

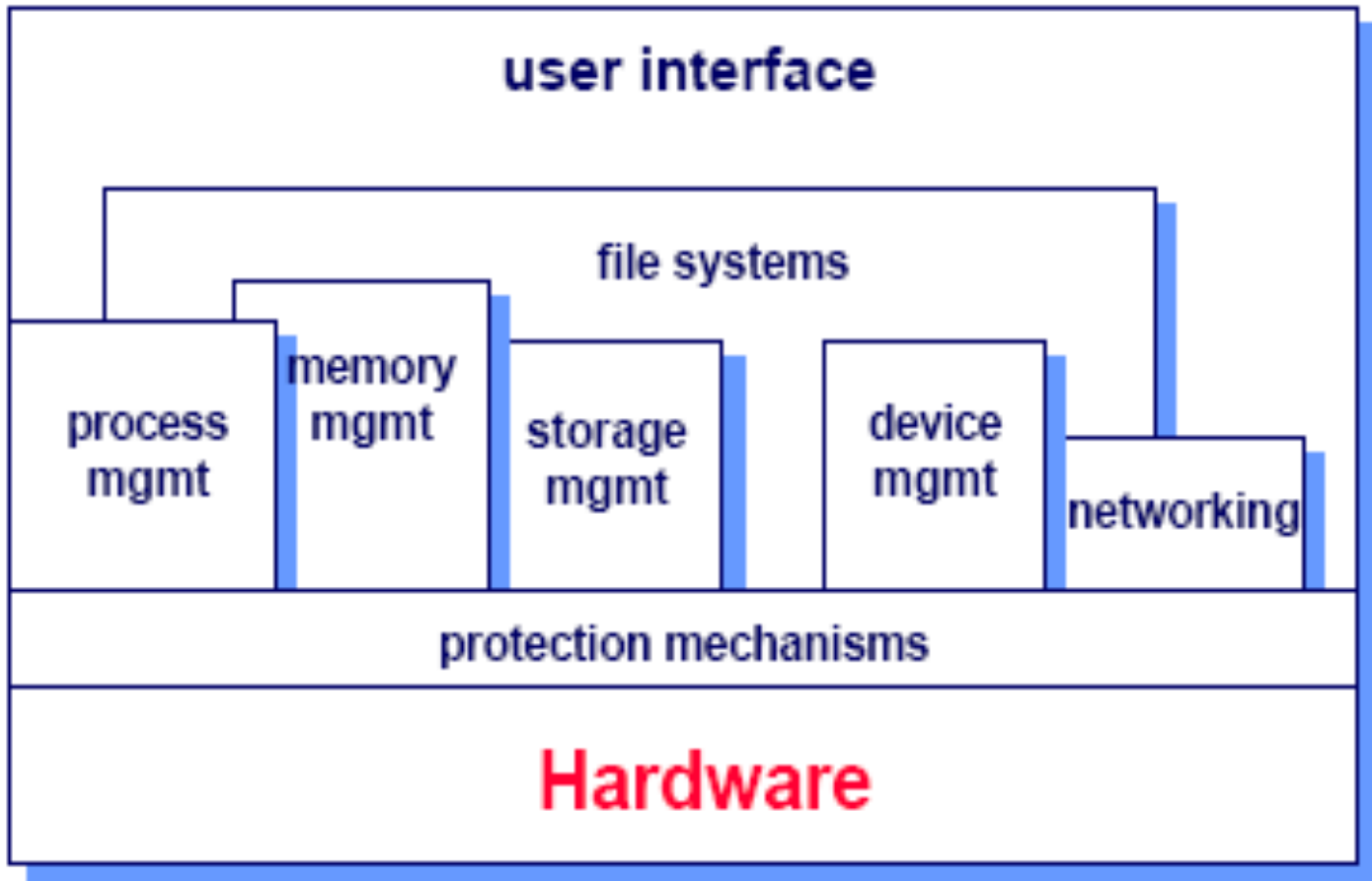
Operating system

- An operating system is the program that controls all the other parts of a computer system - both the hardware and the software.
- An operating system provides orderly and controlled allocation and use (i.e., *sharing*) of the resources by the users (jobs) that compete for them.
- One major function of an operating system is to “hide” the complexity of the underlying hardware and give the *user* a better view (an abstraction) of the computer.


