**UNIX COMMANDS**

Unix is a powerful operating system that uses a variety of commands to perform tasks. Below is a list of common Unix commands with detailed explanations:

**1. ls**

* **Purpose**: Lists directory contents.
* **Usage**: ls [options] [path]
* **Example**: ls -l /home/user
  + Lists files in the /home/user directory in long format.
* **Options**:
  + l: Long listing format (shows permissions, owner, size, and timestamp).
  + a: Includes hidden files (those starting with a dot . ).
  + h: Human-readable sizes (e.g., 1K, 234M).

**2. cd**

* **Purpose**: Changes the current working directory.
* **Usage**: cd [directory]
* **Example**: cd /home/user/Documents
  + Moves to the /home/user/Documents directory.
* **Options**:
  + cd ..: Moves up one directory level.
  + cd ~: Moves to the user's home directory.

**3. pwd**

* **Purpose**: Prints the current working directory.
* **Usage**: pwd
* **Example**: pwd
  + Outputs the full path of the current directory.

**4. cp**

* **Purpose**: Copies files or directories.
* **Usage**: cp [source] [destination]
* **Example**: cp file1.txt /home/user/backup/
  + Copies file1.txt to the /home/user/backup/ directory.
* **Options**:
  + r: Recursively copy directories.
  + i: Prompts before overwriting files.

**5. mv**

* **Purpose**: Moves or renames files and directories.
* **Usage**: mv [source] [destination]
* **Example**: mv file1.txt file2.txt
  + Renames file1.txt to file2.txt.
* **Options**:
  + i: Prompts before overwriting.

**6. rm**

* **Purpose**: Removes files or directories.
* **Usage**: rm [options] [file]
* **Example**: rm file1.txt
  + Removes the file file1.txt.
* **Options**:
  + r: Recursively remove directories.
  + f: Force removal without confirmation.
  + i: Prompt before each deletion.

**7. mkdir**

* **Purpose**: Creates a new directory.
* **Usage**: mkdir [directory\_name]
* **Example**: mkdir new\_folder
  + Creates a directory named new\_folder.
* **Options**:
  + p: Create parent directories as needed.

**8. rmdir**

* **Purpose**: Removes empty directories.
* **Usage**: rmdir [d irectory]
* **Example**: rmdir old\_folder
  + Removes the empty directory old\_folder.

**9. touch**

* **Purpose**: Creates an empty file or updates the timestamp of an existing file.
* **Usage**: touch [file]
* **Example**: touch newfile.txt
  + Creates an empty file newfile.txt.

**10. cat**

* **Purpose**: Concatenates and displays the content of files.
* **Usage**: cat [file]
* **Example**: cat file1.txt
  + Displays the contents of file1.txt to the terminal.

**Combine Multiple Files Into One**

* **Purpose**: You can combine the contents of multiple files into a new file using redirection.
* **Usage**: cat file1.txt file2.txt > combined.txt
* **Example**:
* cat file1.txt file2.txt > combined.txt
  + This combines file1.txt and file2.txt into a new file combined.txt.

**Summary of Common Options:**

* **n**: Numbers all lines.
* **b**: Numbers only non-empty lines (useful for excluding blank lines from numbering).
* **s**: Squeezes blank lines (removes consecutive empty lines).
* **E**: Shows $ at the end of each line.
* **T**: Shows tabs as ^I.
* **v**: Displays non-printing characters.

**11. more**

* **Purpose**: Views content of a file, one screen at a time.
* **Usage**: more [file]
* **Example**: more file1.txt
  + Displays file1.txt content one page at a time.

**Basic commands while using more:**

* **Space**: Move to the next page.
* **Enter**: Move down one line.
* **q**: Quit and exit.

**12. less**

* **Purpose**: Similar to more, but with more navigation features.
* **Usage**: less [file]
* **Example**: less file1.txt
  + Allows scrolling through file1.txt.

**Basic commands while using less:**

* **Space**: Move to the next page.
* **b**: Move backward to the previous page.
* **Up Arrow / Down Arrow**: Move one line up or down.
* **q**: Quit and exit.
* **/search\_term**: Search for a term in the file.
* **n**: Move to the next search result.
* **N**: Move to the previous search result.

