CHALMERS

GÖTEBORGS UNIVERSITET

Introduction to Linux

Lecture 2

Prof. Dr. Philippe Tassin
Chalmers University of Technology

Dr. Evangelos Siminos
Chalmers University of Technology and University of Gothenburg

Göteborg, 3 April 2019

Files

- “Everything in Linux is a file; if it is not a file then it is a process.”

- Several types of files:
  - Regular files
  - Directories
  - Links
  - Named pipes (will discuss when we discuss Bash programming)
  - Sockets (for network communication)

- Files names can contain any almost any character (no /), but some characters have special meaning in Bash and must be escaped or quoted
  - cat “this is a file name with spaces”
  - cat *

Partitions and mount points

- There are often several partitions on the hard disk
- Why? Advantages/disadvantages

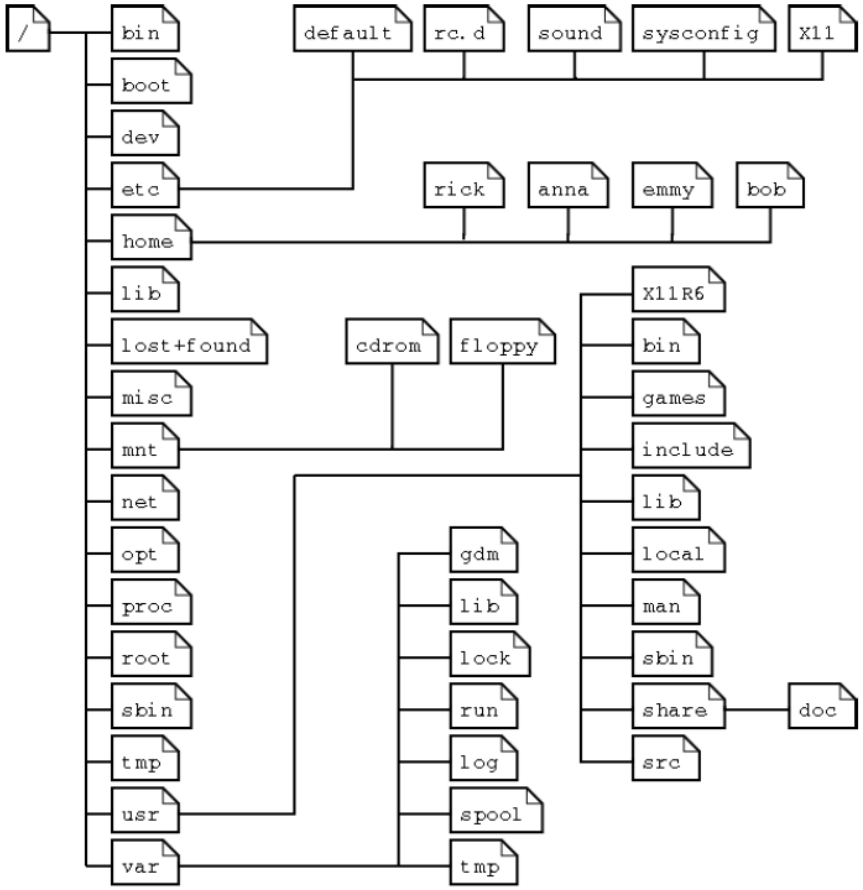
- Data partitions vs. swap partitions

- All partitions are mounted in the same file structure in Linux
  - /boot is often a separate partition
  - /tmp

- The directories that are the top of another file system are called mount points

File system layout

- The file system is like a tree



- The tree-like structure is established by two special files: . and ..

Absolute and relative paths

Files can be referred to in two ways:

- Absolute path
  - Path to a file starting from the top of the tree
  - Starts with /
- Relative path
  - Is relative to the current directory or to a special directory (~)
  - Does not start with /

Peripheral devices

- Peripheral devices are also represented by files
- In /dev, but file systems on peripheral devices can be mounted anywhere in the file system

