

# **CYBR 4423**

## **Linux/Unix Administration**

# **Linux File System and Operations**

# Overview

## Linux file system components

- Directory hierarchy, path
- File attributes: name, path, type, permission, etc.

## Common file operation commands

# File System

A file system is a set of data structures that represent and organize the system's storage resources

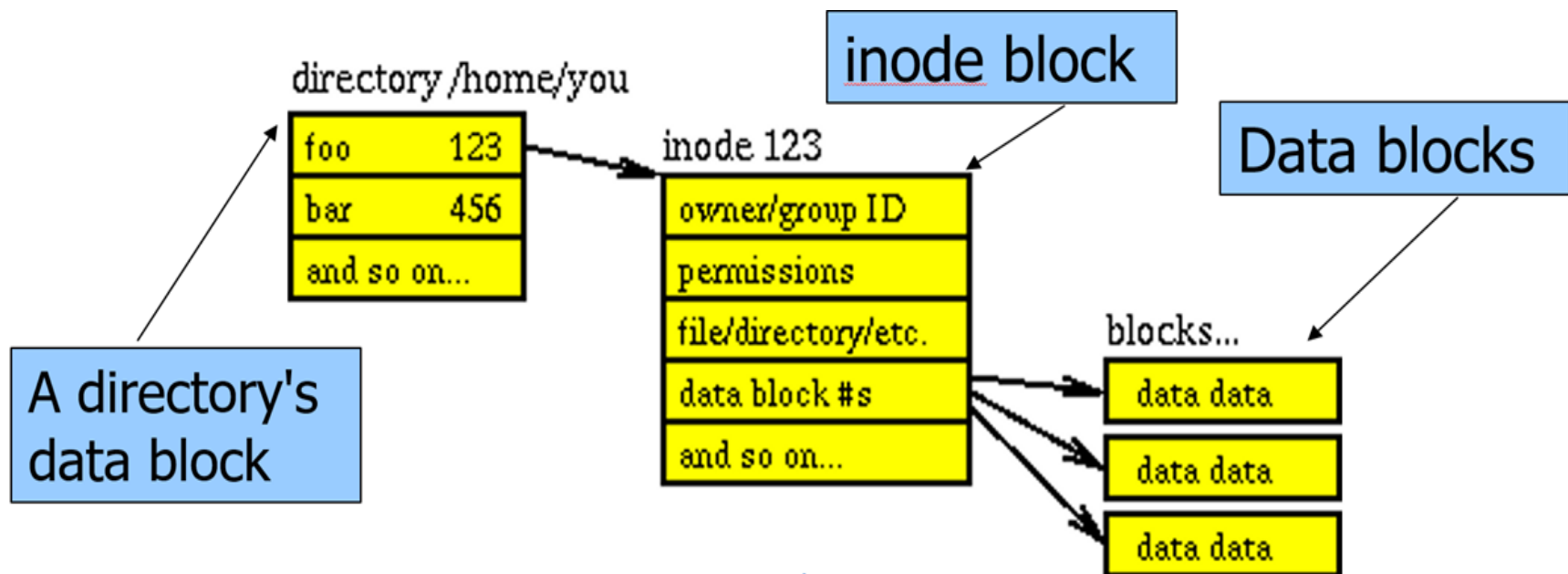
## Common types of disk file systems

- NTFS: used for Windows NT, and more recent Windows
- FAT, FAT32: used for DOS, earlier Windows, and many removable storage devices
- ext2, ext3, ext4: used for Linux
- ISO 9660, UDF: used for optical disks

# Linux File System Structure

## Basic components of a Linux file system

- inode blocks: containing file metadata including
  - inode number
  - File types, timestamp, permissions, etc.
  - Pointers to data (content) blocks
- Data blocks: actual file content. For directory's content block, **file names and inode numbers** are stored in it.
- These two type of blocks are stored in different place of the hard drive.



# Inode

## The inode block contains file metadata

- Inode number: a unique number assigned to a file
- File type
- Timestamp: access, content change, metadata change
- File size
- Number of links
- File permissions
- Owner and group id
- Data block pointers
- Other extended information
- Note there is **NO file name** stored in inode blocks. Files names are stored in directory's content block.