**13. head**

* **Purpose**: Displays the beginning of a file.
* **Usage**: head [options] [file]
* **Example**: head -n 10 file1.txt
  + Displays the first 10 lines of file1.txt.

**14. tail**

* **Purpose**: Displays the end of a file.
* **Usage**: tail [options] [file]
* **Example**: tail -n 10 file1.txt
  + Displays the last 10 lines of file1.txt.
* **Options**:
  + f: Continuously outputs new content as the file grows.

**15. grep**

* **Purpose**: Searches for patterns in files.
* **Usage**: grep [pattern] [file]
* **Example**: grep 'hello' file1.txt
  + Searches for the word "hello" in file1.txt.

**Summary of Common grep Options:**

| **Option** | **Description** |
| --- | --- |
| -i | Case-insensitive search. |
| -n | Show line numbers with matching lines. |
| -v | Invert match (show non-matching lines). |
| -w | Match whole words only. |
| -o | Show only matched parts of the line. |
| -c | Count the number of matching lines. |
| -r or -R | Search recursively through directories. |
| -l | Show only filenames containing the pattern. |
| -L | Show only filenames not containing the pattern. |
| -H | Show filenames in output. |
| -E | Use extended regular expressions (for advanced patterns). |

**16. find**

* **Purpose**: Searches for files and directories in a directory hierarchy.
* **Usage**: find [path] [options]
* **Example**: find /home/user/ -name "\*.txt"
  + Finds all .txt files in /home/user/ and subdirectories.

**17. chmod**

* **Purpose**: Changes file permissions.
* **Usage**: chmod [permissions] [file]
* **Example**: chmod 755 file1.txt
  + Sets the file file1.txt to be readable, writable, and executable by the owner, and readable and executable by others.

The sum of the digits represents the permissions for user, group, and others.

| **Permission** | **Number** |
| --- | --- |
| Read (r) | 4 |
| Write (w) | 2 |
| Execute (x) | 1 |
| No permission (-) | 0 |

For example:

* **7** = 4 (read) + 2 (write) + 1 (execute) = **rwx**
* **6** = 4 (read) + 2 (write) = **rw-**
* **5** = 4 (read) + 1 (execute) = **r-x**
* **4** = 4 (read) = **r--**
* **0** = No permissions = **--**

**18. chown**

* **Purpose**: Changes file ownership.
* **Usage**: chown [owner] [file]
* **Example**: chown user:group file1.txt
  + Changes ownership of file1.txt to user and group to group.

**chown (Change Owner)**

The **chown** command is used to **change the owner** of a file or directory. It can also change the **group** associated with a file or directory.

**Basic Syntax:**

chown [OPTIONS] OWNER[:GROUP] FILE

* **OWNER**: The new owner of the file or directory.
* **GROUP**: The new group that will be assigned to the file or directory (optional).
* **FILE**: The file or directory for which the ownership is to be changed.

**Key Options:**

* **R**: Recursively change ownership of directories and their contents.
* **v**: Verbose mode. It shows which files are being changed.

**Examples:**

1. **Change the owner of a file**:
2. chown alice file.txt
   * This changes the owner of file.txt to alice, while leaving the group unchanged.
3. **Change both the owner and the group**:
4. chown alice:developers file.txt
   * This changes both the owner of file.txt to alice and the group to developers.
5. **Recursively change ownership of a directory and its contents**:
6. chown -R alice:developers /home/alice/project
   * This changes the ownership of the project directory and all files within it to alice (owner) and developers (group).
7. **Check current ownership**: You can check the current owner and group of a file using the ls -l command:
8. ls -l file.txt

**19. ps**

* **Purpose**: Displays information about active processes.
* **Usage**: ps [options]
* **Example**: ps aux
  + Lists all running processes with detailed information.

**20. kill**

* **Purpose**: Sends a signal to a process, often used to terminate it.
* **Usage**: kill [signal] [pid]
* **Example**: kill 1234
  + Terminates the process with PID 1234.

1. **Terminate a process by PID**:
2. kill 1234
   * This sends the **default signal (SIGTERM)** to the process with PID 1234, which requests it to terminate gracefully.
3. **Terminate a process with a specific signal**: You can use **different signals** to control a process. The most common signals are:
   * **SIGTERM (15)**: The default signal that asks a process to terminate gracefully.
   * **SIGKILL (9)**: Forces the process to terminate immediately, without giving it a chance to clean up resources.
4. kill -9 1234
   * This sends **SIGKILL** to the process with PID 1234, which will forcefully terminate the process.
5. **Send a signal to a process by name**: You can use the pkill or killall commands to send signals to processes by name rather than by PID.

