

Understanding Linux Command Structure

Various Forms:

1. → **command**
 2. → **command** **[options]**
 3. → **command** **[arguments]**
 4. → **command** **[options]** **[argument]**
or
→ **command** **[-flag(s) [-option(s) [value]] [argument]**
-

command

This is the name of the command or utility we want to run. They are executable file or program. In Linux, a command is an instruction given by a user to the operating system's shell to perform a specific task. We can locate them using where utility.

Try: → **where pwd**

Examples:

1. **cd**
2. **pwd**
3. **ls**
4. **ps**

Various Types of Utilities:

1. Text Processing Utilities

- **cat**: Concatenates and displays file contents.
- **tac**: Displays file contents in reverse order.
- **nl**: Adds line numbers to file contents.
- **grep**: Searches for patterns in text.
- **awk**: A programming language for text processing.
- **sed**: Stream editor for editing text on the fly.
- **cut**: Extracts sections of a file.
- **sort**: Sorts text lines.
- **uniq**: Removes duplicate lines.
- **tr**: Translates or deletes characters.
- **wc**: Counts words, lines, and bytes in text.

2. Process Management Utilities

- **ps**: Displays information about running processes.
- **top**: Monitors running processes in real time.
- **htop**: Interactive process viewer (alternative to **top**).
- **kill**: Sends signals to terminate or control processes.
- **pkill**: Terminates processes by name.
- **jobs**: Lists active jobs in the current session.
- **bg**: Resumes a job in the background.
- **fg**: Brings a background job to the foreground.
- **nice**: Sets process priority.
- **renice**: Changes the priority of a running process.

3. File Handling Utilities

- **ls**: Lists directory contents.
- **cp**: Copies files and directories.
- **mv**: Moves or renames files and directories.
- **rm**: Deletes files or directories.
- **touch**: Creates empty files.
- **stat**: Displays detailed file information.
- **chmod**: Changes file permissions.
- **chown**: Changes file ownership.
- **ln**: Creates hard or symbolic links.

4. Network Utilities

- **ping**: Checks network connectivity to a host.
- **wget**: Downloads files from the web.
- **curl**: Transfers data from or to a server.
- **netstat**: Displays network connections (deprecated; use **ss**).
- **ss**: Displays network socket statistics.
- **ifconfig**: Configures network interfaces (deprecated; use **ip**).
- **ip**: Manages IP addresses and routes.
- **nslookup**: Queries DNS for domain information.
- **traceroute**: Traces the route packets take to a host.
- **ftp**: Transfers files between systems using the FTP protocol.

And so on...

These Utilities can be of **Admin Level**, that requires root access or membership to sudo group. Be careful they can affect the entire system and other users. Or **User Level**, available for all users without special permission. They affect only user's environment.

[-flag(s) [-option(s) [value]]]

[-flag(s)]:

Flags are single-character options that modify the behaviour of the command. They are usually prefixed with a single dash '-'. Multiple flags can often be combined after a single dash.

Examples:

1. `ls -F`

2. `ls -a`

3. `ps -l`

We can feed multiple flags like this:

4. `ls -la` or `ls -l -a`

[-option(s) [value]]:

Single character Options are prefixed with single dash '-'. They are flags.

Multiple character Options are prefixed with double dash '--'.

Some options can even have associated values.

Examples:

single character options:

1. `head -n 5 file.txt`

2. `grep -e "pattern" file.txt`

multiple character options:

3. `ps --forest`

4. `curl --output output.html https://example.com`

note: The associated value with an option specifies what we are setting or modifying with the option. Options alters the default behaviour of the command.

[argument]

[argument]:

Arguments are the targets or parameters on which the command acts. These can be filenames, directories or other data inputs.

Examples:

1. `cp source/source destination/file`

2. `mv oldname newname`

3. `touch file.txt`

4. `man ls`

Example 1

Command without options and arguments

Some commands can be executed without any additional option or argument. They perform specific function with a predefined behaviour.

