i-nodes, links, attributes of a file and compression commands

Enrique Arias
Universidad de Castilla–La Mancha

Contents

- Main objectives
- Concept of i-node
- Soft and hard links
- Attributes and permissions of files
- File compression and backup tools

Contents

- Main objectives
- Concept of i-node
- Soft and hard links
- Attributes and permissions of files
- File compression and backup tools

Main objectives

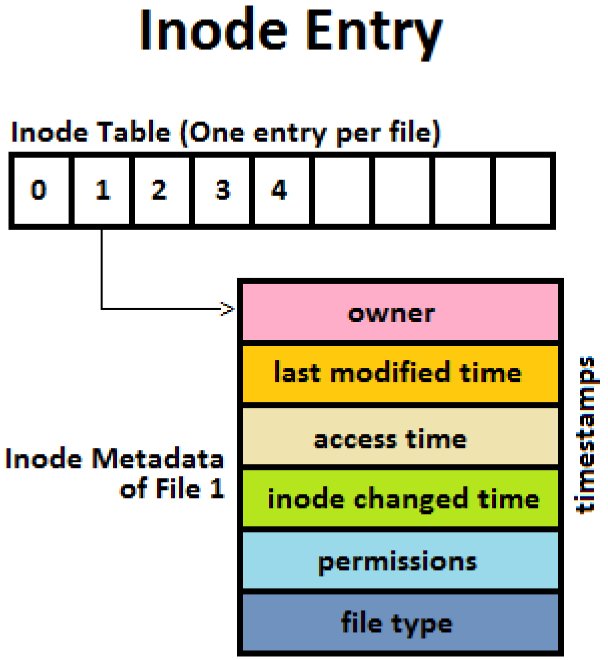
- Know the concept of i-node.
- Know the what are soft and hard links
- Know the attributes of files as well as permissions
- Learn about file compression and backup tools: gzip, gunzip, tar.

Contents

- Main objectives
- Concept of i-node
- Soft and hard links
- Attributes and permissions of files
- File compression and backup tools

