Access Management

Sathish H Bowatta

Changing Host Name

```
echo “newhostname” > /etc/hostname
```

Special Characters

Character	Description	Example
\	Escape Character	
/	Directory separator	/urs/src/linux
.	Current directory. Also can “hide” files when it is first character in a filename	
..	Parent Directory	
~	Home directory	

Character	Description	Example
*	Represent 0 or more characters in a filename, or by itself, all files in directory	log20*.txt represent log201.txt, log2017.1.txt
?	Represent a single character in a filename	hello?.txt can represent hello1.txt, hello.txt, but not hello22.txt
|	Redirect output of a command into another	ls | less
>	Redirect output of a command into a file. If file already exists, over-write it.	ls > directory\ list.txt
>>	Redirect the output of a command onto the end of an existing file	echo “nameserver 8.8.8.8” >> / etc/resolv.conf
;	Command separator. Allows to execute multiple commands on a single line	mkdir test; cd test; touch test.txt

Getting Help

Built-In Help

```
-h or –help
```

Online Manual “Man Pages”

```
man ls
man man
```

Searching for help

```
man -k permission
```

Info Pages

Info pages are similar to man page, but instead of being displayed on one long scrolling screen, they are presented in shorter segments with links to other pieces on information

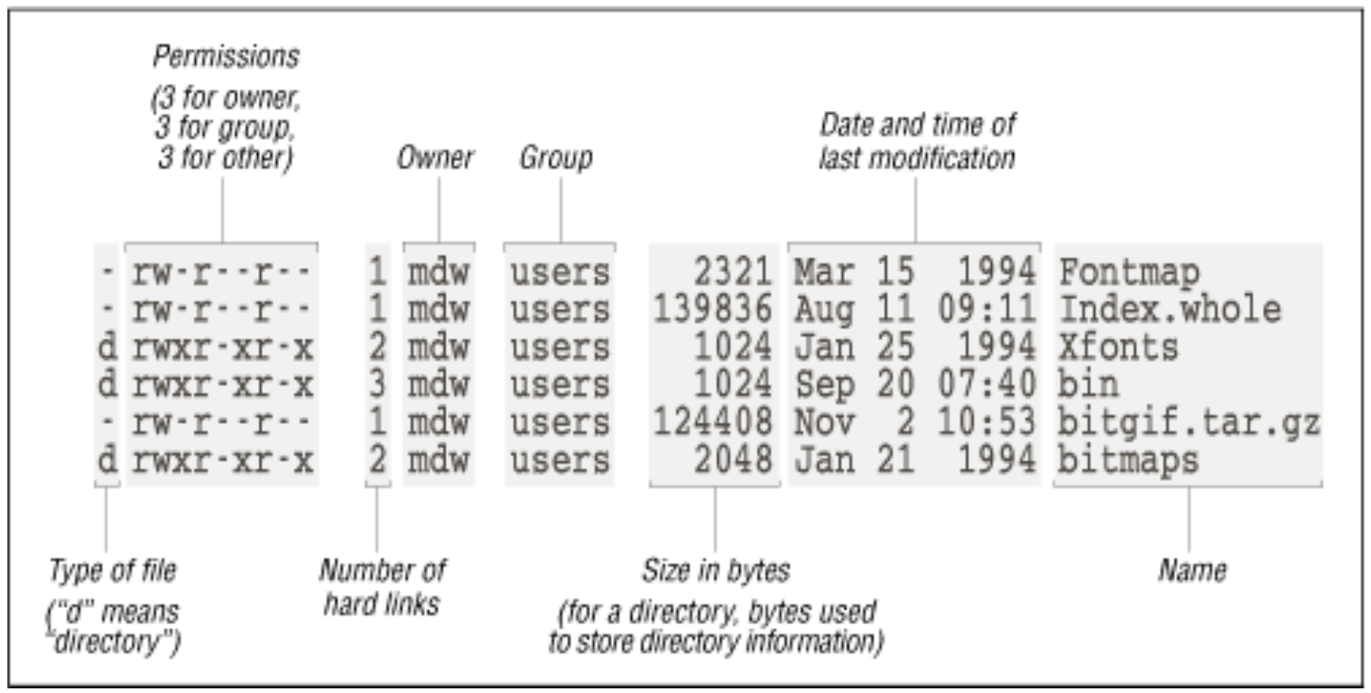
```
info ls
```

Basic Commands

Command	Description
pwd	Print working directory
cd	Change directory
mkdir	Make directory
rmdir	Remove directory
rm	Remove files or directories
file	Find out what kind of file it is
cat	Display the contents of a text file on the screen
cp	Copy files or directories
mv	Move files or directories
which	Shows the full path of shell commands
whereis	Locates the program, source code, and manual page for a command (if available)
locate	A quick way to search for files anywhere on the filesystem
find	It can be used to search for files matching certain patterns, as well as many other types of searches