Use "ls" or "stat" command to view most of these data

# File Types

## The Unix way

Every object is treated as a file and mapped to the file system, including directories, devices, etc.

## File types

- **Regular file:** files that contain content
- **Directory:** a holder for other files
- **Symbolic link:** a pointer to other files
- Character device file
- Block device file
- Local domain socket
- Named pipe

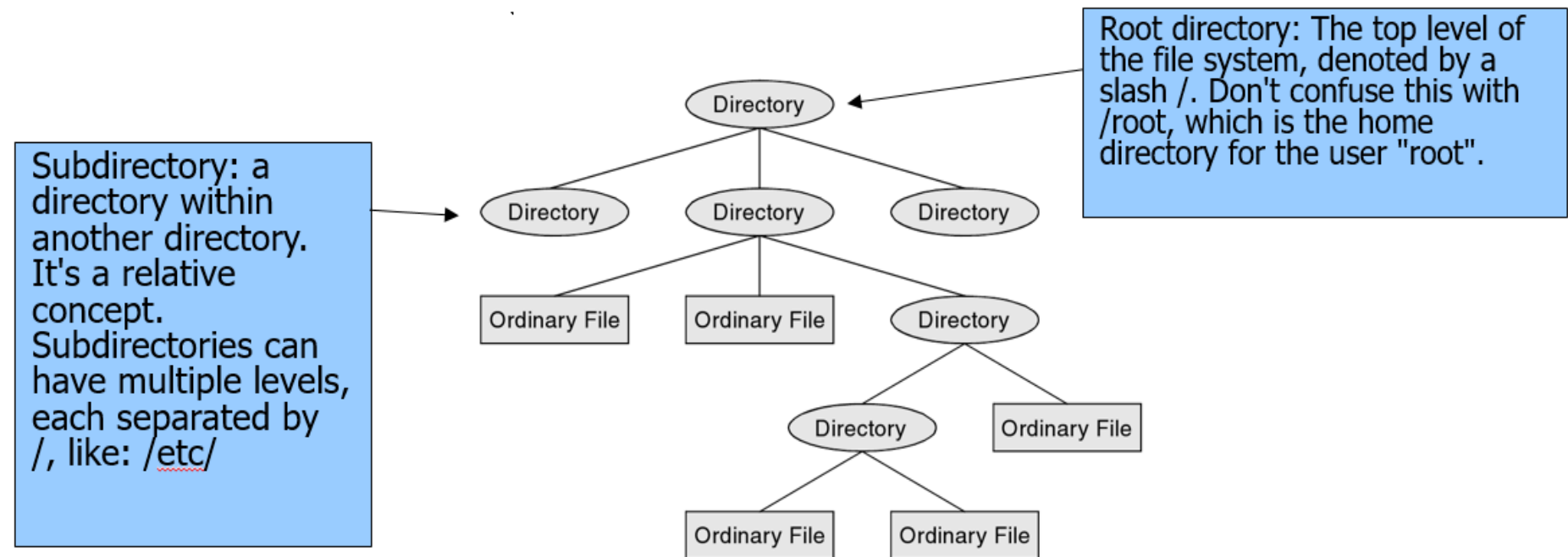
Use “file” command and “ls” command to find out file types

# Directory Hierarchy

Directory is a special file that contains entries of other files/directories

For each entry, the file name and inode are recorded

The complete structure of directories is a tree (hierarchical) structure



# More Directory Terminologies

## Current directory (working directory)

- The directory you are in at a given time, sometimes called the current working directory. The pwd (print working directory) command will tell you the name of the current directory.
- Using a single dot "." to represent the working (current) directory – most of the time you don't really use it.

## Parent directory

- The directory above the current one. Every directory except the top level has a parent.
- Using double dots ".." to represent the parent directory
- For example, if you are in the /usr/spool directory, then ../ or /usr is the parent.

Every directory has at least two files (directories): . and ..