Name	Device
`cdrom`	CD drive
`console`	Special entry for the currently used console.
`cua*`	Serial ports
`dsp*`	Devices for sampling and recording
`fd*`	Entries for most kinds of floppy drives, the default is `/dev/fd0`, a floppy drive for 1.44 MB floppies.
`hd[a-t][1-16]`	Standard support for IDE drives with maximum amount of partitions each.
`ir*`	Infrared devices
`isdn*`	Management of ISDN connections
`js*`	Joystick(s)
`lp*`	Printers
`mem`	Memory
`midi*`	midi player
`mixer*` and `music`	Idealized model of a mixer (combines or adds signals)
`modem`	Modem
`mouse` (also msmouse, logimouse, psmouse, input/mice, psaux)	All kinds of mouses
`null`	Bottomless garbage can
`par*`	Entries for parallel port support
`pty*`	Pseudo terminals
`radio*`	For Radio Amateurs (HAMs).
`ram*`	boot device
`sd*`	SCSI disks with their partitions
`sequencer`	For audio applications using the synthesizer features of the sound card (MIDI-device controller)
`tty*`	Virtual consoles simulating vt100 terminals.
`usb*`	USB card and scanner
`video*`	For use with a graphics card supporting video.

(Try inserting a USB stick and find where the stick gets mounted.)

Viewing file properties and contents

- A file contains
  - Data
  - Metadata (type of file; file permissions; time stamp; etc)

- ls -la : show metadata
- `file`: show data type

```
[ptassin@aphy1 linuxcourse]$ ls -la
total 9
drwxr-xr-x+  4 ptassin localusers  6 Mar 26 22:10 .
drwx------+ 29 ptassin localusers 38 Mar 26 10:20 ..
-rw-r--r--+  2 ptassin localusers  9 Mar 18 20:15 coursename
-rw-r--r--+  2 ptassin localusers  9 Mar 18 20:15 coursename2
drwxr-xr-x+  4 ptassin localusers  5 Mar 18 20:15 notes
drwxr-xr-x+  2 ptassin localusers  4 Mar 18 20:18 tasks
[ptassin@aphy1 linuxcourse]$ file coursename
coursename: ASCII text
```

- Content:
  - `cat`: prints content to standard output
  - `less`: pager

Creating and removing files and directories

- `touch`: update file's timestamp or create empty file
- `cp source target`
- `mv source target`
- `rm`: remove file
- `rmdir`: remove directory

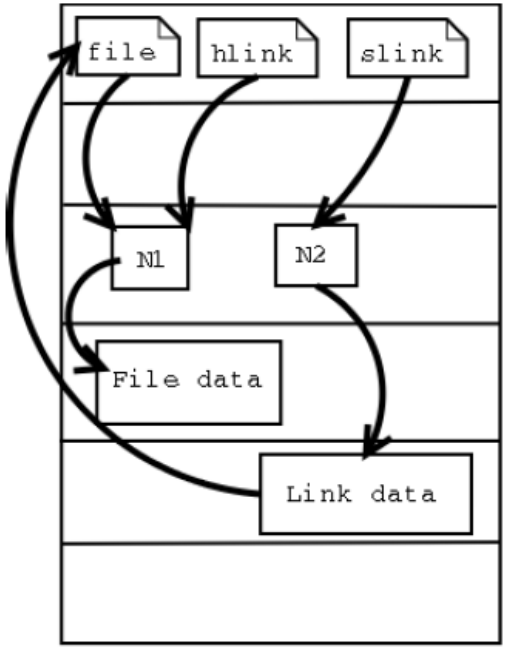
Finding and filtering files

- `ls`: can be used to find files, but you have to know (approximately) where the file is
- `locate`: fast method to find files based on an index
- `find`: slow method to find files by actually searching the file system
- `grep`: is a line filtering tool allowing to extract from a text files lines that match certain conditions

Hard links and symbolic links

- The files on a hard disk are actually stored in a way that doesn't look like a hierarchical tree; but each "file" is an inode
- Multiple file names can be associated with a single inode

  -> hard links
- Symbolic links: each symbolic link is an inode with a pointer to another file name



File permissions

- Every file has access modes
- Three groups: user, group, others
- Three permission bits: read (4), write (2); execute (1)
  - E.g.: full control = 7 = 4 + 2 + 1
  - read+write access = 6 = 4 + 2
  - `chmod 660 coursename`
  - `chmod u+rwx,go-rwx`
- For directories:
  - read: can list the files in the directory
  - write: can add/remove/rename files in the directory
  - execute: can cd to the directory

Special modes

- Sticky bit:
  - Files: if file is loaded in memory, it remains loaded after the process is finished (ignored by Linux kernel)
  - Directories: files in the directory may only be renamed or unlinked by the owner

- SUID/SGID on files: executable with SUID/GUID bit set runs with access permissions of the owner user/group (instead of the user running the executable)

- SGID on directories: new files get the same group owner as the directory

Any other questions?