Operating system components

- An operating system generally consists of the following components:
 - Process management
 - (Disk) storage management
 - Memory management
 - I/O (device) management
 - File systems
 - Networking
 - Protection
 - User Interface

OS architecture



OS kernel

- During ‘ ‘normal” operations of a computer system, some portions of the operating system remain in main memory to provide services for critical operations, such as dispatching, interrupt handling, or managing (critical) resources.
- These portions of the OS are collectively called the *kernel*.
- Kernel = OS – transient components

remains *comes and goes*

Bootstrapping

- The process of initializing the computer and loading the operating system is known as bootstrapping.
- This usually occurs when the computer is powered up or reset.
- The initial loading is done by a small program that usually resides in non-volatile memory (e.g., EPROM).
- This in turn loads the OS from an external device.
- Once loaded, how does the operating system know what to do next?
- It waits for some event to occur: e.g., the user typing a command on the keyboard.

History of the UNIX operating system

- The first version of UNIX was created in 1969 by Kenneth Thompson and Dennis Ritchie, system engineers at AT&T's Bell Labs.
- It gained popularity in 1977, when it was first made commercially available by Interactive Systems Corporation.
- A team from the University of California Berkeley was working to improve UNIX the same time. In 1977 it released the first Berkeley Software Distribution, which became known as **BSD**.
- By 1983 commercial interest was growing and Sun Microsystems produced a UNIX workstation called Solaris.

History of the UNIX operating system

- **System V** appeared, directly descended from the original AT&T UNIX and the prototype of the more widely used variant today.

Modern variants of UNIX/Linux

- Though there are several standards, there are three main versions of Linux in use today: **System V** and **BSD** and **POSIX**
- System V is the more popular of the two and is very similar to BSD
- POSIX standards by IEEE are more commonly accepted uniform standards. POSIX documentation has three parts:
 - 1a → for OS
 - 1b → for RTOS
 - 1c → for threads

