

# Understanding chmod Command in Unix Operating Systems

The `chmod` command is an essential tool that allows us to control file and directory permissions in Unix. In this presentation, we will learn the syntax, different modes of specifying permissions, and frequently asked questions about this command.



<b>on</b>			<b>Multiple Windows</b>	
to move over	backward	forward	When two commands are shown, the second is a similar command frame instead of a window.	
ter	C-b	C-f		
	M-b	M-f	delete all other windows	C-x 1 C-x 1
line beginning (or end)	C-p	C-n	split window, above and below	C-x 2 C-x 2
ce	C-a	C-e	delete this window	C-x 0 C-x 0
aph	M-a	M-e	split window, side by side	C-x 3
	M-{	M-}	scroll other window	C-M-v
	C-x [	C-x ]	switch cursor to another window	C-x o C-x o
on	C-M-b	C-M-f	select buffer in other window	C-x 4 b C-x 5 b
buffer beginning (or end)	C-M-a	C-M-e	display buffer in other window	C-x 4 C-o C-x 5 C-o
	M-<	M->	find file in other window	C-x 4 f C-x 5 f
to next screen	C-v		find file read-only in other window	C-x 4 r C-x 5 r
to previous screen	M-v		run Dired in other window	C-x 4 d C-x 5 d
left	C-x <		find tag in other window	C-x 4 . C-x 5 .
right	C-x >		grow window taller	C-x ^
current line to center of screen	C-u C-l		shrink window narrower	C-x {
			grow window wider	C-x }
<b>ng and Deleting</b>			<b>Formatting</b>	
to kill	backward	forward	indent current line (mode-dependent)	TAB
ter (delete, not kill)	DEL	C-d	indent region (mode-dependent)	C-M-\
o end of)	M-DEL	M-d	indent sexp (mode-dependent)	C-M-q
ce	M-0 C-k	C-k	indent region rigidly arg columns	C-x TAB
	C-x DEL	M-k	insert newline after point	C-o
	M-- C-M-k	C-M-k	move rest of line vertically down	C-M-o
gion	C-w		delete blank lines around point	C-x C-o
egion to kill ring	M-w		join line with previous (with arg, next)	M-^
rough next occurrence of char	M-z char		delete all white space around point	M-\
ack last thing killed	C-y		put exactly one space at point	M-SPC
e last yank with previous kill	M-y		fill paragraph	M-q
<b>ding</b>			set fill column to arg	C-x f
rk here	C-@ or C-SPC		set prefix each line starts with	C-x .
ge point and mark	C-x C-x		set face	M-o
rk arg words away	M-@		<b>Case Change</b>	
paragraph	M-h		uppercase word	M-u
page	C-x C-p		lowercase word	M-l
sexp	C-M-@		capitalize word	M-c
function	C-M-h		uppercase region	C-x C-u
entire buffer	C-x h		lowercase region	C-x C-l
<b>y Replace</b>			<b>The Minibuffer</b>	
actively replace a text string	M-%		The following keys are defined in the minibuffer.	
regular expressions	M-x query-replace-regexp		complete as much as possible	TAB
responses in query-replace mode are			complete up to one word	SPC
o this one, go on to next	SPC		complete and execute	RET
e this one, don't move	,		show possible completions	?
o next without replacing	DEL		fetch previous minibuffer input	M-p
e all remaining matches	!		fetch later minibuffer input or default	M-n
up to the previous match	~		regex search backward through history	M-r
query-replace	RET		regex search forward through history	M-s
recursive edit ( C-M-c to exit)	C-r		abort command	C-g
<b>ling Check</b>			Type C-x ESC ESC to edit and repeat the last command that was in the minibuffer. Type F10 to activate the menu bar items on text.	
spelling of current word	M-\$		<b>Tags</b>	
spelling of all words in region	M-x ispell-region		find a tag (a definition)	M-.
spelling of entire buffer	M-x ispell-buffer		find next occurrence of tag	C-u M-.
<b>ers</b>			specify a new tags file	M-x visit-tags-
another buffer	C-x b		regex search on all files in tags table	M-x tags-search
uffers	C-x C-b		run query-replace on all the files	M-x tags-query-
uffer	C-x k		continue last tags search or query-replace	M-.,