File Attributes

- Size in bytes
- The owner identifier if the file, that is, the user who created the file
- The group identifier of users who own the file
- Number of hard links
- Access permissions
- Dates of access and modification
- Type of file
  - **-** regular file
  - **l** link file
  - **d** directory
  - **p** pipe
  - **c** character special device
  - **b** block special device

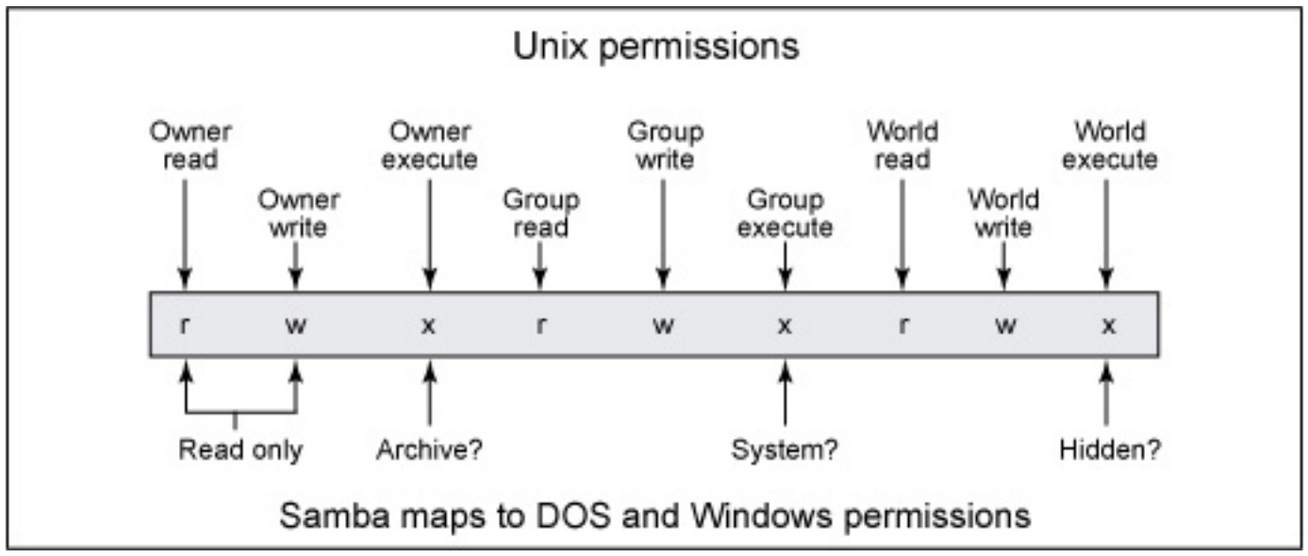


Levels of Permissions

Unix has three levels of permissions :

- `owner`
- `group`
- `other`

The “`other`” level covers everybody who has access to the system and who isn't the owner or a member of the group



File / Directory Permissions

File Access Modes

Read - Grants the capability to read

Write - Grants the capability to modify, or remove the contents of the file

Execute - Users with execute permissions can run a file as a program

Directory Access Modes

Read - Grants the capability to read the contents

Write - Grants the capability to add or delete files from the directory

Execute - Grans the capability to access the directory

{ ls, cd }

Change file modes or Acces Control Lists

Using “chmod” in Symbolic Mode

chmod Operator	Description
+	Adds the designated permission(s) to a file or directory
-	Removes the designated permission(s) from a file or directory
=	Sets the designated permission(s)

