Unix Operating System

- Unix is considered one of the greatest achievements in computer science
- Designed by programmers for programmers
- Unix is a true multi-user operating system
- The primary user interface with Unix is through a command line interface (terminal console)
- It has several GUI interfaces available, all built on X-windows
- Integrated networking capabilities
- Attributes: stability, portability, security

Lineage

- Unix was conceived at AT&T Bell Labs by Dennis Ritchie and Ken Thompson in the late 60’s
- The C language was developed shortly thereafter to support Unix
- UC Berkeley created a new variant that included networking in the late 70’s, known as BSD
- In 80’s the GNU free software movement begins
- In early 90’s Linux was developed as a Unix look-a-like and several BSD based systems appeared

GNU/Linux

The goal of GNU (GNU’s Not Unix):

“To create complete UNIX-compatible software systems entirely composed of free software.”

Richard Stallman

- Unix-like but no unix code (hence GNU).

- The movement created many popular tools (emacs, gcc, gdb...).

Unix vrs. Linux

Commanlities

- Linux is Unix clone
- POSIX compliant

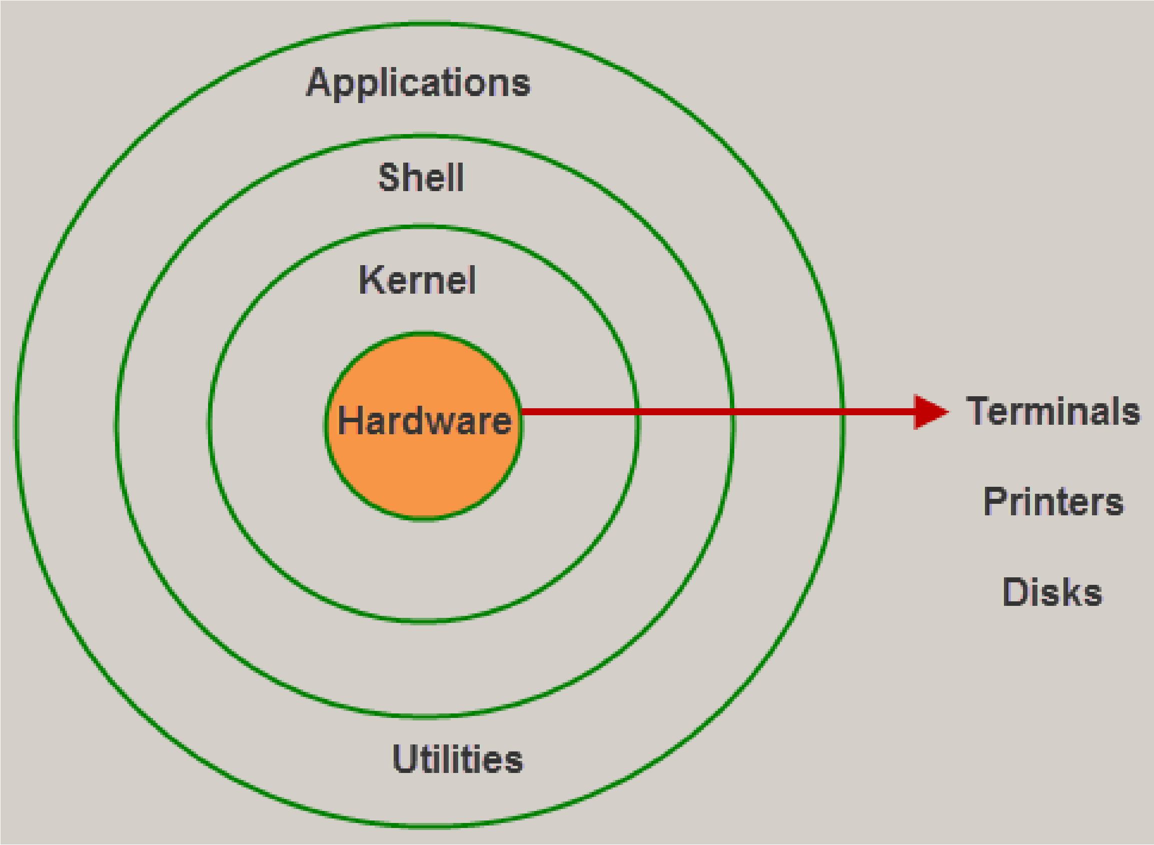
Differences

- Source code
- Linux kernel + GNU utilities, management tools, compilers, editors, applications (openOffice, FireFox) – all from different source
- Open source/Propriety
- Installation (flexibility and cost)
- Command Line and GUI interfaces
- Distributions

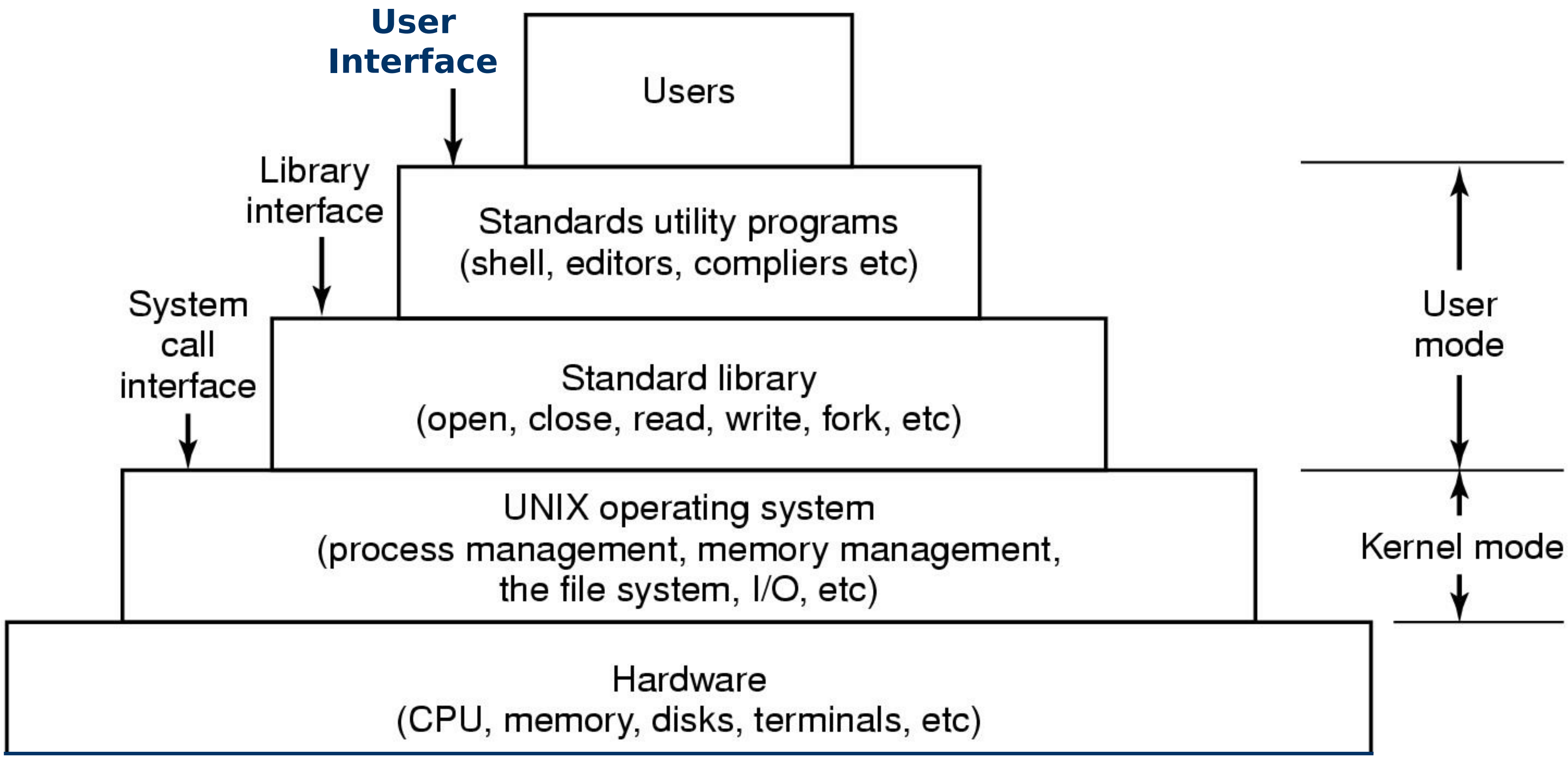
Linux: Ubuntu, Redhat, Suse,Debian, Fedora