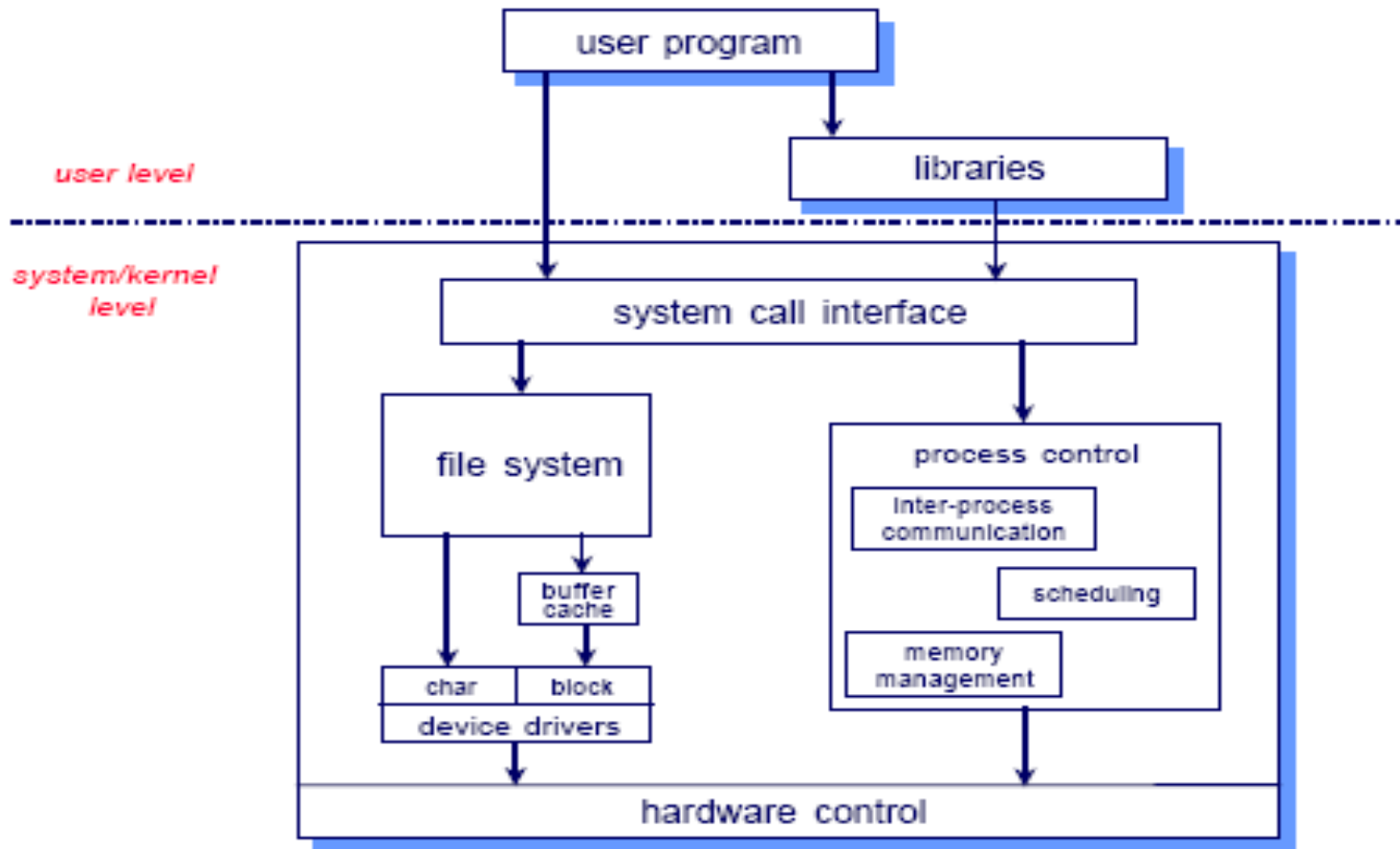
Advantages of Linux OS

- Linux source code is freely distributed
- Linux is Multi-user.
- Linux runs on a wide range of hardware
- Linux is exceptionally stable
- Linux has the tools and applications you need
- Linux is one of the most secure operating systems.

Linux operating system

- The Linux operating system has three important features; a kernel, the shell and a file system.
- The Kernel
- Linux file system
- The Shell

Linux system/kernel structure



Accessing OS services

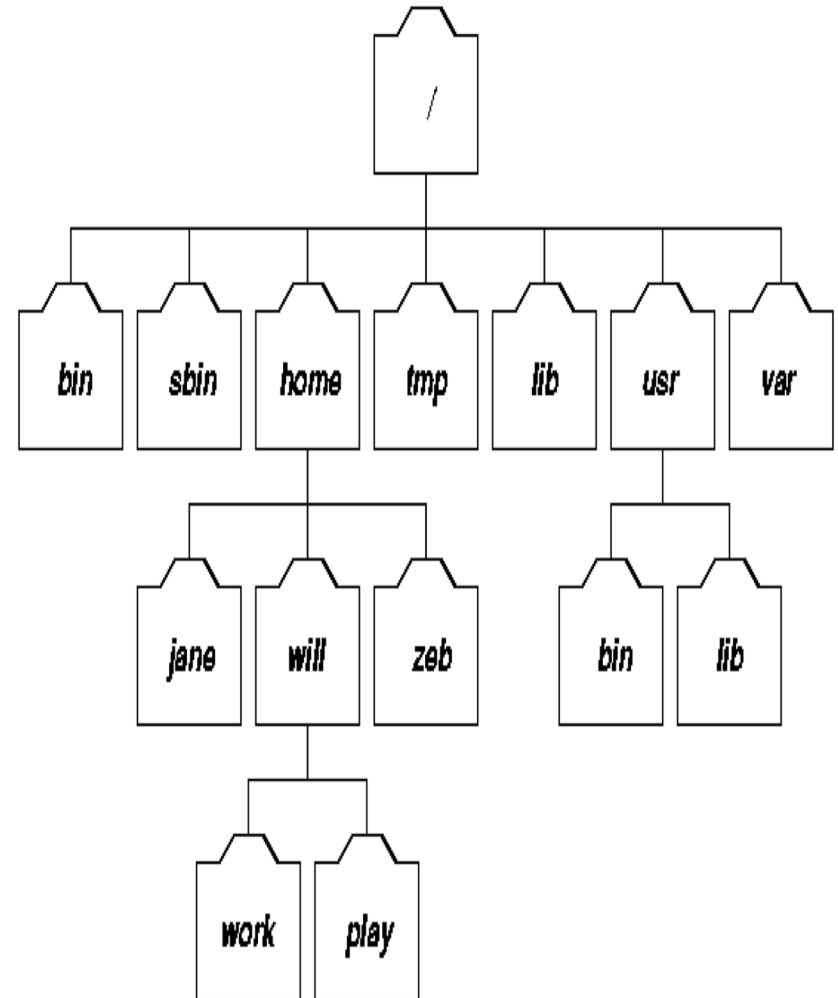
- The mechanism used to provide access to OS services (i.e., enter the operating system and perform a “privileged operation’ ’) is commonly known as a *system call*.
- The (only) difference between a ‘ ‘procedure call’ ’ and a “system call’ ’ is that a system call changes the execution mode of the CPU (to *supervisor mode*) whereas a procedure call does not.
- *System call interface*: A set of functions that are called by (user) programs to perform specific tasks.

