

Managing File Permissions with chmod Command

The `chmod` command is an essential tool used in managing file and directory permissions in Linux and Unix-like operating systems. It allows users to specify who has read, write, and execute permission to a file or folder, which is crucial in upholding system security.

5 VIVEK
IVEK

5 VIVEK

Changing Permissions using Numeric Mode (Octal)

The numeric mode uses three octal digits to represent permission settings—user, group, and others. Each digit sets respective permission levels, where 0 represents no permission and 7 represents read, write, and execute permission. For example, `chmod 755 myfile.txt` grants the user read, write, and execute access while the group and others only have read and execute permission.

```
-zsh
wn Guide % ls -l
e staff 1073 Oct 26 2017 208x128.png
e staff 416 Jul 2 2020 cover
e staff 992 Sep 28 2018 dillinger
e staff 2834 Mar 2 2017 draft-content
e staff 1184 Jun 28 16:55 markdown-guid
e staff 480 Nov 1 2019 markdown-guid
e staff 480 Jun 24 2020 markdown-guid
e staff 128 Oct 3 2019 markdown-test
e staff 4940 Oct 26 2017 newicon.png
e staff 576 Sep 28 2018 old-markdown-
e staff 576 Sep 28 2018 really-old-ma
e staff 160 Oct 8 2019 test-gh-pages
e staff 96 Jan 10 2020 test-mkdocs
wn Guide % chmod 755 markdown-guide
```

```
Help
Tunneling Packages Settings Help
40-vm8.vxindia.veritas.c... X
B:~/Mahesh # ls -l 1.c
t root 71 Feb 13 14:53 1.c
B:~/Mahesh # chmod 1066 1.c
B:~/Mahesh # ls -l 1.c
t root 71 Feb 13 14:53 1.c
B:~/Mahesh # chmod 2066 1.c
B:~/Mahesh # ls -l 1.c
t root 71 Feb 13 14:53 1.c
B:~/Mahesh # chmod 5066 1.c
B:~/Mahesh # ls -l 1.c
t root 71 Feb 13 14:53 1.c
B:~/Mahesh #
```

Changing Permissions using Symbolic Mode

The symbolic mode of `chmod` uses letters and symbols to represent permission settings: u (user), g (group), o (others), a (all), + (add permission), - (remove permission), and = (set permission explicitly). For instance, `chmod u+x myfile.txt` adds execute permission for the file owner, and `chmod go-r myfile.txt` removes read permissions for the group and others.

Chmod Calculator

Chmod calculator allows you to quickly generate permissions in numerical and symbolic formats. All extra options are included (recursive, sticky, etc). You'll be ready to copy paste your chmod command into your terminal in seconds.

Mode Explanation: Numeric vs. Symbolic Mode

The two modes of `chmod` may seem different, but they represent the same information. It is merely a matter of syntax; symbolic mode is more expressive and flexible, while numeric mode is simpler and more straightforward. Choose the mode that suits your workflow and better identifies with your expertise level.

- ☐ Verbose ⓘ

☐ Changes ⓘ

☐ Silent ⓘ

☒ Default ⓘ

☒ Recursive ⓘ

☐ Preserve-Root ⓘ

☐ Reference File ⓘ

☐ Setuid ⓘ

☐ Setgid ⓘ

☐ Sticky Bit ⓘ

```
ected token `)`
```

```
2)chmod u - x $b; ls -l;
```

Working with `chmod` Options

Useful options can control the execution of the `chmod` command. `-c` displays messages only if changes are made, `-R` changes permissions recursively for directories and their contents, and `-v` displays the execution process. For example, `chmod -R 755 mydirectory` recursively changes the directory and all files inside it to have read, write, and execute access for the user and read and execute permission for the group and others.

```
mcone — bash —
login: Tue May 10 20:44:43 on co
ll:~ mcone$ ls -l
0
-----+ 10 mcone  staff   340 May
---wx+   5 mcone  staff   170 Apr
-----+  6 mcone  staff   204 May
-----@ 35 mcone  staff  1190 Apr
-----+  3 mcone  staff   102 Apr
-----+  4 mcone  staff   136 Apr
-----+  4 mcone  staff   136 Apr
-r-xr-x+ 5 mcone  staff   170 Apr
ll:~ mcone$ chmod 755 Documents/
```

Best Practices with `chmod`

- Understand the permission system and the potential implications of changing access levels.
- Avoid granting permissions to the `others` group, as it poses security risks to the system and files.
- Check permissions regularly and if necessary, update them accordingly.
- Use appropriate syntax and options when executing changes to prevent errors and accidental changes.
- Use security-enhanced tools like SELinux or ACL to enforce additional access control mechanisms for a more robust security system.