Unix: BSD (MacOS), HP-UX, IRIX, IBM-AIX, SUN-Solaris

Elements of Unix



UNIX



User Interface

Library interface

System call interface

Users

Standards utility programs
(shell, editors, compliers etc)

Standard library
(open, close, read, write, fork, etc)

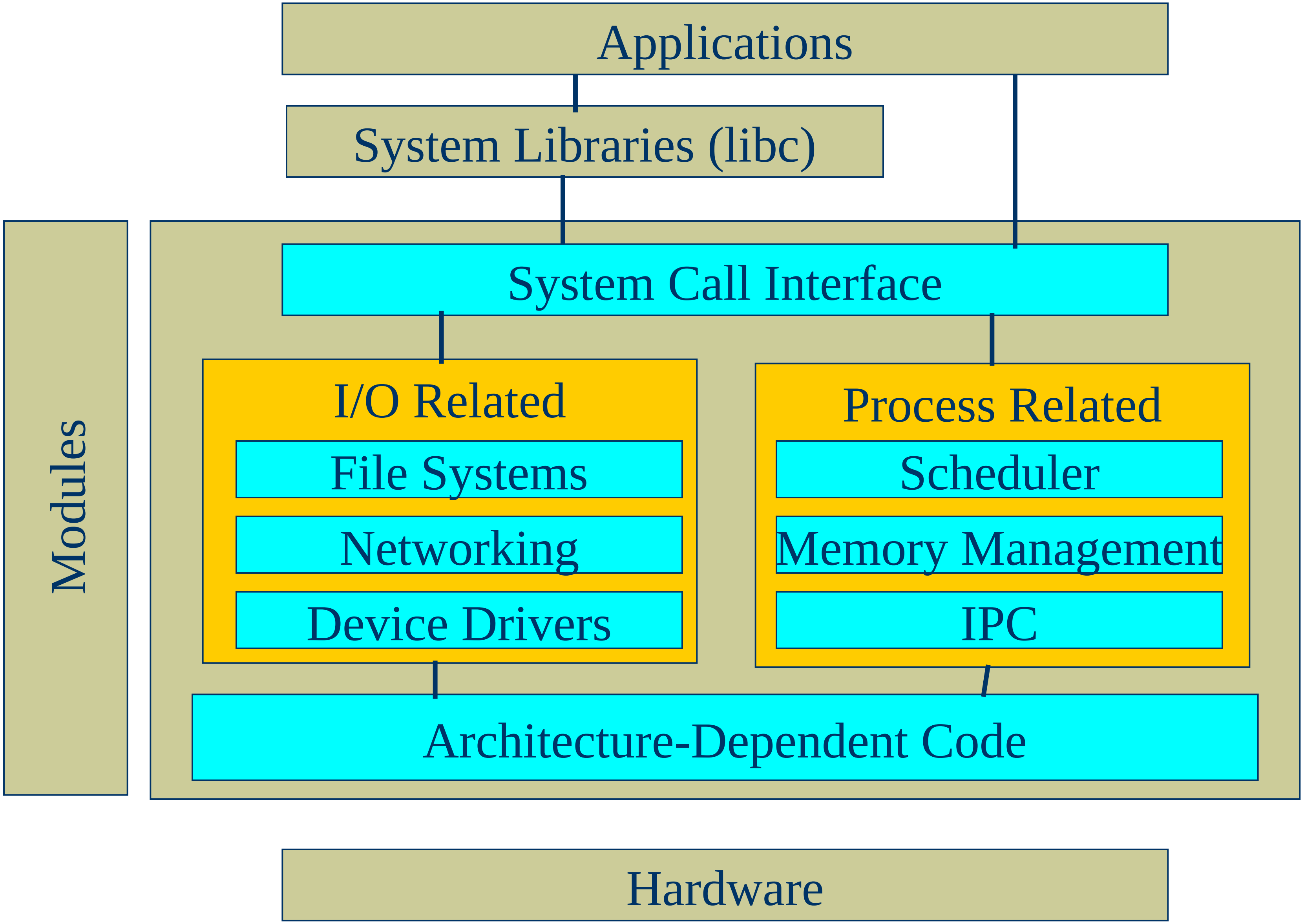
UNIX operating system
(process management, memory management,
the file system, I/O, etc)

