rm -rf *`	**Dangerous**: Removes everything in current directory
Pattern deletion	`rm -rf a*`	Removes all files starting with "a"
Extension deletion	`rm -rf *.java`	Removes all .java files

Directory Removal

Situation	Command	Limitations
Empty directory	`rmdir dir1`	Only works on empty directories
Multiple empty dirs	`rmdir dir1 dir2 dir3`	Removes multiple empty directories
Note: `rmdir` provides no confirmation prompts		

6. SAFETY PRECAUTIONS ⚠

Dangerous Commands to Avoid

```
# NEVER RUN THESE UNLESS ABSOLUTELY SURE
rm -rf /          # Deletes entire system
rm -rf *          # Deletes all files in current directory
rm -rf .*         # Deletes all hidden files
```

Safe Alternatives

```
# Always check before deleting
ls before_removing/
rm -ri directory/  # Interactive deletion
```

7. MEMORY HOOKS & TIPS

Command Associations

- `lsblk` = "List Block Devices"
- `lscpu` = "List CPU Information"
- `mkdir -p` = "Make directory with Parents"

Brace Expansion Magic

```
# Create multiple resources quickly
touch document_{a..c}.txt
mkdir server_{01..10}
rm -rf temp_{1..5}
```

Practical Scenarios

```
# Quick workspace setup
mkdir -p project/{src,bin,doc,test}
touch project/src/main_{1..3}.java

# Cleanup specific patterns
rm -rf *.log
rm -rf temp_*
```

8. QUICK REFERENCE CARD

Essential Hardware Commands

```
lscpu      # CPU info
free -h    # Memory in human format
df -h      # Disk space
lsblk      # Block devices
```

Essential File Operations

```
touch new_file.txt          # Create file
mkdir new_directory         # Create directory
ls -la                      # List all with details
rm -rf old_directory        # Remove directory
rm *.tmp                    # Remove all .tmp files
```

🐧 DIRECTORY NAVIGATION, FILE OPERATIONS & EDITORS

1. DIRECTORY NAVIGATION

Action	Command	Description
Show Current Directory	`pwd`	Print Working Directory
Change to Folder	`cd folder1`	Enter specific directory
Go to Root Directory	`cd /`	Navigate to system root
Return to Previous Directory	`cd -`	Switch between current and last directory
Move Up One Level	`cd ..`	Go to parent directory
Move Up Two Levels	`cd ../..`	Go to grandparent directory
Navigate Deep Path	`cd folder1/folder2/folder3`	Direct multi-level navigation
Go to Home Directory	`cd ~` or `cd`	Return to user home directory

2. FILE & DIRECTORY CREATION

Quick Creation Commands

Need	Command	Result
Create File in Specific Path	`touch folder/deep.txt`	Creates file in target folder from anywhere
Preview Folder Contents	`ll folder`	Lists directory contents before entering
Create Nested Directories	`mkdir -p folder1/folder2/folder3`	Creates full path with parent directories
Key Option: `-p` = "create parents if absent"		

Practical Examples

```
# Create project structure
mkdir -p myproject/{src,bin,doc,test,logs}

# Create multiple config files
touch myproject/config/{app,db,web}.config

# Verify structure
ll myproject/
```

3. REMOVAL OPERATIONS ⚠

Dangerous Deletion Commands

Target	Command	Effect & Warning
Everything in Current Directory	`rm -rf *`	**DANGEROUS**: Removes all files and directories recursively
All Subdirectories Only	`rm -rf */`	Removes all directories but leaves files
Current Directory Contents	`rm -rf .`	**EXTREME DANGER**: Removes everything including hidden files

Safety First Practices

```
# Always check before deleting
ls before_removing/

# Use interactive mode for safety
rm -ri directory/

# Test with dry run first (if available)
rsync -av --delete --dry-run source/ empty_dir/
```