Examples :

```
chmod o+wx filename
chmod u-x filename
chmod g=rx filename
```

Using “chmod” with Absolute Permissions

Number	Description	Ref
0	No Permission	- - -
1	Execute Permission	- - x
2	Write Permission	- w -
3	Execute and write permission : 1 (execute) + 2 (write) = 3	- w x
4	Read permission	r - -
5	Read and execute permission : 4 (read) + 1 (execute) = 5	r - x
6	Read and write permission : 4 (read) + 2 (write) = 6	r w -
7	All permissions : 4 (read) + 2 (write) + 1 (execute) = 7	r w x

Examples :

```
chmod 755 filename
chmod 705 filename
chmod 043 filename
```

File / Directory Ownership

chown - stands for “change owner”. used to change the owner of a file

```
chown username filelist
chown :groupname filelist
chown username:groupname filelist
```

chgrp - stands for “change group”. used to change the group of a file

```
chgrp groupname filename
```

User Mask

The user file-creation mode mask (umask) is use to determine the file permission for newly created files

It can be used to control the default file permission for new files. It is a four-digit octal number

You can setup umask in `/etc/bashrc` or `/etc/profile` file for all users. By default most Linux distributions set it to 0022 (022) or 0002 (002)

To view your current umask setting, open a terminal and run the command:

```
umask
```

To change the umask setting of the current shell to something else, say 077, run:

```
umask 077
```

User Mask Value Table

umask Value Octal (xyz)	Default File Permissions	666 - xyz	Default Directory Permissions	777 - xyz
000	rw-rw-rw	666	rwxrwxrwx	777
002	rw-rw-r--	664	rwxrwxr-x	775
022	rw-r--r--	644	rwxr-xr-x	755
026	rw-r-----	640	rwxr-x--x	751
046	rw--w----	620	rwx-wx--x	731
062	rw----r--	604	rwx--xr-x	715
066	rw-------	600	rwx--x--x	711
222	r--r--r--	444	r-xr-xr-x	555
600	---rw-rw-	066	--xrwxrwx	177
666	---------	000	--x--x--x	111
777	---------	000	---------	000

Special File Permissions

The set-group identification (setgid) permission is similar to setuid, except that the process's effective group ID (GID) is changed to the group owner of the file, and a user is granted access based on permissions granted to that group.

The /usr/bin/mail command has setgid permissions

```
- -r-x--s--x 1 root mail 63628 Sep 16 12:01 /usr/bin/mail
```

The sticky bit is a permission bit that protects the files within a directory.

If the directory has the sticky bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by root.

This special permission prevents a user from deleting other users' files from public directories such as /tmp

```
- drwxrwxrwt 7 root sys 400 Sep 3 13:37 tm
```

Real, Effective and Saved UID's

Each Linux process has 3 UIDs associated to it.

Real UID

The UID of the process that created THIS process.

Effective UID

This is used to evaluate privileges of the process to perform a particular action.

Saved UID

For the binary image file with a setuid bit on

Example

“**passwd**” program’s setuid bit is on and the owner is“**root**”. When a normal user, say “**sathish**”, runs “**passwd**”, passwd starts with

RUID= sathish EUID= sathish SUID=root

Then, the program calls a system call “**setuid**” and since the setuid bit is on, the UIDs will be

RUID= sathish EUID=root SUID=root

After that, “**passwd**” will be able to access **`/etc/passwd`** file and change the file for “**sathish**”.

(Note that “**sathish**”cannot write to **`/etc/passwd`** on its own).

Time Associated with a File

Three times tracked for each file in Unix are these:

access time - atime

change time - ctime

modify time - mtime

atime — File Access Time

Access time shows the last time the data from a file was accessed – read by one of the Unix processes directly or through commands and scripts.

ctime — File Change Time

ctime also changes when you change file's ownership or access permissions. It will also naturally highlight the last time file had its contents updated.

mtime — File Modify Time

Last modification time shows time of the last change to file's contents. It does not change with owner or permission changes, and is therefore used for tracking the actual changes to data of the file itself.

Users and Groups

Command	Description
useradd	Adds accounts to the system
usermod	Modifies account attributes
user del	Deletes accounts from the system
groupadd	Adds groups to the system
groupmod	Modifies group attributes
groupdel	Removes groups from the system
passwd	Modifies user’s password

Cron Jobs

List current crontab files for specific users

```
crontab -l -u username
```

Remove current crontab

```
crontab -r
```

Edit current crontab

```
crontab -e
```