Hardware
(CPU, memory, disks, terminals, etc)

User mode

Kernel mode

Essential Unix Architecture



Applications

System Libraries (libc)

Modules

System Call Interface

I/O Related

File Systems

Networking

Device Drivers

Process Related

Scheduler

Memory Management

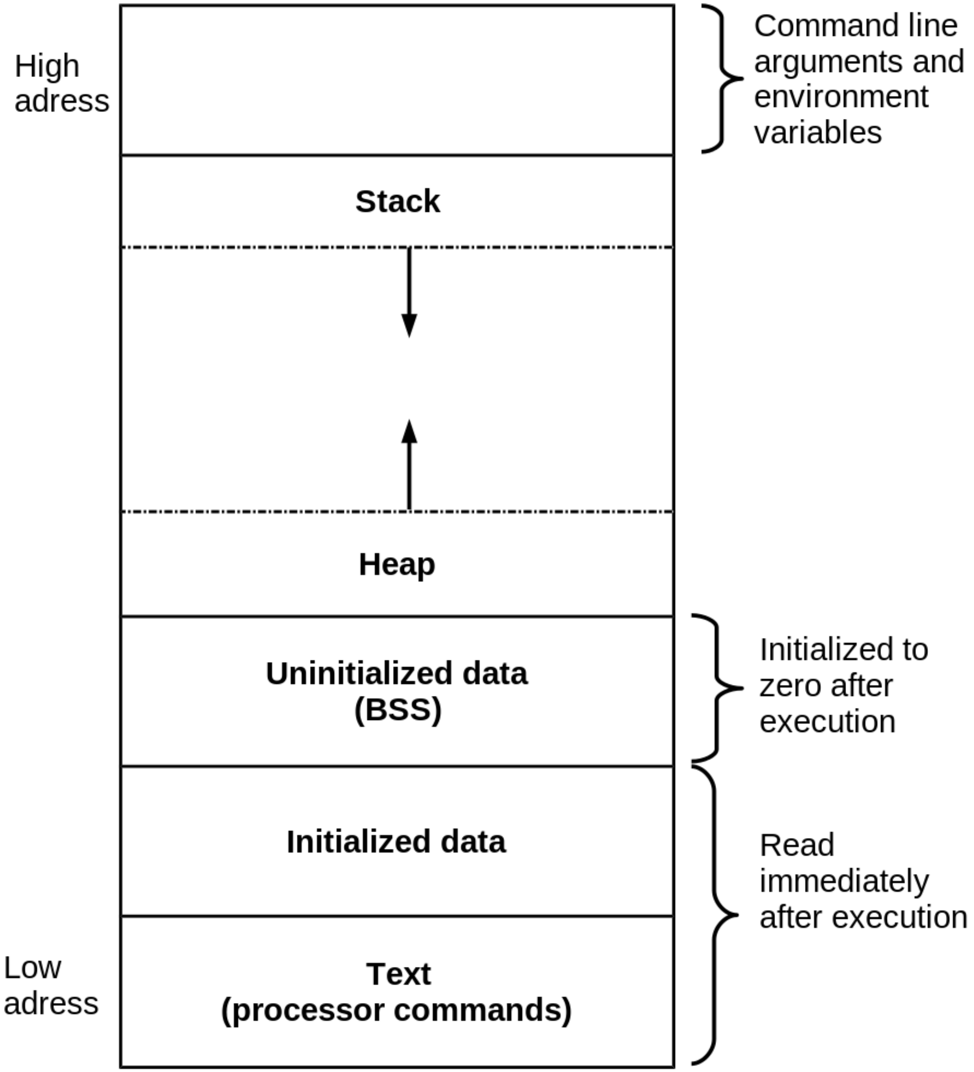
IPC

Architecture-Dependent Code

Hardware

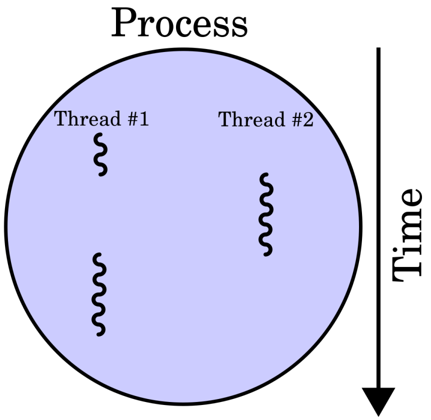
Processes

- A process is a “unit” of executable code
- Everything that runs in a Unix system occurs within the context of a process
- Each process created is assigned a unique id
- Only one process/CPU can be executing at any time, each process receives “fair” access to the CPU
- Switching between processes is a context change