4. COPY & MOVE OPERATIONS

Copy Commands

Scenario	Command	Usage
Copy Single File	`cp src.txt dest.txt`	Basic file copy
Copy Multiple Files to Directory	`cp file1 file2 file3 dir/`	Last argument must be destination directory
Recursive Copy	`cp -r source_dir/ dest_dir/`	Copy directories and contents
Preserve Attributes	`cp -a source/ dest/`	Copy with permissions and timestamps

Move/Rename Commands

Scenario	Command	Usage
Move File	`mv old.txt new.txt`	Rename or relocate file
Move Multiple Files	`mv file1 file2 file3 dir/`	Move several files to directory
Bulk Renaming	`mv *.txt text_files/`	Move all .txt files to directory

Practical Examples

```
# Backup configuration
cp -a /etc/myapp/ /backup/myapp_backup/

# Organize files
mkdir -p images/ documents/
mv *.jpg *.png images/
mv *.pdf *.doc documents/

# Rename with pattern
mv old_filename.txt new_filename.txt
```

5. CAT COMMAND TRICKS 🐱

Action	Command	Usage
View File Contents	`cat file1`	Display entire file
Create/Overwrite File	`cat > file1`	Type content, press `Ctrl+D` to save
Append to File	`cat >> file1`	Add content, press `Ctrl+D` to save
View Multiple Files	`cat file1 file2`	Concatenate and display
Number Lines	`cat -n file1`	Show with line numbers
Display Non-printable	`cat -A file1`	Show all characters including endings

Advanced Cat Usage

```
# Create multi-line file
cat > script.sh << EOF
#!/bin/bash
echo "Hello World"
date
EOF

# Combine files
cat file1.txt file2.txt > combined.txt

# View with pagination
cat long_file.txt | less
```

6. VIM EDITOR - ESSENTIALS

Basic Vim Operations

Task	Command/Key	Description
Open File	`vim filename`	Start editing file
Modes	**Command Mode** (default)	For navigation, deletion, search
Save & Exit	`:wq` or `:x`	Write and quit
Save Only	`:w`	Write without exiting
Quit Without Save	`:q!`	Force quit ignoring changes
Navigation	`G`	Jump to last line
	`gg`	Jump to first line
	`:5`	Jump to line 5
Insert Mode	`i`	Insert before cursor
	`a`	Append after cursor
	`o`	Open new line below

Essential Vim Commands

```
# Basic editing workflow
vim file.txt          # Open file
i                     # Enter insert mode
[type your content]   # Edit text
Esc                   # Return to command mode
:wq                   # Save and exit

# Quick navigation
20G                   # Go to line 20
/text                 # Search for "text"
n                     # Next search result
```

7. MEMORY HOOKS & BEST PRACTICES

Quick Mnemonics

- `cd -` = "Dash back to previous directory"
- `mkdir -p` = "Make parent directories"
- `rm -rf` = "Recursive Force - use with extreme caution"
- `cat > file` = "Create file like a cat scratching content"

Safety Protocols

```
# Always verify paths
pwd && ls -la

# Use aliases for safety
alias rm='rm -i'      # Interactive removal
alias cp='cp -i'      # Interactive copy

# Backup before major operations
cp important_file.txt important_file.txt.backup
```

Productivity Tips

```
# Quick directory navigation
cd /var/log && pwd

# Batch operations with brace expansion
touch report_{jan,feb,mar}.txt
mkdir -p project/{2023,2024}/{q1,q2,q3,q4}

# Use history for complex commands
history | grep "cp -r"
```

🐧 VIM POWER MOVES - COMPREHENSIVE CHEAT SHEET

1. FAST NAVIGATION (COMMAND MODE)

Vertical Navigation

Jump To	Key(s)	Description
Top of File	`gg`	First line of file
Bottom of File	`G`	Last line of file
Middle of File	`M`	Middle of visible screen
Specific Line	`7gg` or `7G`	Jump to line 7
Specific Line (Command)	`:18` + `Enter`	Jump to line 18
Relative Line	`10j`	Jump 10 lines down
Relative Line	`5k`	Jump 5 lines up