→ **pwd**

description: Prints the current working directory

```
amit@archlinux > ~/Test > pwd
/home/amit/Test
```

Example 2

Command With Option

Command with multiple character option

→ **ls --version**

description: Show version related information. Double '-' specifies multiple character option.

```
amit@archlinux > ~/Test > ls --version
ls (GNU coreutils) 9.5
Copyright (C) 2024 Free Software Foundation, Inc.
```

Command with single character option

→ **ls**

description: By default ls lists the content of current working directory.

```
amit@archlinux > ~/Test > ls
dir  example.log  file.json  file.txt
amit@archlinux > ~/Test > ls -l
```

→ **ls -l**

description: List the contents in long listing format.

```
amit@archlinux > ~/Test > ls -l
total 16
drwxr-xr-x 2 amit users 4096 Dec 12 21:10 dir
-rw-r--r-- 1 amit users 1403 Dec 12 20:52 example.log
-rw-r--r-- 1 amit users  531 Dec 12 20:49 file.json
-rw-r--r-- 1 amit users   20 Dec 12 20:57 file.txt
```

→ **ls -a**

description: List the contents including hidden contents.

```
amit@archlinux > ~/Test > ls -a
.  ..  dir  example.log  file.json  file.txt  .~lock.cat-vs-bat.pdf#
```

Example 3

Command With Multiple Option

→ **ls -al**

description: List the contents including hidden contents in long listing format.

```
amit@archlinux > ~/Test > ls -la
total 28
drwxr-xr-x  3 amit users 4096 Dec 14 19:39 .
drwx----- 35 amit users 4096 Dec 14 20:18 ..
drwxr-xr-x  2 amit users 4096 Dec 12 21:10 dir
-rw-r--r--  1 amit users 1403 Dec 12 20:52 example.log
-rw-r--r--  1 amit users  531 Dec 12 20:49 file.json
-rw-r--r--  1 amit users   20 Dec 12 20:57 file.txt
-rw-r--r--  1 amit users   73 Dec  9 18:43 .~lock.cat-vs-bat.pdf#
```

Example 4

Command With Argument

→ **mkdir new_directory**

description: Create new directory named new_directory

```
amit@archlinux > ~/Test > ls
dir example.log file.json file.txt
amit@archlinux > ~/Test > mkdir new_directory
amit@archlinux > ~/Test > ls
dir example.log file.json file.txt new_directory
```

Example 5

Command With Multiple Argument

→ **mv old_directory new_directory**

description: change the name of old_directory to new_directory.

```
amit@archlinux > ~/Test > ls
dir example.log file.json file.txt old_directory
amit@archlinux > ~/Test > mv old_directory new_directory
amit@archlinux > ~/Test > ls
dir example.log file.json file.txt new_directory
```

Example 6

Command With Option and Argument

→ **rm -r new_directory**

```
amit@archlinux > ~/Test > ls
dir example.log file.json file.txt new_directory
amit@archlinux > ~/Test > rm -r new_directory
amit@archlinux > ~/Test > ls
dir example.log file.json file.txt
```

description: recursively removes the contents of non empty directory provided as argument including itself.

Example 7

Command With Multiple Option and Multiple Argument

→ **cp -r -v original_directory copy_directory**

description: recursively copy the original_directory to copy_directory with verbose output.

```
amit@archlinux > ~/Test > ls
example.log file.json file.txt original_directory
amit@archlinux > ~/Test > cp -r -v original_directory copy_directory
'original_directory' -> 'copy_directory'
'original_directory/dir.txt' -> 'copy_directory/dir.txt'
'original_directory/rid.txt' -> 'copy_directory/rid.txt'
amit@archlinux > ~/Test > ls
copy_directory example.log file.json file.txt original_directory
```

Example 8

Command With Option That Takes Value

→ **head -n 5 file.json**

description: Print top 5 lines of file.json file.

```
amit@archlinux > ~/Test > head -n 5 file.json
{
  "employees": [
    {
      "id": 1,
      "name": "John Doe",
```

→ **grep --color=always "name" file.json**

description: print the all lines of file.json that contains "name" pattern, highlighting it.

```
amit@archlinux > ~/Test > grep --color=always "name" file.json
  "name": "John Doe",
  "name": "Jane Smith",
  "name": "Alice Johnson",
  "name": "Bob Brown",
```

Note:

I appreciate your openness to feedback and corrections. If you notice any errors or have suggestions for improvement regarding the content, please feel free to provide them here

amit101giri@gmail.com. Your input is valuable, and I'm here to assist you with any questions or clarifications you may need.

Thank You