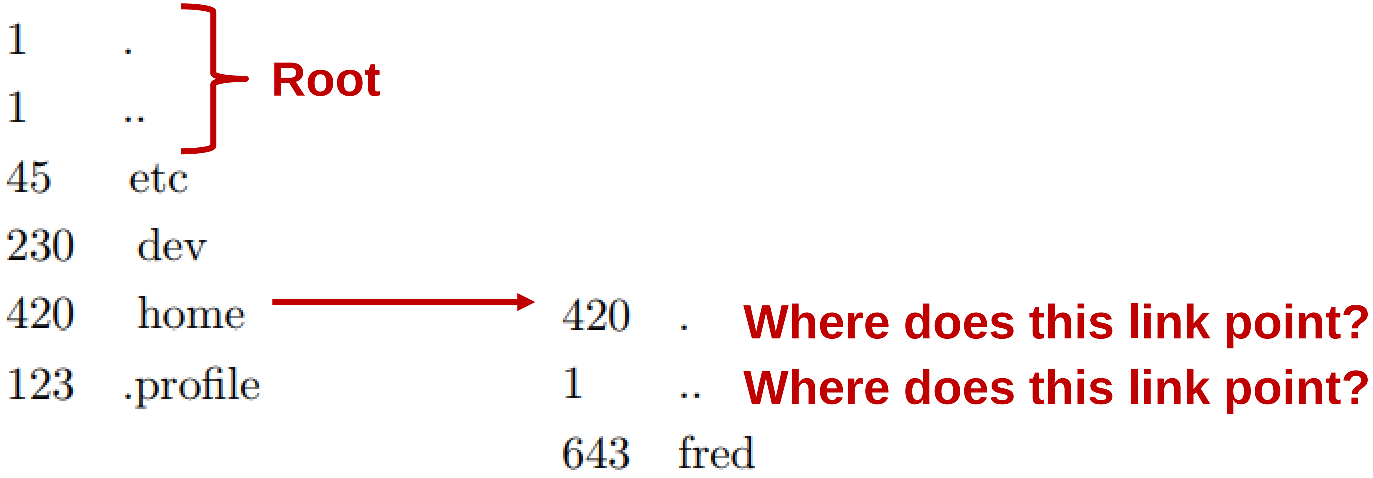
Concept of i-node

- An inode represents a file → metadada of a file
- Which metadata?
  - □ Owner
  - □ Storage allocation
  - □ Counter of links
  - □ Etc.



Concept of i-node

- Inode allows to reconstruct the file system



1	.
1	..
45	etc
230	dev
420	home
123	.profile

Root

420	.
1	..
643	fred

Where does this link point?

Where does this link point?

Contents

- Main objectives
- Concept of i-node
- Soft and hard links
- Attributes and permissions of files
- File compression and backup tools

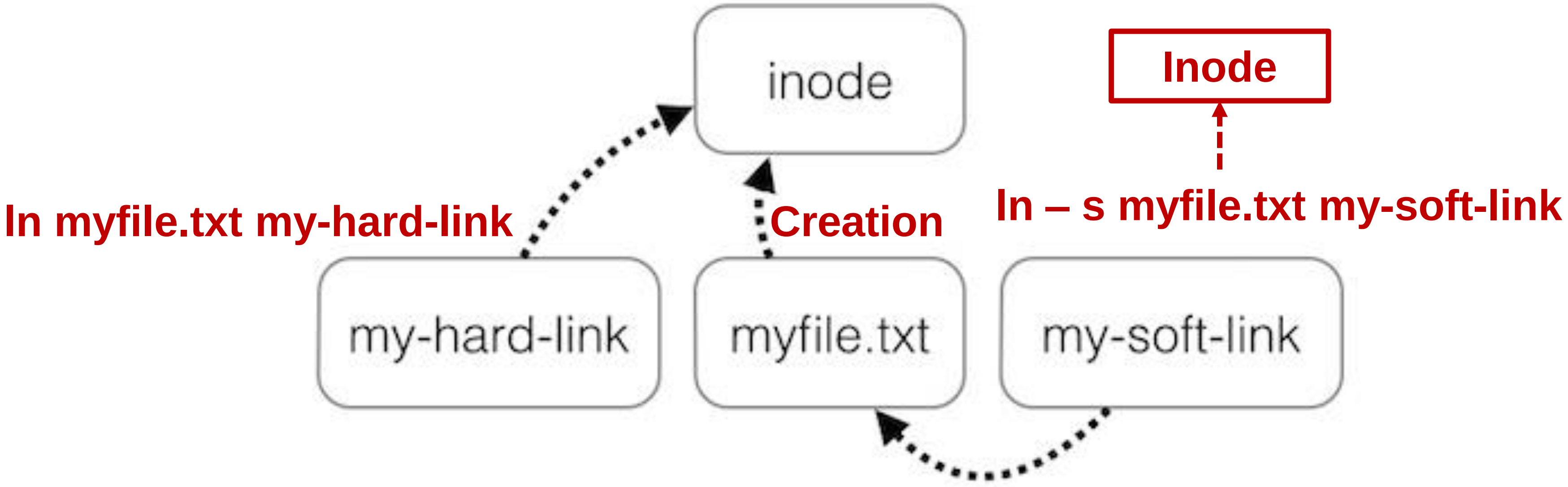
Soft and hard links

- Hard links → $ ln [original filename] [link name]
    - Each hard linked file is assigned the same Inode value as the original, therefore they reference the same physical file location.
    - Links have actual file contents
    - ls -l command shows all the links → number of links.
    - Removing any link, just reduces the link count, but doesn’t affect other links.
    - We cannot create a hard link for a directory to avoid recursive loops.