## Home directory (personal directory)

- A user's personal directory. The user normally has complete control over all files stored in directories beneath the home directory. Exception: the root user can go everywhere.
- Use "~" to represent the home directory
- For example: if the user name is gzheng, then your home directory is /home/gzheng/, or ~/ ; ~/Documents is same as /home/gzheng/Documents





# File Path

Path defines the complete reference name to a file

Unique for each file

## Type

Absolute: starting from the root “/”. You may use the same path to refer to the file no matter where you are.

Relative: the path to a file is relative to the current (working) directory. It varies depending on where you are.

# File Path

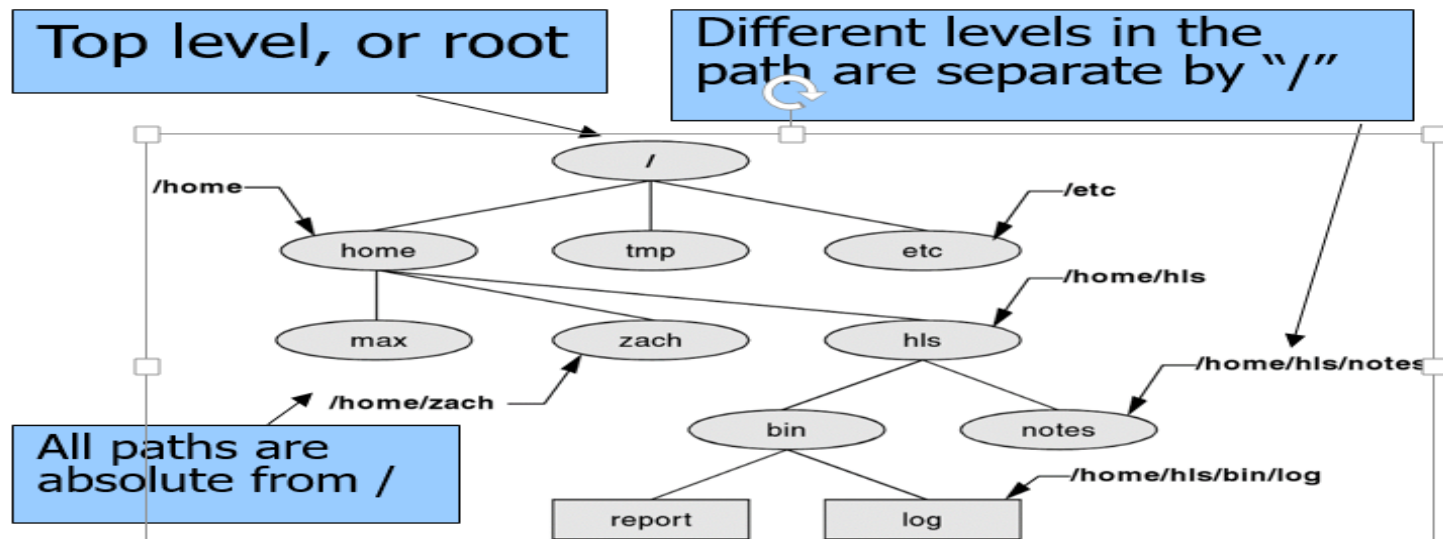


Figure 6-5 Absolute pathnames

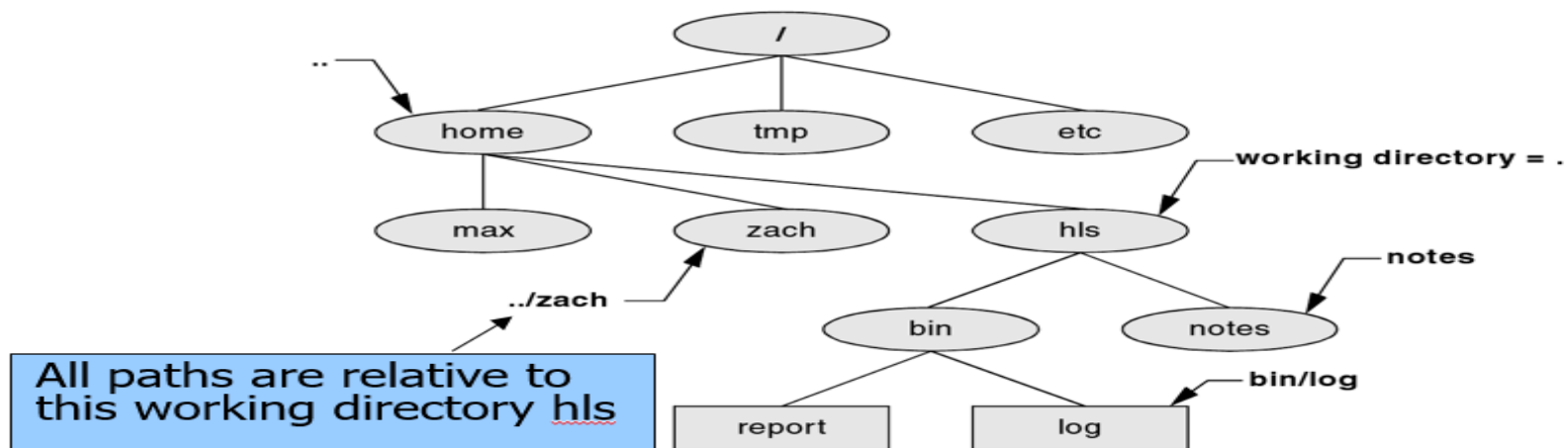


Figure 6-6 Relative pathnames

9

# Important Directories

Filesystem Hierarchy Standard (FHS) defines the main directories and their contents in Linux file systems

[Filesystem Hierarchy Standard](#)

[Linux Directory Structure Explained with Examples](#)

## Major directories

- /bin essential command binaries