Linux file system

- *"On a Linux system, everything is a file; if something is not a file, it is a process."*
- A file system is a logical method for organizing and storing large amounts of information in a way which makes it easy manage.
- The file is the smallest unit in which information is stored.
- The file system used by linux is **ext3** or **ext4**
- The linux file system is also called as **inverted tree structure**

Linux FILE SYSTEM HIERARCHY

- The Linux file system is laid out as a hierarchical tree structure which is anchored at a special top-level directory known as the root .
- A directory can have many child directories, but only one parent directory.



Linux file system

- Sorts of files
 - REGULAR Files: Most files contain normal data, for example text files, executable files.
 - Directories: files that are lists of other files.
 - Special files: the mechanism used for input and output. Most special files are in /dev.
 - Links: a system to make a file or directory visible in multiple parts of the system's file tree.
 - (Domain) sockets: a special file type, similar to TCP/IP sockets, providing inter-process networking
 - Named pipes: act more or less like sockets and form a way for processes to communicate with each other.

Linux file system

- In Linux file system, a file is represented by an *inode*
- At the time a new file is created, it gets a free inode.
- The following information in that inode :
 - Owner and group owner of the file.
 - File type (regular, directory, ...)
 - Permissions on the file
 - Date and time of creation, last read and change.
 - Date and time this information has been changed in the inode.
 - Number of links to this file
 - File size
 - An address defining the actual location of the file data.

FILE DESCRIPTOR

- A file descriptor is a non-negative integer created when a file is opened.
- File descriptors provide a primitive, low-level interface to input and output operations.
- A new descriptor is created each time the file is opened.
- Opens always choose the lowest-numbered file descriptor available.
- File descriptors are retired when the file is closed or the process terminates.

FILE DESCRIPTOR

- It is associated with a file object which includes information such as the mode in which the file was opened and the *offset pointer* where the next operation will begin.
- This information is called the *context* of the file.
- The first three slots are generally allocated to the standard streams
 - 0-standard input
 - 1-standard output
 - 2-standard error

The Shell

- Whenever you login to a Linux system you are placed in a program called the shell.
- On most Linux systems a program called bash acts as the shell program.
- bash stands for Bourne Again SHell, an enhanced version of the original Bourne shell program, sh, written by Steve Bourne
- There are several additional shell programs available on a typical Linux system.
- These include: ksh, tcsh and zsh.