Horizontal Navigation

Movement	Key(s)	Description
Start of Line	`0`	First character of line
First Non-blank	`^`	First non-whitespace character
End of Line	`$`	Last character of line
Word Forward	`w`	Next word start
Word Backward	`b`	Previous word start
End of Word	`e`	End of current word

Screen Navigation

Action	Key(s)	Description
Half Page Down	`Ctrl + d`	Scroll down half page
Half Page Up	`Ctrl + u`	Scroll up half page
Full Page Down	`Ctrl + f`	Scroll down full page
Full Page Up	`Ctrl + b`	Scroll up full page
Center Screen	`zz`	Center current line
Toggle Line Numbers	`:set number`	Show/hide line numbers
Toggle Relative Numbers	`:set relativenumber`	Show relative line numbers

2. YANK / PASTE / DELETE OPERATIONS

Copy (Yank) Operations

Action	Key(s)	Result
Copy Current Line	`yy`	Yank entire line
Copy 5 Lines	`5yy`	Yank next 5 lines
Copy to End of Line	`y$`	Yank from cursor to line end
Copy Word	`yw`	Yank current word
Copy to Character	`ytx`	Yank until character 'x'

Paste Operations

Action	Key(s)	Result
Paste After Cursor	`p`	Paste after current position
Paste Before Cursor	`P`	Paste before current position
Paste 5 Times	`5p`	Paste clipboard 5 times
Paste from System Clipboard	`"+p`	Paste from system clipboard

Delete Operations

Action	Key(s)	Result
Delete Current Line	`dd`	Cut entire line
Delete 10 Lines	`10dd`	Cut next 10 lines
Delete to End of Line	`d$`	Cut from cursor to line end
Delete Word	`dw`	Cut current word
Delete Character	`x`	Delete character under cursor

Undo/Redo

Action	Key(s)	Description
Undo	`u`	Undo last change
Redo	`Ctrl + r`	Redo undone change
Undo Line	`U`	Undo all changes on current line

3. INSERT MODE SHORTCUTS

Entering Insert Mode

Action	Key(s)	Description
Insert at Cursor	`i`	Insert before cursor
Append After Cursor	`a`	Insert after cursor
Insert at Line Start	`I`	Insert at beginning of line

Action	Key(s)	Description
Append at Line End	`A`	Insert at end of line
New Line Below	`o`	Open new line below current
New Line Above	`O`	Open new line above current
Return to Command Mode	`Esc` or `Ctrl + [`	Exit insert mode

4. SEARCH & REPLACE OPERATIONS

Search Operations

Task	Command	Description
Forward Search	`/word` + `Enter`	Search forward for "word"
Backward Search	`?word` + `Enter`	Search backward for "word"
Next Match	`n`	Go to next search result
Previous Match	`N`	Go to previous search result
Search Current Word	`*`	Search for word under cursor
Case Insensitive Search	`:set ignorecase`	Enable case-insensitive search

Replace Operations

Task	Command	Description
Replace All in File	`:%s/old/new/g`	Replace all "old" with "new"
Replace with Confirmation	`:%s/old/new/gc`	Replace with confirmation prompt
Replace in Current Line	`:s/old/new/g`	Replace in current line only
Replace in Visual Selection	Select text + `:s/old/new/g`	Replace in selected area
Case Insensitive Replace	`:%s/old/new/gi`	Replace ignoring case

5. SAVE & QUIT COMMANDS

File Operations

Purpose	Command	Description
Save File	`:w`	Write changes to file
Save As	`:w newfilename`	Save with different name
Quit	`:q`	Quit Vim
Save and Quit	`:wq` or `:x`	Save changes and exit
Force Quit	`:q!`	Quit without saving
Force Save & Quit	`:wq!`	Force save and quit
Save All & Quit	`:xa`	Save all files and exit