Soft and hard links

- Soft (symbolic) links → $ ln -s [original filename] [link name]
  - □ A soft link is similar to the file
  - □ Each soft linked file contains a separate Inode value that points to the original file.
  - □ ls -l command shows all links with first column value 1 and the link points to original file
  - □ Soft Link contains the path for original file and not the contents.
  - □ Removing soft link doesn’t affect anything but removing original file, the link becomes “dangling” link which points to nonexistent file
  - □ A soft link can link to a directory

Soft and hard links



Inode

In myfile.txt my-hard-link

Creation

In – s myfile.txt my-soft-link

Soft and hard links

- Exercise 1
  - At HOME directory create a file called “a” containing your name
  - Create a hard link to “a” called “b” and a soft link to “a” called “c”
  - Obtain the inode number (ls –i) of “a”, “b” and “c”.
  - Delete “a”. What happens with “b” and “c”? Could you access to the content?

Contents

- Main objectives
- Concept of i-node
- Soft and hard links
- Attributes and permissions of files
- File compression and backup tools

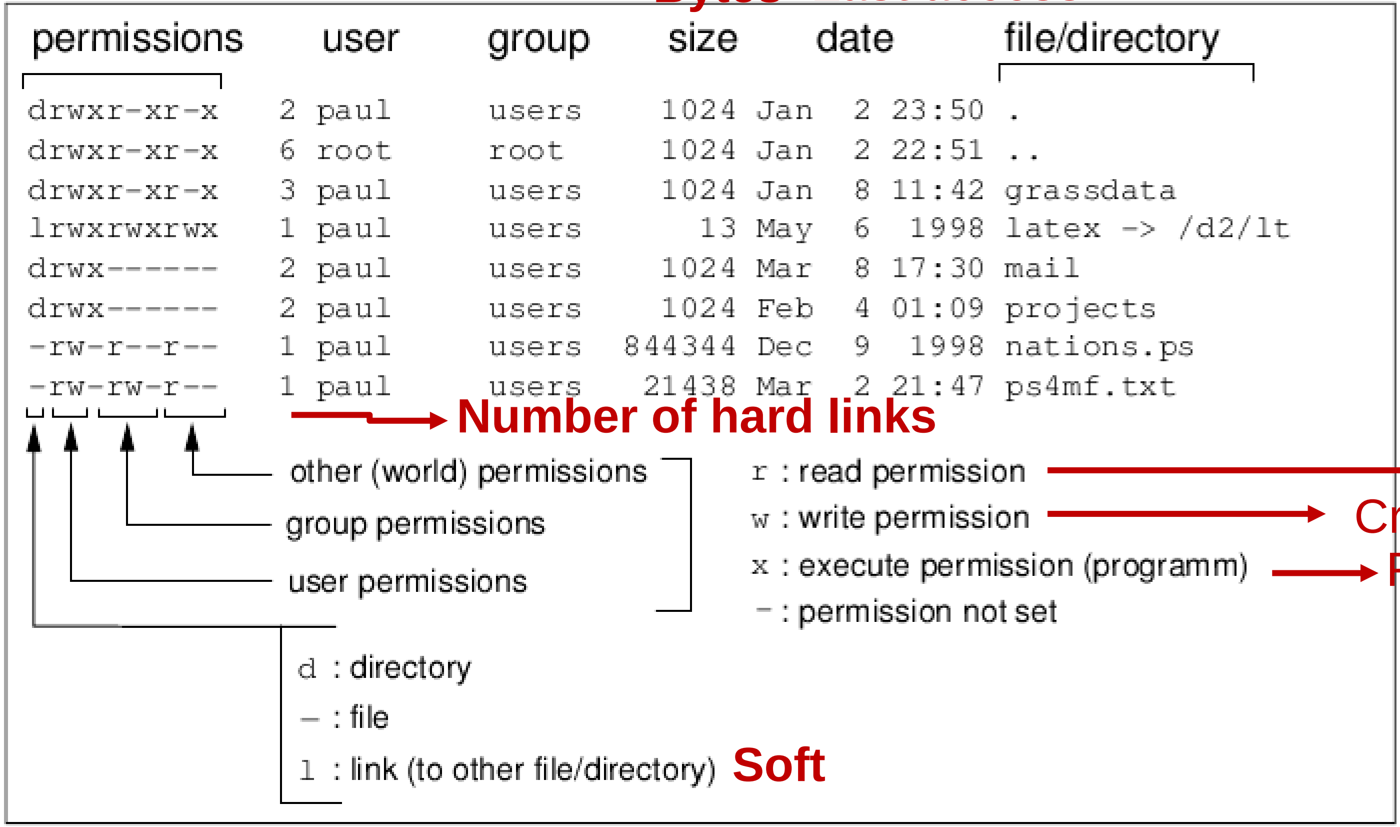
Attributes and permissions of files

- **ls name (file or directory)→ list**
  - Use of patterns for names
    - *: whatever string
    - ?: whatever character
  - Options
    - -l → long
    - -a → hidden
    - -R → recursive
    - -r → reverse alphabetical order
    - -d → attributes of a directory
    - -i → number of inode

Attributes and permissions of files

- **ls name (file or directory)→ list**

Bytes **Last access**



```
drwxr-xr-x   2 paul     users     1024 Jan  2 23:50 .
drwxr-xr-x   6 root     root      1024 Jan  2 22:51 ..
drwxr-xr-x   3 paul     users     1024 Jan  8 11:42 grassdata
lrwxrwxrwx   1 paul     users       13 May  6  1998 latex -> /d2/lt
drwx------   2 paul     users     1024 Mar  8 17:30 mail
drwx------   2 paul     users     1024 Feb  4 01:09 projects
-rw-r--r--   1 paul     users   844344 Dec  9  1998 nations.ps
-rw-rw-r--   1 paul     users    21438 Mar  2 21:47 ps4mf.txt
```