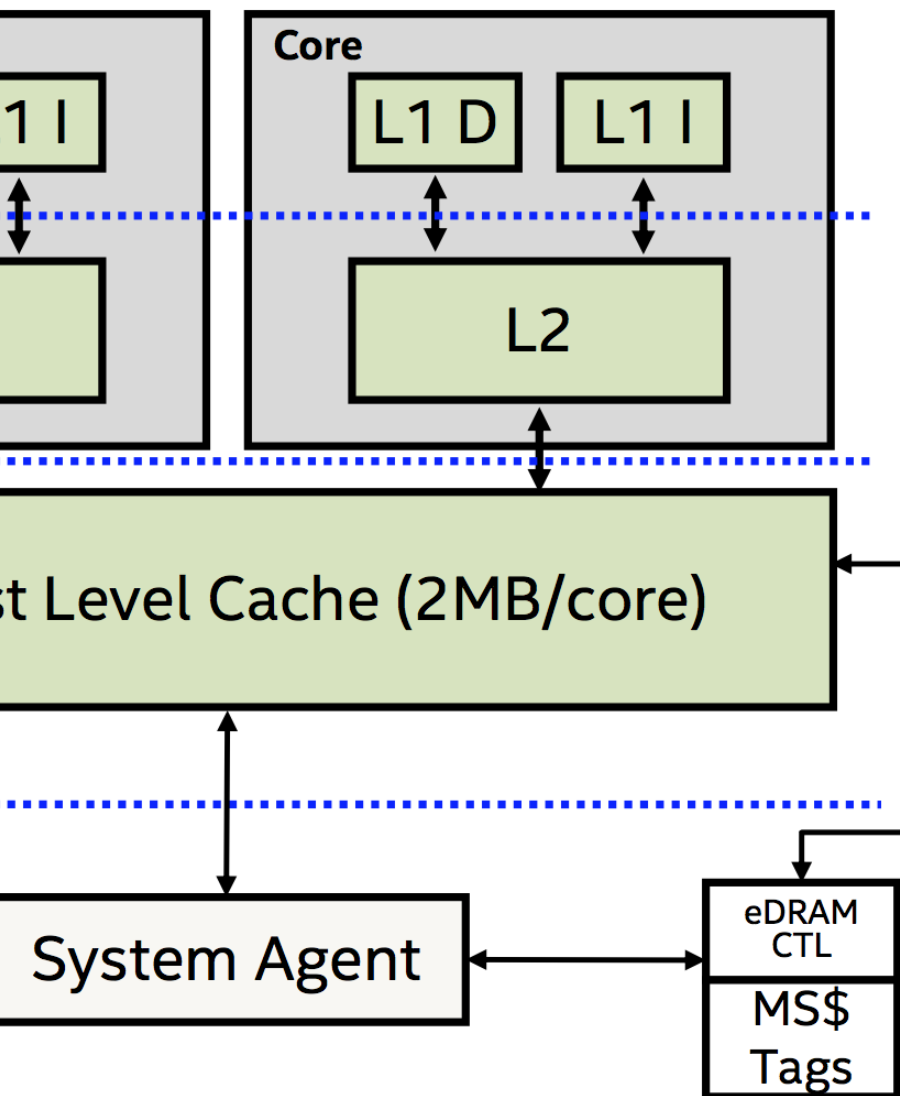
```
Box: /home/hozaifa/Downloads
```

```
home/hozaifa/Downloads# ls
                                saves
                                set_ptenv.s
                                set_qtenv.s
                                Sounds
071_64bit_linux.tar.gz templates
home/hozaifa/Downloads# packettrac
1
home/hozaifa/Downloads# ./set_qten
ATIO environment variable to /etc/
home/hozaifa/Downloads# packettrac
1
home/hozaifa/Downloads#
```

Changing File Permissions in Bulk with `chmod`

In situations where many files or directories require changes to the file permissions, `chmod` still comes in handy. For example, `chmod 644 *.txt` changes the permissions of all files with `.txt` extension in the current folder to be readable and writable by the file owner but only readable by everyone else.