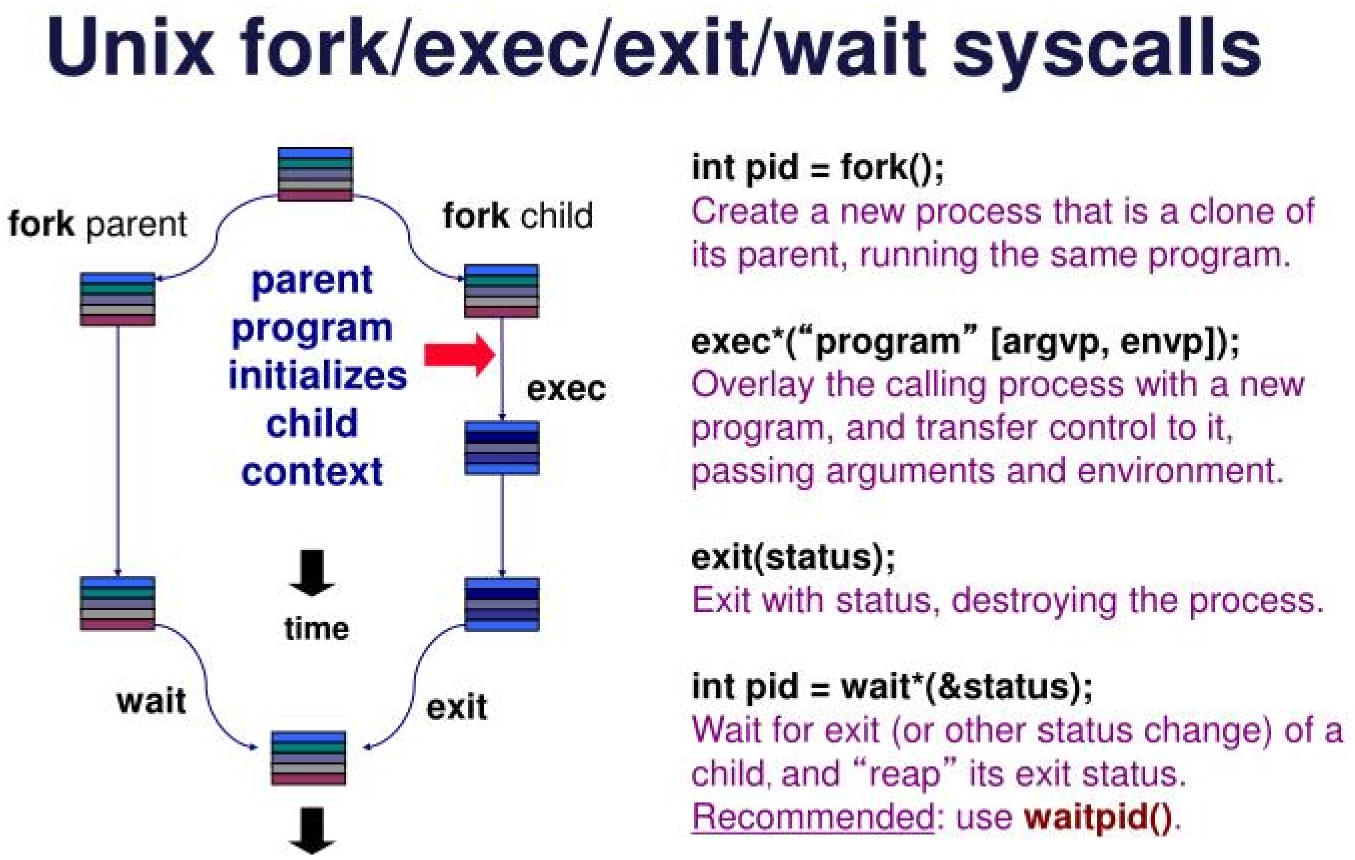

High adress
Command line arguments and environment variables
Stack
Heap
Uninitialized data (BSS)
Initialized to zero after execution
Initialized data
Read immediately after execution
Text (processor commands)
Low adress

Processes - Threads

- A thread is also called a light-weight process, a thread executes within the context of its parent process
- A thread is one of many execution streams sharing the same address space of a process
- Switching between threads does not require a context change
- Using threads requires careful programming
- In Linux each thread uses an entry in the process table but they are not full fledged processes



Processes – system calls



Unix fork/exec/exit/wait syscalls

int pid = fork();
Create a new process that is a clone of its parent, running the same program.

exec*("program" [argvp, envp]);
Overlay the calling process with a new program, and transfer control to it, passing arguments and environment.

exit(status);
Exit with status, destroying the process.

int pid = wait*(&status);
Wait for exit (or other status change) of a child, and "reap" its exit status.
Recommended: use **waitpid()**.

Fork System Call

- Current process split into 2 processes: parent, child
- Returns -1 if unsuccessful
- Returns 0 in the child
- Returns the child's identifier in the parent