Number of hard links

Soft

Directory meaning

Read the content

Create or delete files in it

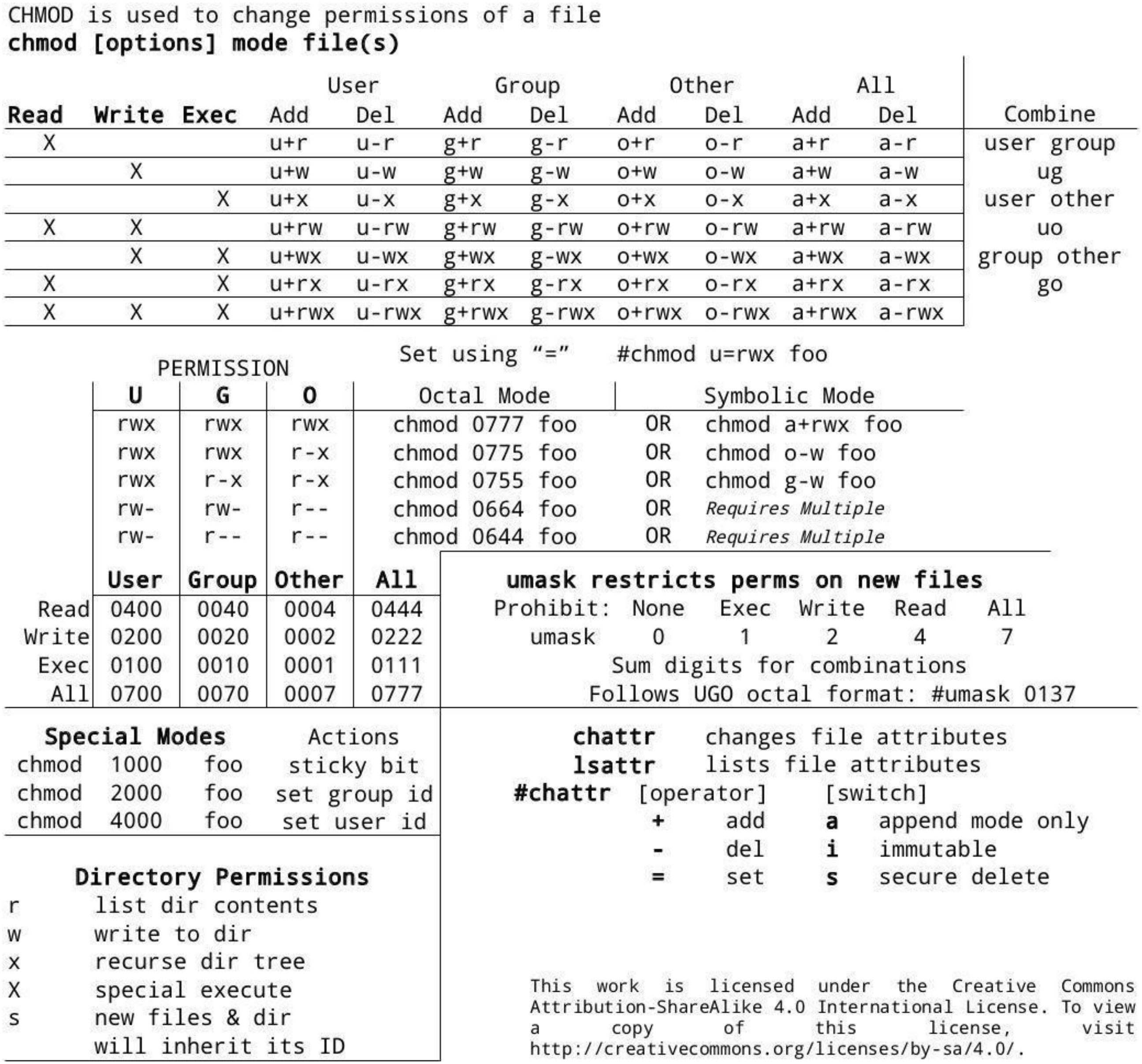
Find files in it or access subdirectories

Attributes and permissions of files

- Exercise 2
  - At HOME list the directory showing all the information including hidden files and directories. Take note of the information
  - Now, create 2 subdiretories at HOME and do the same that before. Is there any difference? In affirmative case, could you explain it?

Attributes and permissions of files

- Permission
  - □ chmod
  - □ chown
  - □ chgrp
  - □ umask



CHMOD is used to change permissions of a file

chmod [options] mode file(s)

			User		Group		Other		All		