Cache



Setting Permissions based on Access Levels

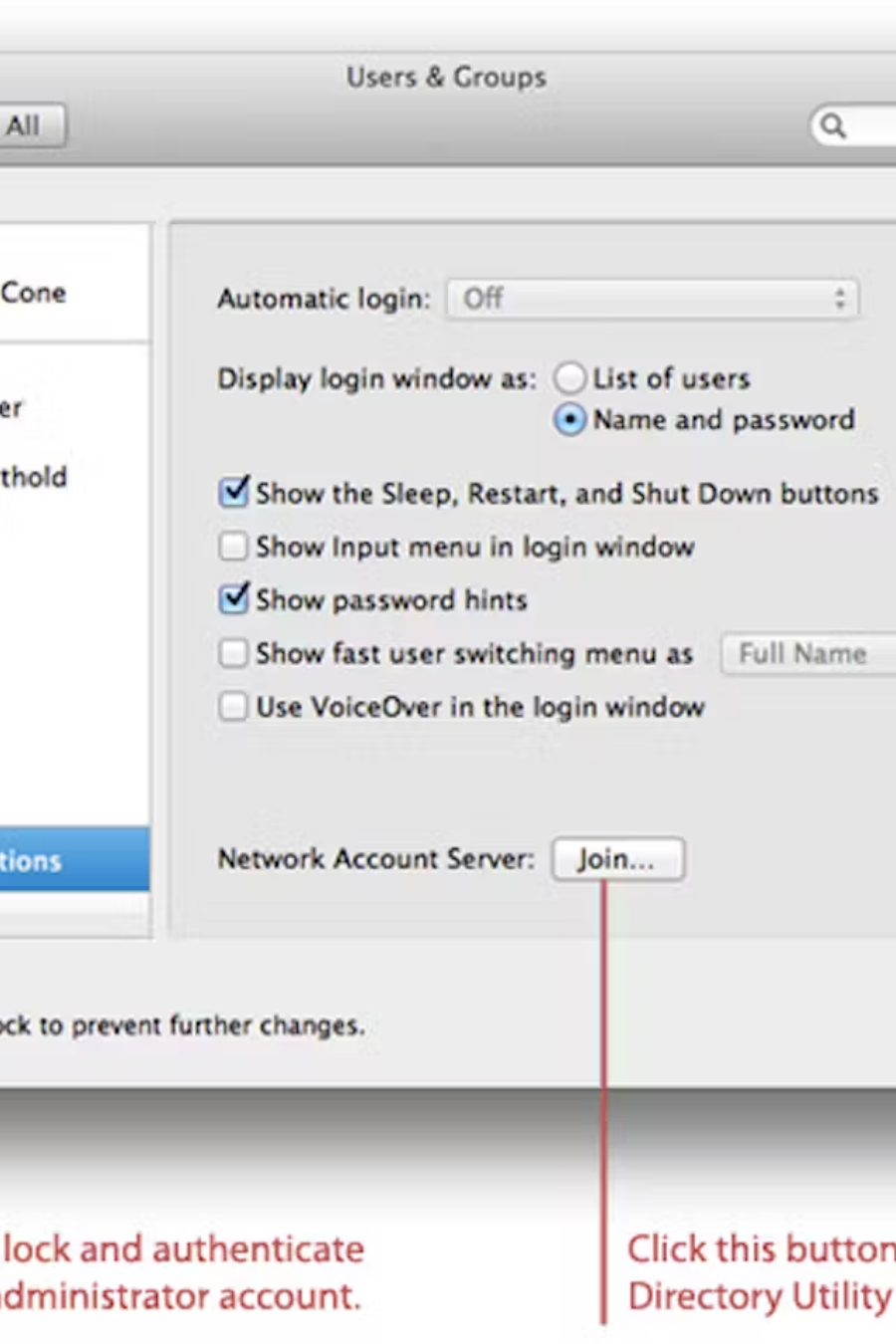
Effective use of ``chmod`` requires understanding the levels of access and their corresponding numeric values. In general, users with read, write, and execute access level = 7, read and write access level = 6, read and execute access level = 5, write and execute access level = 3, read access level = 4, execute access level = 1, and no access level = 0.



```
-rw-rw-rw- 1 root root 156597 Nov 16 06:19 Cerita Rakyat.docx
-rw-r--r-- 1 root root 20708 Sep 22 18:44 PKN 26,27.docx
-rw-r--r-- 1 root root 653488 Oct 20 19:55 StikerLinux.docx
-rw-r--r-- 1 root root 8365 Oct 20 19:19 tes.docx
-rw-r--r-- 1 root root 34486 Dec 10 21:17 Tugas Bahasa Inggris.docx
-rw-r--r-- 1 root root 37842 Dec 11 14:36 TugasFisikaRemidi.docx
-rw-r--r-- 1 root root 62107 Oct 29 19:52 Tugas Sosiologi.docx
root@iyung:~/Documents# chmod g-w,o-w /root/Documents/Writer/Cerita\ Rakyat.docx
root@iyung:~/Documents# ls -l Writer
total 1228
-rw-r----- 1 root root 25479 Oct 11 20:43 100perintahdasarlinux-130224233003-phpapp02.docx
-rw-r--r-- 1 root root 88906 Nov 26 19:46 BackupBlog.txt
-rw-r--r-- 1 root root 100750 Nov 16 06:40 Biografi Bill Gates.docx
-rw-r--r-- 1 root root 34867 Nov 17 18:59 Biografi Bill Gatesnnn.docx
-rw-r--r-- 1 root root 156597 Nov 16 06:19 Cerita Rakyat.docx
-rw-r--r-- 1 root root 20708 Sep 22 18:44 PKN 26,27.docx
-rw-r--r-- 1 root root 653488 Oct 20 19:55 StikerLinux.docx
-rw-r--r-- 1 root root 8365 Oct 20 19:19 tes.docx
```