- /boot boot loader files
- /dev devices
- /etc system configuration files
- /home user's place
- /media mount points for removable storage
- /mnt temporarily mounted file system
- /root for the root (admin) user
- /sbin essential system binaries
- /usr user's read-only data
- /var
  - /var/log

# File Names

File names are **NOT** recorded in inodes, but in the data (content) block of directory files

## Naming rules

- Can contain any characters except "/"
- Length: 255 characters
- Case sensitive
- No extension is required
- No same file name in the same directory

## Hidden files/directories

File/directory name starting with "."

## Special characters in file names

- Such as: \*, -, (blank space), etc.
- Use ' ' around these names

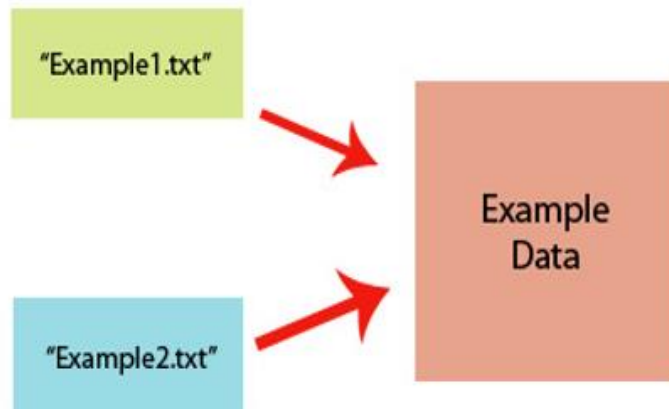
# Link

Links are used to point to files from different places.

- Hard link is created to have the same inode as the target file
- Soft link is created to have a different inode
- Hard links don't rely on one another; while a soft link is dependent on another file name and location.

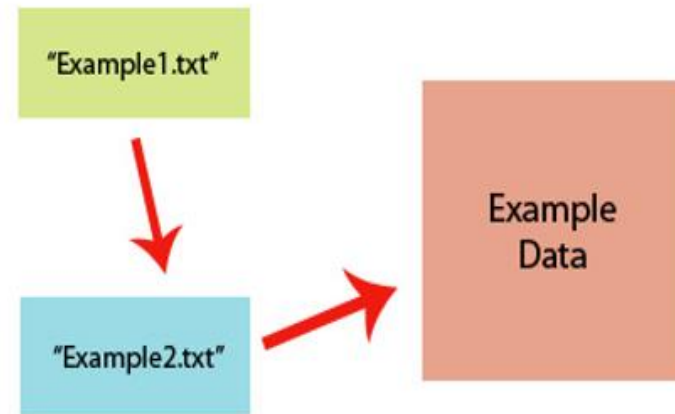
Use the "ln" command to create links

- Hard link



vs.