Read	**Write**	**Exec**	Add	Del	Add	Del	Add	Del	Add	Del	Combine
X			u+r	u-r	g+r	g-r	o+r	o-r	a+r	a-r	user group
	X		u+w	u-w	g+w	g-w	o+w	o-w	a+w	a-w	ug
		X	u+x	u-x	g+x	g-x	o+x	o-x	a+x	a-x	user other
X	X		u+rw	u-rw	g+rw	g-rw	o+rw	o-rw	a+rw	a-rw	uo
	X	X	u+wx	u-wx	g+wx	g-wx	o+wx	o-wx	a+wx	a-wx	group other
X		X	u+rx	u-rx	g+rx	g-rx	o+rx	o-rx	a+rx	a-rx	go
X	X	X	u+rwx	u-rwx	g+rwx	g-rwx	o+rwx	o-rwx	a+rwx	a-rwx	

Set using “=”   #chmod u=rwx foo

PERMISSION					
U	**G**	**O**	Octal Mode		Symbolic Mode
rwx	rwx	rwx	chmod 0777 foo	OR	chmod a+rwx foo
rwx	rwx	r-x	chmod 0775 foo	OR	chmod o-w foo
rwx	r-x	r-x	chmod 0755 foo	OR	chmod g-w foo
rw-	rw-	r--	chmod 0664 foo	OR	*Requires Multiple*
rw-	r--	r--	chmod 0644 foo	OR	*Requires Multiple*