# Chmod Syntax and Options

1

## Syntax

`chmod [options] [mode] [File_name]`

## Options

2

- **-R** – Apply the permission change recursively to all files and directories within the specified directory.
- **-v** – Display a message for each file that is processed, indicating the permission change that was made.
- **-c** – Display messages only for files whose permission is changed.
- **-f** – Avoid displaying error messages.
- **-h** – Change the permissions of symbolic links instead of the files they point to.

3

## Conclusion

The ``chmod`` command in Linux is a very essential tool for managing file and directory permissions. It allows you to modify the permissions and access mode of files and directories, controlling who can read, write, and execute them. Understanding and effectively using the ``chmod`` command is crucial for managing permissions in Linux.

# Symbolic Mode

## What is it?

The most common way to specify permission. Uses letters and operators to set or modify permissions.

## Operators

- **+** - Add permissions
- **-** - Remove permissions
- **=** - Set permissions

## Letters

- **r** - Read permission
- **w** - Write permission
- **x** - Execute permission

# Octal Mode

- 4 - Read permission
- 2 - Write permission
- 1 - Execute permission

Values

1

What is it?

Another way to specify permissions.  
Uses a three-digit number to represent permissions.

2

# Viewing Permissions

```
ls -lah
```

```
iplant-everyone 4.0K Sep 23 22:40 .  
root            4.0K Sep 15 09:48 ..  
iplant-everyone 4.0K Sep 15 11:19 apcb  
root            2.2K Sep 15 10:49 .bas  
iplant-everyone 61 Sep 16 19:46 .bas  
iplant-everyone 220 Apr 3 2012 .bas  
iplant-everyone 3.6K Sep 15 09:48 .bas  
iplant-everyone 4.0K Sep 15 09:52 .cac
```