## Symbolic (soft) link



# Basic File Permissions

Each file has 9 permission bits, which are divided into 3 sets

(In order): Owner, Group, Others

Each permission set has 3 kinds of permissions

(In order): r (read), w (write), x (execute)

For each permission bit, “-” means denied

Example:

`rwxr-x---` The owner has all permissions, while the group does not have the write permission; all other users do not have any permission.

Each user fits into only one of the three permission sets. The permissions used are those that are most specific.

The “root” user has read and write permissions for all files, regardless how their permissions are set.

Permissions are not inheritable

# Permission Effects

Symbol	File	Directory	Symbolic Link
R	View content	Read file names only in the directory	Permissions are determined by its linked target file.
W	Change content	Create, delete, move, rename files if "x" is granted	
X	Execute binary or script files	Can be accessed, entered or passed through (cd), can access its files ( <u>inode</u> information)	

## Common directory settings

r-x: allows the content of the directory to be listed.

-wx: allows files to be created, deleted, and renamed within the directory.

# Permissions Representation

Octal Value	Binary Format	Symbolic Value	Permission
0	000	---	none
1	001	--x	execute only
2	010	-w-	write only
3	011	-wx	write and execute
4	100	r--	read only
5	101	r-x	read and execute
6	110	rw-	read and write
7	111	rw <del>x</del>	all granted

Octal values can also be used

rw~~x~~rw~~x~~r-x = 775

rw-rw-rw- = 666

**Table 6-2** Examples of numeric permission specifications

Mode	Meaning
777	Owner, group, and others can read, write, and execute file
755	Owner can read, write, and execute file; group and others can read and execute file
711	Owner can read, write, and execute file; group and others can execute file
644	Owner can read and write file; group and others can read file
640	Owner can read and write file, group can read file, and others cannot access file





# Default Permissions

Default permissions are set when a new file or directory is created

In Unix all base permissions begin as the following;

Directories [ Octal 777 / Binary 11111111 ]

Files [ Octal 666 / Binary 110110110 ]

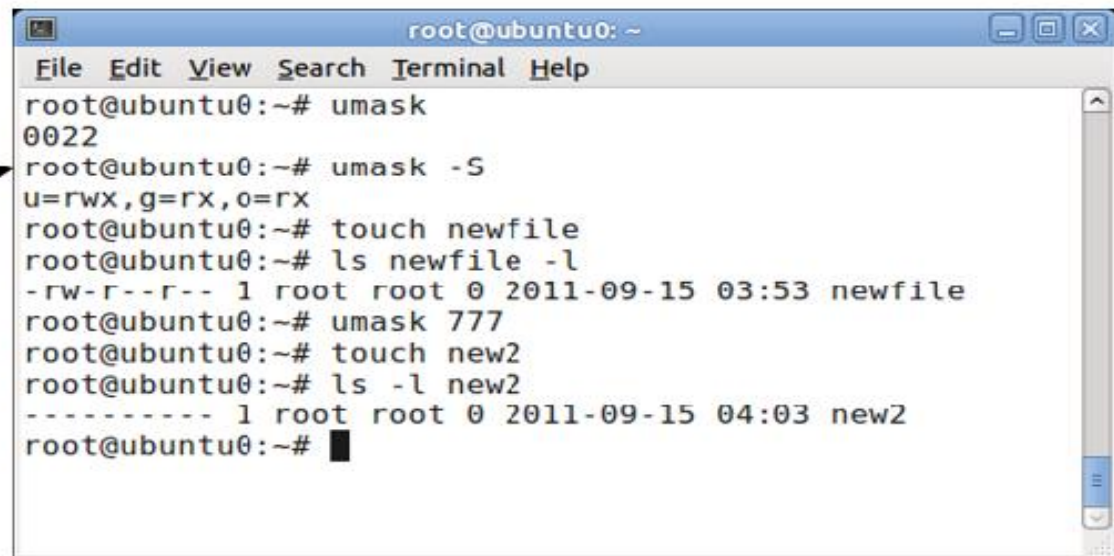
## Permission mask

Defines the permissions denied by default

For example: 022 (the default mask) means 2 (write permission) is restricted for group and others.