	User	**Group**	**Other**	**All**
Read	0400	0040	0004	0444
Write	0200	0020	0002	0222
Exec	0100	0010	0001	0111
All	0700	0070	0007	0777

umask restricts perms on new files

Prohibit:	None	Exec	Write	Read	All
umask	0	1	2	4	7

Sum digits for combinations

Follows UGO octal format: #umask 0137

Special Modes			Actions
chmod	1000	foo	sticky bit
chmod	2000	foo	set group id
chmod	4000	foo	set user id

chattr changes file attributes

lsattr lists file attributes

#chattr	[operator]		[switch]	
	+	add	**a**	append mode only
	-	del	**i**	immutable
	=	set	**s**	secure delete

Directory Permissions

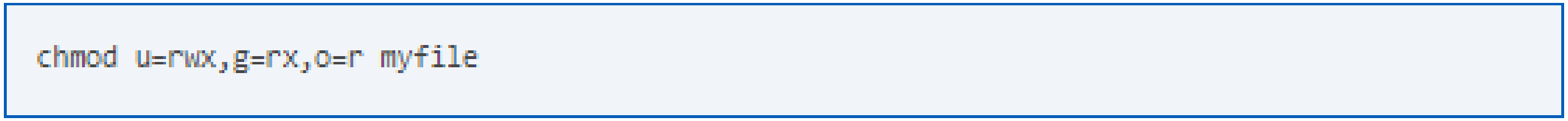
r	list dir contents
w	write to dir
x	recurse dir tree
X	special execute
s	new files & dir will inherit its ID

Attributes and permissions of files

- chmod [-R] who action permission, who action permission, … file/directory name

Who	Action	Permission
u= user	+=add	r = read
g= group	-=delete	w= write
o= other	= = set	x= execute