6. TERMINAL INTEGRATION

Paste Operations by Environment

Environment	Paste Method	Description
Linux Terminal	`Ctrl + Shift + V`	Standard Linux paste
PuTTY / Windows	Right-click	Default PuTTY behavior
macOS Terminal	`Cmd + V`	Standard macOS paste
Vim Paste Mode	`:set paste`	Enter paste mode (prevents formatting issues)
Exit Paste Mode	`:set nopaste`	Return to normal mode

Clipboard Integration

Action	Command	Description
Copy to System Clipboard	`"+y`	Yank to system clipboard
Paste from System Clipboard	`"+p`	Paste from system clipboard
Cut to System Clipboard	`"+d`	Delete to system clipboard

7. MEMORY HOOKS & PRO TIPS

Essential Mnemonics

- **"Gg - go, yy - yank, dd - delete, :wq - done!"**

- **"i for Insert, Esc for Escape"**
- **"/ for Find, :%s/ for Replace"**

Productivity Tips

```
# Open file at specific line
vim +25 filename.txt

# Open multiple files
vim file1.txt file2.txt
:n          # Next file
:N          # Previous file

# Split screen editing
:sp file2.txt     # Horizontal split
:vsp file2.txt    # Vertical split
Ctrl + w + w      # Switch between splits
```

Common Workflows

```
# Quick editing session
vim config.txt    # Open file
/text_to_find     # Search for text
i                # Edit text
Esc               # Return to command mode
:wq               # Save and exit

# Bulk operations
:10,20s/old/new/g  # Replace between lines 10-20
:g/pattern/d       # Delete all lines containing pattern
```

🐧 LINUX USER & GROUP MANAGEMENT - COMPREHENSIVE CHEAT SHEET

1. VIEW EXISTING ACCOUNTS

User Information Commands

What You Want	Command	Description & Output
All Local Users	`cat /etc/passwd`	Raw user database file
All Local Users (System)	`getent passwd`	System database query
Specific User Info	`getent passwd username`	Single user details

What You Want	Command	Description & Output
Current User ID	`id`	Current user's UID, GID, and groups
Specific User ID	`id username`	User's UID, GID, and group memberships
Who is Logged In	`who` or `w`	Currently logged in users
Last Logins	`last`	Login history

Group Information Commands

What You Want	Command	Description
All Local Groups	`cat /etc/group`	Raw group database file
All Local Groups (System)	`getent group`	System database query
User's Group Memberships	`groups username`	Groups a user belongs to
Specific Group Info	`getent group groupname`	Single group details

Understanding /etc/passwd Format

```
username:x:UID:GID:Full Name:/home/username:/bin/bash
```

- **Username**: Login name
- **x**: Password placeholder (stored in /etc/shadow)
- **UID**: User ID (0=root, 1-999=system, 1000+=regular)
- **GID**: Primary Group ID
- **Full Name**: GECOS field (user info)
- **Home Directory**: User's home path
- **Shell**: Default shell

2. ADD / DELETE USERS 👦 💻

User Creation Commands

Task	Command	Options & Description
Create Basic User	`useradd username`	Minimal user creation
Create User with Home	`useradd -m username`	Create with home directory

Task	Command	Options & Description
Create with Specific UID	`useradd -u 1500 username`	Assign specific User ID
Create with Specific GID	`useradd -g 1001 username`	Assign specific Primary Group
Create with Comment	`useradd -c "Full Name" username`	Add user description
Create with Specific Shell	`useradd -s /bin/bash username`	Set default shell
Create with Home Path	`useradd -m -d /custom/home username`	Custom home directory

User Deletion Commands

Task	Command	Effect
Remove User Only	`userdel username`	Delete user account only
Remove User + Home	`userdel -r username`	Delete user and home directory
Force Removal	`userdel -f username`	Force delete logged-in user

Password Management

Task	Command	Description
Set User Password	`passwd username`	Set/change password for user
Change Own Password	`passwd`	Change current user's password
Lock User Account	`passwd -l username`	Lock user account
Unlock User Account	`passwd -u username`	Unlock user account
Expire Password	`passwd -e username`	Force password change on next login