## Command

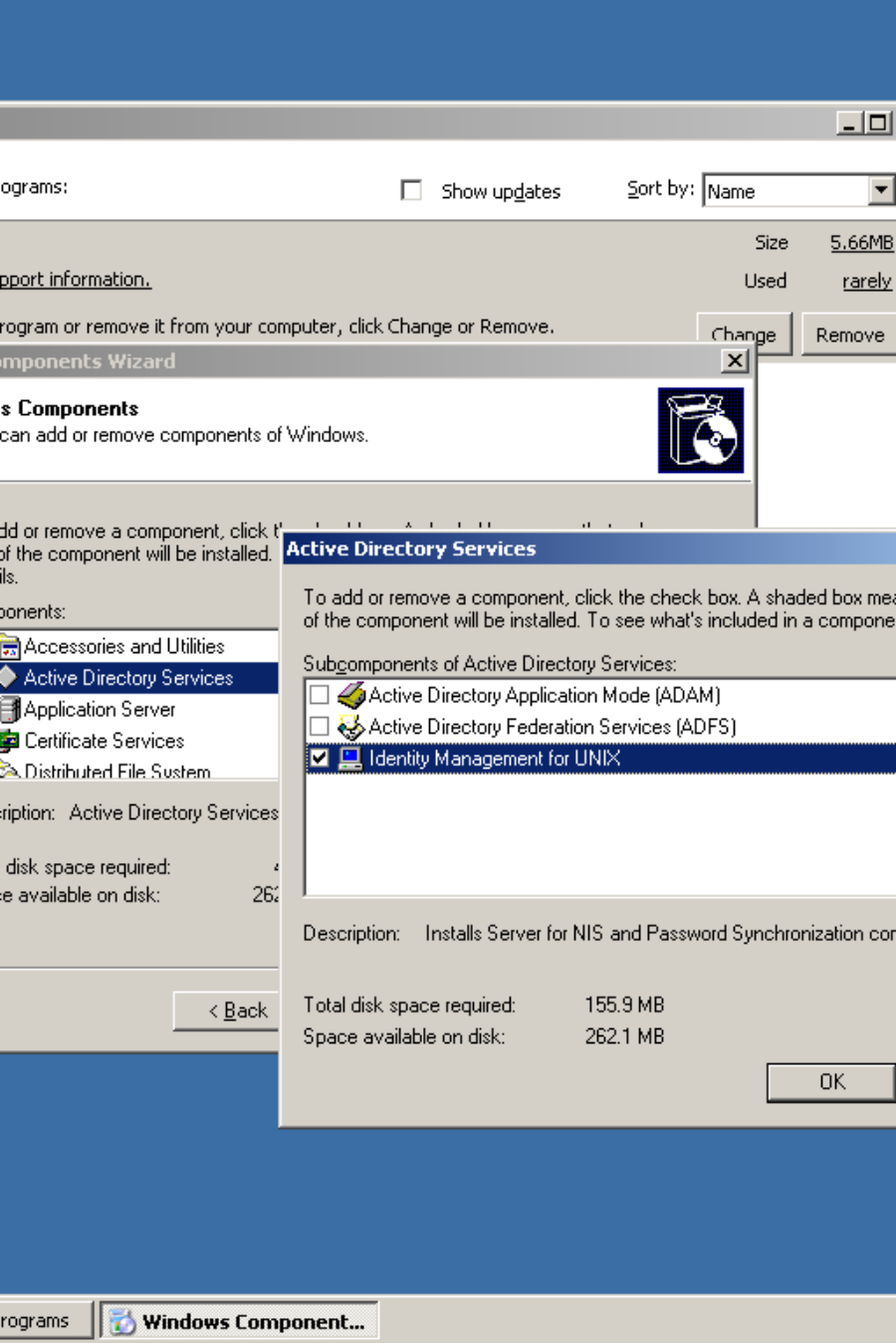
Use the command `ls -l` to view current permissions of a file or directory.

```
ls -lah
```

```
iplant-everyone 4.0K Sep 23 22:40 .  
root            4.0K Sep 15 09:48 ..  
iplant-everyone 4.0K Sep 15 11:19 apcb  
root            2.2K Sep 15 10:49 .bas  
iplant-everyone 61 Sep 16 19:46 .bas  
iplant-everyone 220 Apr 3 2012 .bas  
iplant-everyone 3.6K Sep 15 09:48 .bas  
iplant-everyone 4.0K Sep 15 09:52 .cac
```

## Specific File/Directories

Use the command `ls -l [file_name]` to see all the permissions that a particular directory or file has.



# Frequently Asked Questions

## 1 How Do I View File Permissions?

Use the command `ls -l` to view the current permissions of a file or directory.

## 2 What Are the Different Types of Permissions?

Read (r), Write (w), and Execute (x) permissions are applied to the owner (u), group (g), and others (o).

## 3 How Can I Revert Changes Made by `chmod`?

Use the `chmod` command again with the correct permission. For example, to revert the changes to "rw-r--r--," use the command `chmod 644 [file_or_directory_name]`.

# Changing Permissions for Multiple Files

## What is it?

Allows changing permissions for multiple files.

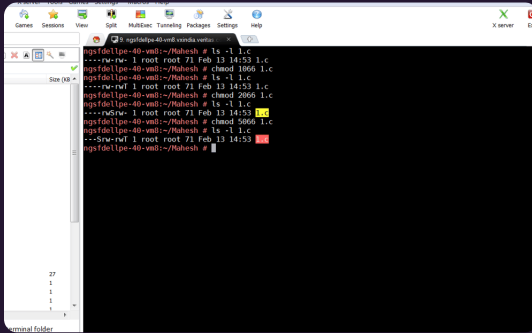
## Recursive Changes

Use the `-R` option to apply the permission change recursively to all the files and directories within the specified directory.