Example with pkill:

pkill firefox

* + This sends the **default signal (SIGTERM)** to all processes named firefox, attempting to close them.

1. **Send a signal to all processes of a user**:
2. kill -u username
   * This will send the default signal (SIGTERM) to all processes owned by the user username.

**Key Signals:**

The **kill** command can send many different signals, and you can specify these with either the signal name (like SIGKILL) or the signal number (like 9 for SIGKILL).

**Common signals:**

* **SIGTERM (15)**: Gracefully terminates a process (default signal).
* **SIGKILL (9)**: Forces the termination of a process, and it cannot be ignored or handled by the process.
* **SIGSTOP (19)**: Pauses (stops) the process. It can be resumed later using SIGCONT.
* **SIGCONT (18)**: Resumes a paused (stopped) process.
* **SIGINT (2)**: Interrupts the process, usually generated by pressing Ctrl + C.
* **SIGHUP (1)**: Hangs up a process, often used to restart services (like a daemon process).
* **SIGQUIT (3)**: Causes the process to terminate and create a core dump (used for debugging).

**21. top**

* **Purpose**: Displays real-time system information including CPU, memory usage, and running processes.
* **Usage**: top
* **Example**: top
  + Displays a dynamic, real-time view of system processes.

**22. df**

* **Purpose**: Displays disk space usage.
* **Usage**: df [options]
* **Example**: df -h
  + Shows disk space usage in human-readable format.

**23. du**

* **Purpose**: Estimates file space usage.
* **Usage**: du [options] [file]
* **Example**: du -sh /home/user
  + Displays the size of the /home/user directory.

**24. tar**

* **Purpose**: Archives files and directories into a single file.
* **Usage**: tar [options] [archive\_name] [files]
* **Example**: tar -cvf archive.tar file1.txt
  + Creates a tar archive of file1.txt named archive.tar.
* **Options**:
  + c: Create a new archive.
  + v: Verbose output.
  + f: Specify the archive file name.
  + x: Extract an archive.

**25. wget**

* **Purpose**: Downloads files from the web.
* **Usage**: wget [URL]
* **Example**: wget <https://example.com/file.zip>
  + Downloads the file from the specified URL.

**26. curl**

* **Purpose**: Transfers data to or from a server.
* **Usage**: curl [options] [URL]
* **Example**: curl -O <https://example.com/file.zip>
  + Downloads the file from the specified URL.

**Commonly Used Options:**

* **O**: Saves the downloaded file with the same name as on the server.
* curl -O <https://example.com/file.zip>
* **o filename**: Saves the downloaded content to a specific file.
* curl -o myfile.zip <https://example.com/file.zip>
* **L**: Follow redirects (if the URL redirects to another location).
* curl -L <https://example.com/file.zip>
* **u username:password**: Pass authentication credentials (basic auth).
* curl -u username:password <https://example.com/file.zip>
* **I**: Fetch the HTTP headers only (useful for debugging).
* curl -I <https://example.com/>
* **d**: Sends data in a POST request (commonly used with APIs).
* curl -d "key1=value1&key2=value2" -X POST <https://example.com/api>
* **H**: Send custom HTTP headers (useful for APIs).
* curl -H "Authorization: Bearer token" <https://example.com/api>

**27. man**

* **Purpose**: Displays the manual page for a command.
* **Usage**: man [command]
* **Example**: man ls
  + Displays the manual page for the ls command.

**28. alias**

* **Purpose**: Creates a shortcut for a command.
* **Usage**: alias [alias\_name]='[command]'
* **Example**: alias ll='ls -l'
  + Creates a shortcut ll for ls -l.

**29. uname**

* **Purpose**: Displays system information.
* **Usage**: uname [options]
* **Example**: uname -a
  + Displays all system information.

**30. history**

* **Purpose**: Displays the command history.
* **Usage**: history
* **Example**: history
  + Shows the list of recently executed commands.

This list includes the most commonly used Unix commands and options. There are many more advanced commands and variations, but these should give you a solid foundation for using Unix-based systems.