PROCESS

- A process is a running instance of a program.
- A process is said to be born when starts executing and remains alive as long as the program is active.
- As files, processes do have attributes,
 - PID: Process ID
 - PPID: Parent process ID
 - TTY: Terminal on which the process is connected

The Shell

- The Shell is the program that directly executes your commands.
- The shell acts as a command interpreter.

Activities

- The shell issues a prompt and waits for command.
- The shell scans for meta-character and expands them.
- It then passes command to kernel.
- Waits till the command is executed by the kernel and reappears again.

UNIX COMMAND SET

- GENERAL FORMAT

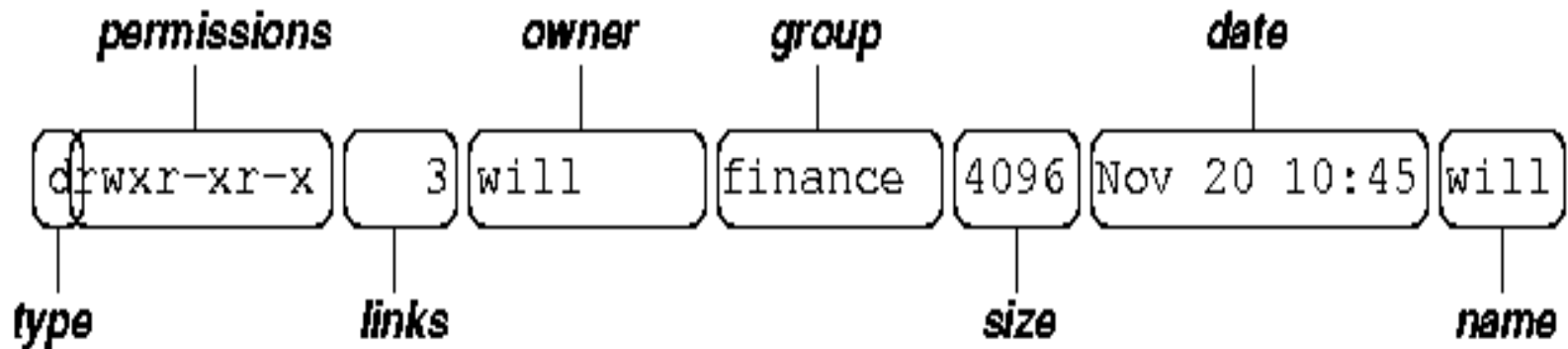
\$ command -options targets

GENERAL COMMANDS

- date
- cal
- echo
- clear
- exit

DIRECTORY AND FILE HANDLING COMMANDS

- pwd (print [current] working directory)
- ls (list directory)
 - \$ ls -a
 - \$ ls -a -l
 - \$ ls -al
- Each line of the output looks like this:



DIRECTORY AND FILE HANDLING COMMANDS

File types in a long list

<i>Symbol</i>	<i>Meaning</i>
• -:	Regular file
• d:	Directory
• l:	Link
• c:	Special file
• s:	Socket
• p:	Named pipe
• b:	Block device

Getting Help

- The command **man** gives you access to an on-line manual containing a complete description of every command available on this system.
- **man** can also provide you with one line descriptions of commands specified by name; or for all commands whose description contains any of a set of keywords.

Getting Help

- The manual is organised into several sections (pages) each of which covers a different area.
- Important pages(sections) are:
 - 1 for commands
 - 2 for system calls
 - 3 for functions
- On some systems the command gives the relevant info
`man section_number <command/function>`

DIRECTORY AND FILE HANDLING COMMANDS

cd (change [current working] directory)

- \$ cd *path*

mkdir (make directory)

- \$ mkdir *directory*

rmdir (remove directory)

- \$ rmdir *directory*

cp (copy)

- \$ cp *source-file(s) destination*

rm (remove/delete)

- \$ rm *target-file(s)*

Process Control Commands

- There are several commands that can be used to control processes. They are:
- `ps` - list the processes running on the system
- `kill` - send a signal to one or more processes (usually to "kill" a process)
- `jobs` - a way of listing processes in the current terminal
- `bg` - put a process in the background (using job id)
- `fg` - put a process in the foreground. (using job id)