



Agenda

The Linux File System (chapter 4 in text)

- Setting Access Permissions
 - Directory vs File Permissions
 - chmod Utility
 - Symbolic Method
 - Absolute Method
 - umask Utility



Access Permissions

- Limiting unauthorized access to your directories and files is a very important concern for ALL Linux (Unix) users.
- Consequences of Unauthorized Access:
 - Copying your assignments (cheating)
 - Using your account for illegal activity
 - Using your account to send obscene messages
 - Tampering with files



File / Directory Permissions

- The Linux (Unix) OS can allow the user to specify **read**, **write** and **execute** permissions to the **user**, **group** or all **others** (UGO) for files.
- A user can also specify **read**, **write** and **execute** permissions for a directory. The **execute** permission for a directory allows the person to view files in that directory



chmod Command (Relative Method)

Used to change the access permissions of a file or directory

Format:

```
chmod [option] [who] [operation] [permission] file  
chmod [option] [permission] file-list
```

- **who** relates to user (u) , group (g) or all others (o)
- **operation** relates to adding (+), removing (-) or setting (=) permissions
- **permissions** are read (r), write (w) and execute (x)

chmod Command (Relative Method)

Examples:

- Add Permission
`chmod g+w file.name`
`chmod o+x file.name`
- Remove Permission
`chmod g-w file.name`
`chmod a-w file.name` (removes write for ugo)
- Set Permission
`chmod o=rx file.name`
`chmod go=rx filename`

Note: you can use wildcard symbols (eg *) to match particular files

chmod Command (Absolute Method)

You can use the chmod command with octal number to represent (in binary) a permission (1) or removal of a permission (0) for the file or directory

This is referred to as an Absolute command, and many prefer this “short-cut” method to changing file / directory permissions

Relationship of a Binary to an Octal Number

Notice the Pattern:

- Largest 3 digit binary is 111
- 1 octal digit will represent a 3 digit binary number
- Highest Octal digit is 7
- Therefore: $111_2 = 7_8$

Binary to Octal

Relationship:

<u>Octal</u>	<u>Binary</u>
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Therefore:

Octal number **755** is equal to:

11101101 in binary

This can be related to the permissions:

rwxr-xr-x



chmod - Example (Absolute Method)

Applying octal values of rwx use the absolute chmod command:

```
chmod 777 filename - r w x r w x r w x
chmod 755 filename - r w x r - x r - x
chmod 711 filename - r w x - - x - - x
chmod 644 filename - r w - r - - r - -
```



Practical Applications of chmod Command

- Pass-Through Permissions
 - Pass-Through Permissions allow users to pass-through the home directories and other subdirectories until they reach a directory that provides read and execute permissions to read files. (pass-through permissions drwx--x--x)
 - To deny any access other than yourself, you can remove pass-through permissions of your home directory (drwx-----)



Practical Applications of chmod Command

- Linking & Sharing Files
 - Set up directory and file permissions to allow users to modify a file or set up permissions of file to allow user to view, but not modify a file.
- Webpages
 - Allow or deny other access to files. For example, use chmod command to allow group & others read and execute permissions to "pass-through" your directories.



Creating a Mask

- Are you tired of continually changing access permissions for newly-created files or directories?
- The `umask` command automatically sets the file permissions upon creation of the file.
- This process is useful, since user may sometimes forget to change the permissions of newly-created files or when they transfer files via the FTP application



umask Command

Used to automatically establish file permission upon creation

Format:

`umask [mask]`

where `mask` represents a 3-digit octal number regarding UGO and permissions to be assigned.

- Note: The rules vary between setting file masks and directory masks



Setting Directory Mask

To change directory mask:

- Determine octal number that would set directory permission
- Subtract octal number 777 from octal number determined above to get result
- issue the command :
`umask [octal number]`

Note: should also be able to use "relative method" with umask command – may be easier



Setting Directory Mask

Example:

- To set mask for newly-created directories to:
`rwxr--r--`
- Determine octal number
`111 100 100 = 744`
- Subtract 777 from 744 = 033
- Issue command `umask 033`
- Issue command `umask` to verify change



Setting File Mask

To change directory mask:

- Determine binary number that would set directory permission
- Subtract above binary number from 110110110 and convert result to octal number to determine umask value
- issue the command :
`umask [octal number]`

Setting File Mask – Example1

Example:

- To set mask for newly-created files to:
r w - r - - r - -
- Convert to binary
110100100
- Subtract above from 110110110
110110110
- 110100100
= 000010010 (which is 022)
- issue umask 022 (enter umask to verify)

Setting File Mask – Example2

Example:

- To set mask for newly-created files to:
r w x r - - r - -
- Convert to binary
111100100
- Subtract above from 110110110
110110110
- 111100100 ← Cannot subtract 1 from 0
= 000010010 (which is 022 octal)
- issue umask 022 (enter umask to verify)