Use the “umask” command to view and set mask values.

Use "umask -S" to display symbolic output of the mask effect

A terminal window titled 'root@ubuntu0: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
root@ubuntu0:~# umask
0022
root@ubuntu0:~# umask -S
u=rwx,g=rx,o=rx
root@ubuntu0:~# touch newfile
root@ubuntu0:~# ls newfile -l
-rw-r--r-- 1 root root 0 2011-09-15 03:53 newfile
root@ubuntu0:~# umask 777
root@ubuntu0:~# touch new2
root@ubuntu0:~# ls -l new2
----- 1 root root 0 2011-09-15 04:03 new2
root@ubuntu0:~#
```

# Other File Attributes

## Timestamp

- Access time
- Modify time
- Change time

## File owner: user, group

## Number of links

For normal files, this shows the number of hard links

For directories, this shows the number of sub-directories (always including "." and "..")

# Directory/File Commands List

## Navigation

cd, pwd

## View directory/file content and properties

ls  
file, stat

## Modification (creation, change, deletion)

mkdir, cp, mv, rm, rmdir, ln

## Permissions

chmod, umask



# cd – Change (Current) Directory

## Use absolute path

Example:

```
cd /etc/
```

## Use relative path

Example:

```
cd ../bin
```

```
cd Documents/
```

```
cd ../../../ (.. Represents parent directory)
```

## Use home directory

Example:

```
cd
```

```
cd ~/Documents
```

# cd

The image shows a terminal window with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar (gzheng@ubuntu: /etc/firefox). The terminal output shows a series of 'cd' commands and their results. Annotations in blue boxes explain various aspects of the 'cd' command.

~ means the home directory for a user

Starting from the home directory: change to the "Documents" sub-directory. Note directory names are case sensitive. Notice the change of the command prompt on the next line.

See the command prompt to confirm directory changes

"/" means the root directory, the parent of all directories.

cd without any target will change to the home directory ~

.. means the parent directory, the immediate upper level directory.

Use relative path (starting from the current directory) to go to any level of sub-directory.

Use absolute path (always starting with "/") to go to any directory

The root directly is reserved for the "root" user.

```
gzheng@ubuntu:~$ cd Documents/
gzheng@ubuntu:~/Documents$ cd /
gzheng@ubuntu:/$ cd
gzheng@ubuntu:~$ cd ..
gzheng@ubuntu:/home$ cd ..
gzheng@ubuntu:/$ cd home/gzheng/Documents
gzheng@ubuntu:~/Documents$ cd /root/Desktop
-bash: cd: /root/Desktop: Permission denied
gzheng@ubuntu:~/Documents$ cd /etc/firefox
gzheng@ubuntu:/etc/firefox$
```

# ls – List Files (Contents)

## Basic usage

List files in the current directory: `ls`

List a specified file/directory: `ls [file/directory path]`

## Common options

- l: long listing (see next slide)
- i: view inode number
- h: human readable format for file size
- F: show file type sign
- a show all files (including hidden files)
- d display directories as ordinary files – do not show its content
- R Includes the contents of all levels of subdirectories.
- 1 1 item per line

Sorting:            -S: size,                      -t: time,                      -r: reverse order

## Reference

[Linux ls Command](#)



# ls