Combining Multiple Permissions in a Single Command

It is possible to combine various permissions and groups in a single `chmod` command using commas to separate groups and no spaces between individual permission settings. A valid example is `chmod u=rw,g=rx,o=x myfile.txt` which gives the file owner read-write access, the group read and execute access, and everyone else execute access only.



Controlling Permissions for Root User

The root user has access to execute all system tasks, including modifying permissions of any file or directory. It is important to regulate the privileges granted to the root user using best practices, some of which include limiting root access, enforcing passwords or passphrases, and using sudo access features.

```
2422.188.13-1/RELEASE_ARM64
d7ffffff00000000 - 0xffffffff00000000
```

```
ved.
```

```
00000000, 0x00000000
tion complete
```

```
management
```

```
51F
```

```
sources/<string>Cmy>188sources/patch/<my><string>ID=22>test-wild-andie/<string></dict>
PCI/SRT1801F.2/appleIntelPchdr-launcher/rdtmem/launchDevice/appLaunchID/launcher/188C181echStorageDevice/1881echS
00000000)
```

```
count[ 128]
```

```
Arbalt IPD packet for route = 0d0 part = 11 using = 0
Arbalt IPD packet for route = 0d0 part = 12 using = 0
1          see launchd[1] has started up. see
1          see Verbosa host, will log to /dev/console. see
1          see Shutdown logging is enabled. see
```

```
id 2: device was reinitialized
file or directory
```

Undoing `chmod` Changes with `chown`

The `chown` command changes the ownership of a file or directory, which includes the file owner, the group, and the default file mode. If a changes process to `chmod` goes wrong, restoring the default settings for a file or directory is as simple as running the `chown` command.