## Messages Output

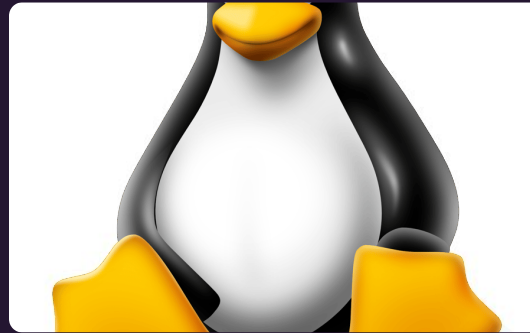
Use `-v` option to view messages for each file modified and `-c` to view messages only for the files whose permission has changed.

# Commonly Used Modes of Symbolic Mode



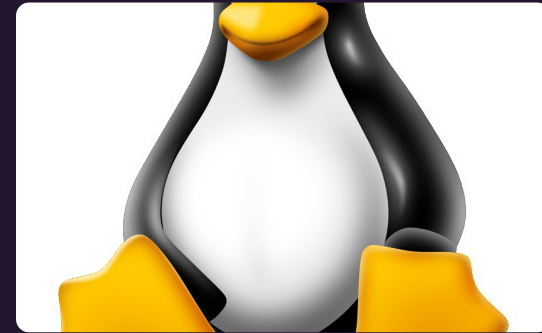
**rwX**

**rwX** represents all permissions.



**+**

**+** adds the permission to the file/directory.



**-**

**-** removes the permission from the file/directory.



# Octal Mode Reference Table

Permission	Value
Read	4
Write	2
Execute	1

# Disk Utilities

Managing disk space and maintaining data integrity are important tasks for any computer user. Disk utilities are software tools designed to help you perform these tasks effectively. Here are some common disk utilities:

- **Disk Cleanup:** This tool helps you free up disk space by deleting temporary files, log files, and other unnecessary files that accumulate over time.
- **Disk Defragmenter:** This tool helps you optimize disk performance by reorganizing files and folders on your hard drive so that they are stored in contiguous clusters.
- **Disk Repair:** This tool helps you detect and repair file system errors, bad sectors, and other issues that can cause data loss or corruption.
- **Disk Cloning:** This tool helps you create a backup copy of your entire hard drive or a selected partition. This can be useful for disaster recovery, system migration, or upgrading to a larger hard drive.
- **Disk Encryption:** This tool helps you protect your sensitive data by encrypting it with a password or key. This can be useful for laptops or USB drives that may be lost or stolen.

Using these disk utilities regularly can help you keep your computer running smoothly and avoid data loss or corruption. However, it's important to use them carefully and understand their limitations.

## df command

The **df** command stands for "disk-free," and shows available and used disk space on the Linux system. Here are some useful options:

- **df -h:** Shows disk space in human-readable format.
- **df -a:** Shows the file system's complete disk usage even if the Available field is 0.
- **df -T:** Shows the disk usage along with each block's filesystem type (e.g., xfs, ext2, ext3, btrfs, etc.).
- **df -i:** Shows used and free inodes.

You can use the **df** command in the terminal to quickly check disk usage on your Linux system. For example, **df -h** will display the disk usage in a human-readable format, making it easy to understand and manage your disk space.

You can also get this information in a graphical view using the Disks (gnome-disk-utility) in the GNOME desktop. Launch it to see all disks detected by your computer, and click a partition to see details about it, including space used and space remaining.

# Linux du command

The `du` command shows the disk usage of files, folders, and other objects in the default kilobyte size. Here are some useful options:

- `du -h`: Shows disk usage in human-readable format for all directories and subdirectories.
- `du -a`: Shows disk usage for all files.
- `du -s`: Provides total disk space used by a particular file or directory.

You can use the `du` command in the terminal to quickly check disk usage of files and folders on your Linux system. For example, `du -h` will display the disk usage of all directories and subdirectories in a human-readable format, making it easy to understand and manage your disk space.

Using the `du` command can be helpful when you need to identify which files and folders are taking up the most space on your system. By using options like `du -h` or `du -s`, you can quickly get an overview of your disk usage and take steps to free up space if needed. ### Other cards in the deck - Disk Utilities - df command