```
root@ubuntu: ~  
File Edit View Search Terminal Help  
root@ubuntu:~# ls  
Desktop Downloads Pictures Templates  
Documents Music Public Videos  
root@ubuntu:~# ls -F  
Desktop/ Downloads/ Pictures/ Templates/  
Documents/ Music/ Public/ Videos/  
root@ubuntu:~# ls -a  
.. Downloads .local Templates  
. .gconf .mozilla .themes  
.bashrc .gconfd Music .thumbnails  
.cache .gnome2 .nautilus Videos  
.config .gnome2_private Pictures .xsession-errors  
.dbus .gtk-bookmarks .profile .xsession-errors.old  
Desktop .gvfs Public  
.dmrc .ICEauthority .pulse  
Documents .icons .pulse-cookie  
root@ubuntu:~# ls -aF  
./ Downloads/ .local/ Templates/  
../ .gconf/ .mozilla/ .themes/  
.bashrc .gconfd/ Music/ .thumbnails/  
.cache/ .gnome2/ .nautilus/ Videos/  
.config/ .gnome2_private/ Pictures/ .xsession-errors  
.dbus/ .gtk-bookmarks .profile .xsession-errors.old  
Desktop/ .gvfs/ Public/  
.dmrc .ICEauthority .pulse/  
Documents/ .icons/ .pulse-cookie  
root@ubuntu:~# ls -l  
total 32  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Desktop  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Documents  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Downloads  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Music  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Pictures  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Public  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Templates  
drwxr-xr-x 2 root root 4096 2012-01-06 11:24 Videos  
root@ubuntu:~#
```

The ls command displays all files and sub-directories

-F option will show file type symbol: "/" means this is a directory.

-a option will show all files: hidden files which start with a dot .

Two options combined to show all files with file type symbol

-l option will show files in a long format.

# ls -l Output

## File type by file colors

Black: normal file  
Blue: directory  
Green: executable file  
Cyan: link

- Long listing:

File type	Symbol	Type of file	File access permissions	ACL flag	Links	Owner	Group	Size	Date and time of modification	Filename
Regular file	-									
Directory	d									
Character device file	c									
Block device file	b									
Local domain socket	s									
Named pipe	p									
Symbolic link	l									

-rwxrwxr-x+	3	max	pubs	2048	2010-08-12	13:15	memo
-------------	---	-----	------	------	------------	-------	------

**Figure 6-12** The columns displayed by the `ls -l` command



# ls and Directories

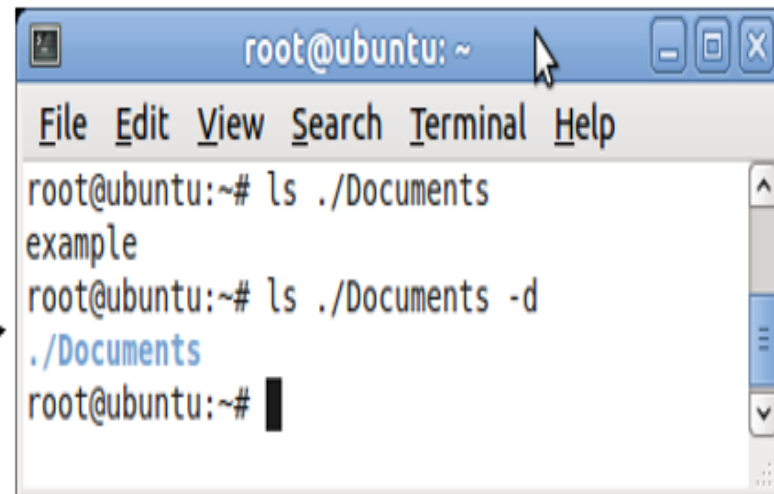
## -d option

If an argument is a directory, it only lists its name not its contents (files in the directory).

## -R option

Display the contents of all levels of subdirectories.

Note with the -d option, the file "example" is not displayed. Compare the two command effect.