Key Facts About User Creation

- **Automatic Primary Group**: Created with same name as user
- **Home Directory**: Typically `/home/username` (with `-m` flag)
- **Default Shell**: Usually `/bin/bash`
- **System Users**: UID 1-999 (varies by distribution)
- **Regular Users**: UID 1000+ (varies by distribution)

3. SWITCH USERS & SESSIONS

User Switching Commands

Task	Command	Description
Switch to User	`su - username`	Login as user with full environment
Switch to User (minimal)	`su username`	Switch without full environment
Become Root	`su -` or `sudo -i`	Switch to root user
Return to Previous	`exit` or `Ctrl + D`	Exit current shell
Run Single Command as User	`sudo -u username command`	Execute command as specific user

Session Management

```
# Multiple user switching example
whoami                    # Check current user
su - john                 # Switch to john
exit                      # Return to original user
sudo -u john whoami       # Run command as john without switching
```

4. GROUP OPERATIONS 👥

Group Management Commands

Need	Command	Description
Create Group	`groupadd groupname`	Create new group
Create System Group	`groupadd -r groupname`	Create system group (GID < 1000)
Create with Specific GID	`groupadd -g 1500 groupname`	Create group with specific GID
Delete Group	`groupdel groupname`	Remove group
Modify Group	`groupmod -n newname oldname`	Rename group

User Group Membership

Need	Command	Description
Add User to Secondary Group	`usermod -aG groupname username`	Append user to group
Set User Primary Group	`usermod -g groupname username`	Change user's primary group
Add Multiple Groups	`usermod -aG group1,group2 username`	Add to multiple groups

Need	Command	Description
Remove from Group	`gpasswd -d username groupname`	Remove user from group
View User Groups	`groups username`	Show user's group membership

Group Password & Administration

Task	Command	Description
Set Group Password	`gpasswd groupname`	Set group password
Add Group Admin	`gpasswd -A username groupname`	Add group administrator
Remove Group Admin	`gpasswd -A "" groupname`	Remove all administrators

5. HOME DIRECTORY MANAGEMENT

Home Directory Operations

Task	Command	Description
List All Home Directories	`ls -la /home/`	View all user homes
Check Specific Home	`ls -la /home/username/`	Check user's home contents
Create Home Directory	`mkhomedir_helper username`	Create home directory for existing user
Set Home Directory	`usermod -d /new/home username`	Change user's home directory
Move Home Directory	`usermod -m -d /new/home username`	Move content to new home

Permission & Security

```
# Check home directory permissions
ls -ld /home/username/

# Typical secure permissions:
# drwx------ username username /home/username
# (Only owner can read/write/execute)

# Fix insecure permissions
chmod 700 /home/username/
chown username:username /home/username/
```

6. PRACTICAL SCENARIOS & EXAMPLES

Create Development Team

```
# Create groups
groupadd developers
groupadd designers

# Create users with home directories
useradd -m -c "John Developer" -G developers johndev
useradd -m -c "Jane Designer" -G designers janedesign

# Set passwords
passwd johndev
passwd janedesign

# Add user to multiple groups
usermod -aG developers,designers adminuser
```

User Account Maintenance

```
# Check account status
chage -l username    # Password aging info
last username        # Login history

# Lock and unlock accounts
usermod -L username   # Lock account
usermod -U username   # Unlock account

# Set account expiration
usermod -e 2024-12-31 username   # Expire account on date
```

Batch User Operations

```
# Create multiple users
for user in alice bob charlie; do
    useradd -m $user
    echo "$user:password123" | chpasswd
done

# Delete multiple users
for user in alice bob charlie; do
    userdel -r $user
done
```

7. SECURITY BEST PRACTICES

Account Security

- Use `passwd -l` to lock inactive accounts
- Regularly review `/etc/passwd` for unauthorized users
- Monitor `/var/log/secure` for authentication attempts

- Use `chage` to enforce password policies

Permission Guidelines

- Home directories should be 700 (drwx------)
- Users should not have write access to other home directories
- Use groups for shared resource access instead of loose permissions