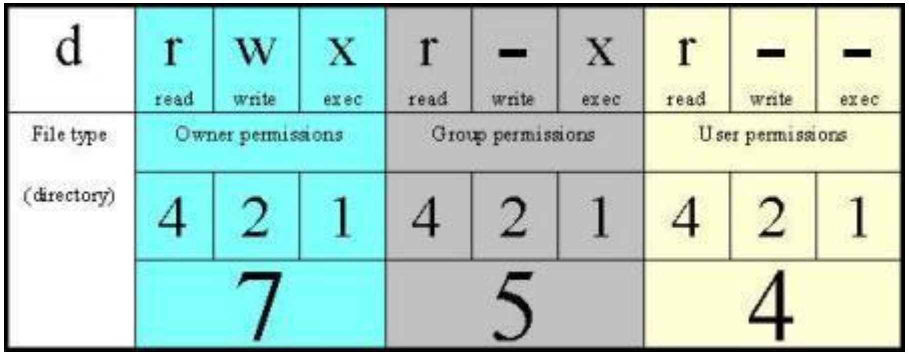
- Example



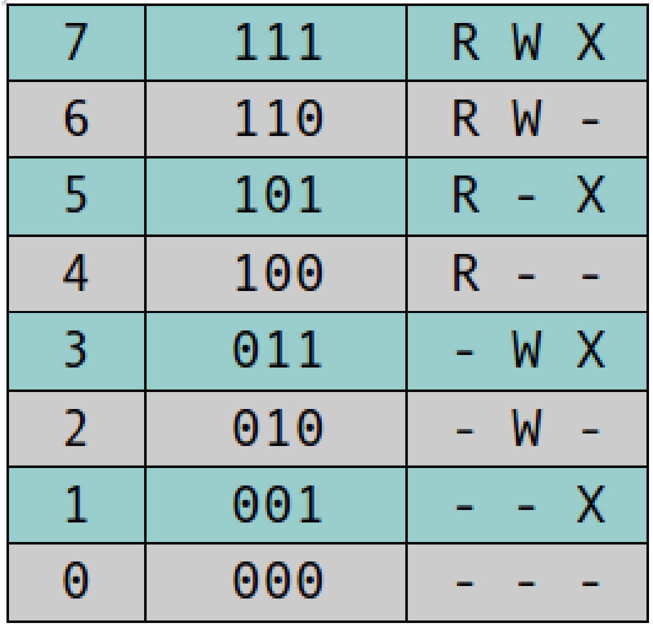
```
chmod u=rwx,g=rx,o=r myfile
```

Attributes and permissions of files

- chmod [-R] number file/directory name

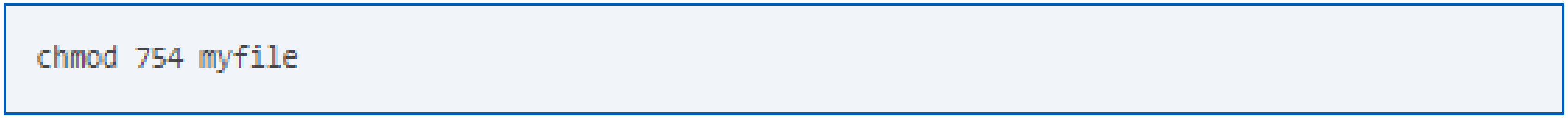


d	r read	w write	x exec	r read	- write	x exec	r read	- write	- exec
File type (directory)	Owner permissions			Group permissions			User permissions		
	4	2	1	4	2	1	4	2	1
	7			5			4		



7	111	R W X
6	110	R W -
5	101	R - X
4	100	R - -
3	011	- W X
2	010	- W -
1	001	- - X
0	000	- - -

- Example



```
chmod 754 myfile
```

Attributes and permissions of files

- chown [-R] user file → changes ownership of files and directories (only root)
- chgrp [-R] group file → changes group ownership of a files or files (only root)
- umask mask
  - Description: set the value of the system's file mode creation mask
  - Mask must be an even number which will be subtracted to 666 to know the file permission
  - Example: To create a new file with permissions rw-rw-r– (664) the mask must be 002 (666-002=664).

Attributes and permissions of files

- Exercise 3
  - □ Look at HOME directory and the permissions of the different files and directories. Could you explain those permissions associated to file “b”?
  - □ Now, delete the read, write and execute permission for the group and other using both chmod forms.
  - □ Agree with a colleague to access the file. What happens?
  - □ Change the reading permission for the group. And now, what happens?
  - □ Set the new creation permissions for reading and writing for the user and only reading permission for the group and other. Create a new file and check that this mask has been really use

Contents

- Main objectives
- Concept of i-node
- Soft and hard links
- Attributes and permissions of files
- File compression and backup tools

File compression and backup tools

- gzip (compress files) and gunzip (decompress files)

gzip file1.pdf file2.pdf … → file1.pdf.gz file2.pdf.gz …

gunzip → reverse operation

- Tar [options] file.tar directory → create or extract file.tar with the contect/the content of the directory
  - □ Options
    - c: creates a file.tar with the content of the directory
    - x: extracts the content in file.tar
    - z: use gzip or gunzip if option c or x is used
    - v: verbose. Show infomation of the processed files
    - t: show the content of a tar file
    - f: this options must always be used to indicate that a file is going to be read or wtitten
  - □ Example: tar –cvzf file.tgz /home/user

File compression and backup tools

- Exercise 4
  - Create a tar file of the HOME directory.
  - Now check the content of this file and if the previous action has been correctly done

Contents

- Main objectives
- Concept of i-node
- Soft and hard links
- Attributes and permissions of files
- File compression and backup tools

i-nodes, links, attributes of a file and compression commands

Enrique Arias

Universidad de Castilla–La Mancha