```
root@ubuntu: ~  
File Edit View Search Terminal Help  
root@ubuntu:~# ls ./Documents  
example  
root@ubuntu:~# ls ./Documents -d  
./Documents  
root@ubuntu:~#
```

The terminal window shows the user running two commands to list the contents of the ./Documents directory. The first command, `ls ./Documents`, displays the file `example`. The second command, `ls ./Documents -d`, displays only the directory name `./Documents` and does not show its contents. A blue box on the left contains a note explaining that the -d option prevents the contents of the directory from being listed.

# Pathname Expansion

`*`, `?` and `[]` can be used for matching file names for commands that take multiple file names, such as `ls`, `cp`, `rm`, `mv`, `file`, etc.

`*`: any character

`?`: any one character

`[]`: any one character in a defined range

Hidden files not included unless starting the pattern with `"."`

## Examples

`ls /etc/*.conf` list all files ending with `.conf` in the `/etc/` directory

`ls .*` list all hidden files in the current directory

`rm file?` Remove all files starting with `"file"` and ending with any one character

`cp ../file[0-9]` copy `file0`, `file1`, ..., `file9` (10 files) in the parent directory to the current directory

# More ls Examples

## Use . \* / in ls arguments

- . current directory, or start of a hidden file
- \* wildcard file name expansion
- / directory

## Combination examples

ls ./\* list all files in the current directory and sub-

ls /\* list all hidden files in the root directory, and  
subdirectory under the

root directory

ls \*/ list all hidden sub-directories

directories  
files in the hidden

# View File Metadata

## file

This command describes the type of file(s).

file examplefile

file \*

## stat

This command display detailed information of a file

stat examplefile

# Copy/Move Files and Directories

## cp

Copy files/directories

cp [source] [destination] (Note: default is to overwrite)

cp ../file1 file2 //copy file1 from the parent directory to current directory  
as file2 (renamed)

cp ../file1 . //copy file1 from the parent directory to current directory  
as file1 (not renamed)

-i prompt for overwrite choices

-r copy all files from a directory and its sub-directories

## mv

Move (rename) files/directories

mv [source] [destination] (Note: default is to overwrite)

mv ../file1 /root/Documents

mv file1 file2 (rename if in the same directory)

mv files1 ../file2 (move to the parent directory and rename)

-i prompt for overwrite choices

# Create and Delete Directory

## mkdir

Create directory

mkdir *[directory name]*

Options

-p            No error if existing, make parent directories as needed.

## rmdir

Delete directory

rmdir *[directory name]*

Option

-p            Remove DIRECTORY and its ancestors.

E.g., `rmdir -p a/b/c' is similar to `rmdir a/b/c a/b a'.

# Removing Directory and File

**rm**

Deletes a file without confirmation (by default).

## Options

- i: interactive. Prompted for confirmation of deletion
- r: recursive. Delete all file and sub-directories.

[Alert!](#)

# ln

## Create a hard link

ln [target file] [link name]

Example:

ln Documents/file1 linka

## Create a soft link -s

Example:

ln -s Documents/file1 linka



# chmod

## Syntax

chmod [permission setting] files

## Permission settings

+	add permission
-	remove permission
=	directly set permission
a	all
u	user (owner)
g	group
o	others

## Examples

u+w

Adds write permission for the owner of the file

ug=rw,o=r

Gives r/w permission to owner and group, and read permission to others

a-x

Removes execute permission for all categories (owner/group/other)

g=u

Makes the group permissions be the same as the owner permissions

# Summary

## Key concepts

File system

inode

File, directory, working directory

Path, absolute path, relative path

Link, symbolic link, hard link

File permission symbols, octal number meanings

## Key practices

Use the following commands for common file system operations

Navigate and view files and directories: ls, cd, pwd

Manipulate files: mkdir, cp, mv, rm, rmdir, ln

View file attributes: file, stat, (ls)

Permission: chmod, umask

# Good Resrouces

[Directory paths](#)

[Unix LS Command: 15 Practical Examples](#)

[File permissions](#)

[Linux Directory Permission Confusion](#)

[Fixing Unix/Linux/POSIX Filenames](#)



# File protection

=> Alias rm / mv / cp command as rm -i / mv -i  
alias cp='cp -i'  
alias mv='mv -i'  
alias rm='rm -i'

=> Make important file copy before editing

## [How to Keep Files Safe from Accidental Overwriting](#)

<http://www.cyberciti.biz/faq/how-to-make-a-file-unchangeable-unalterable-so-that-no-one-can-modify-it/>

Chattr

Lsattr

- accidental operations <http://www.cyberciti.biz/faq/unix-linux-undelete-unerase-command/>