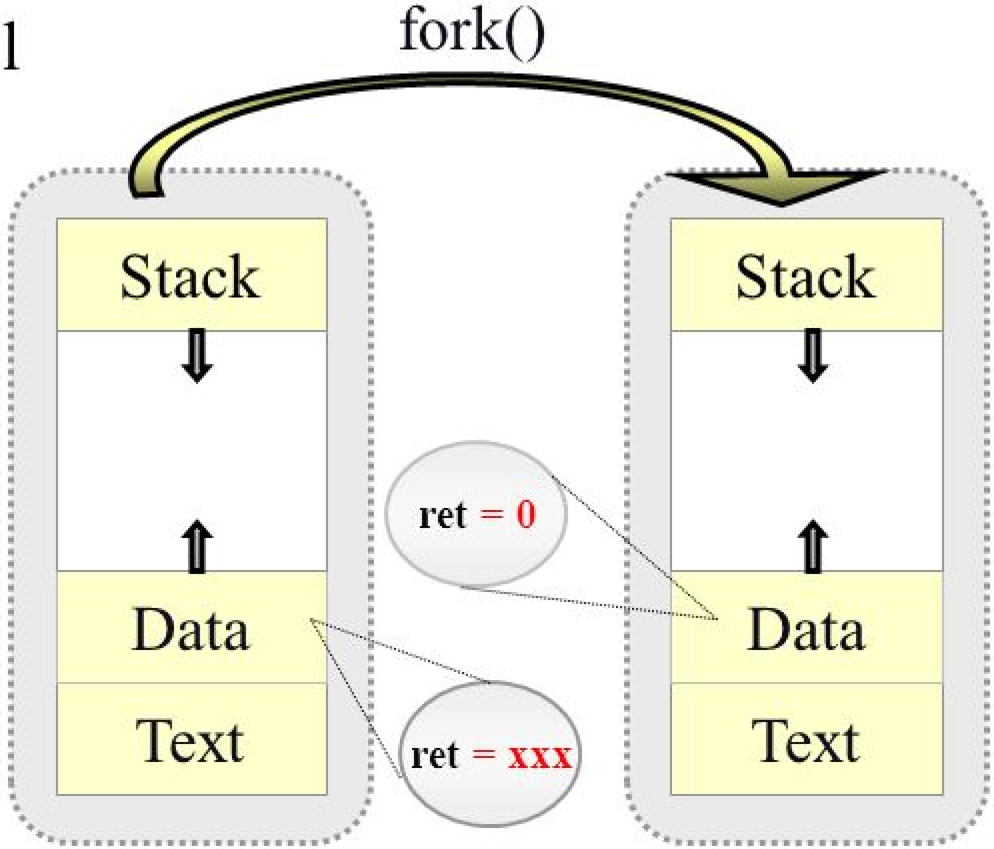
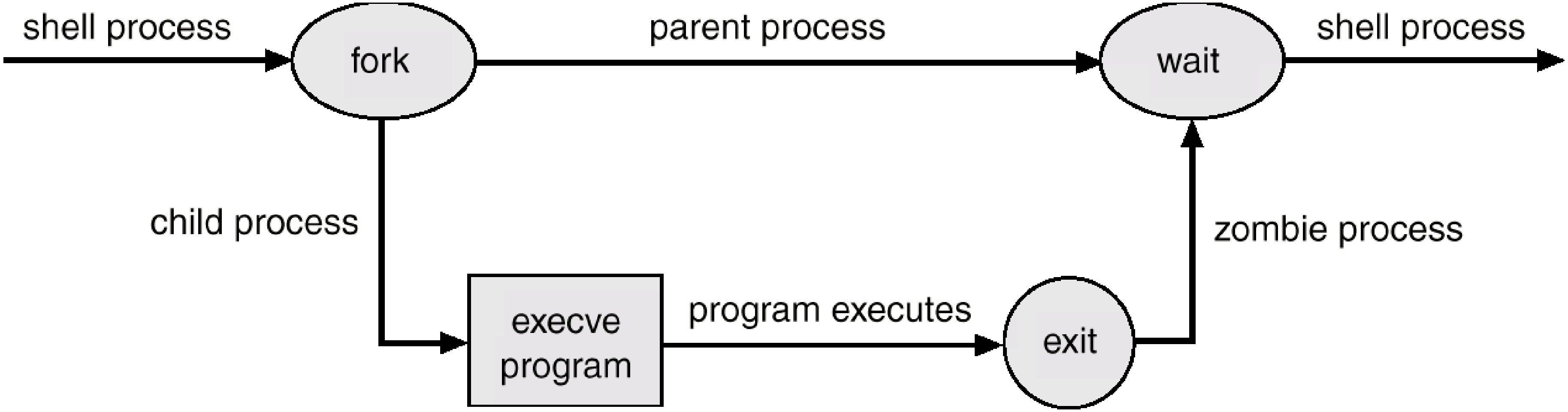


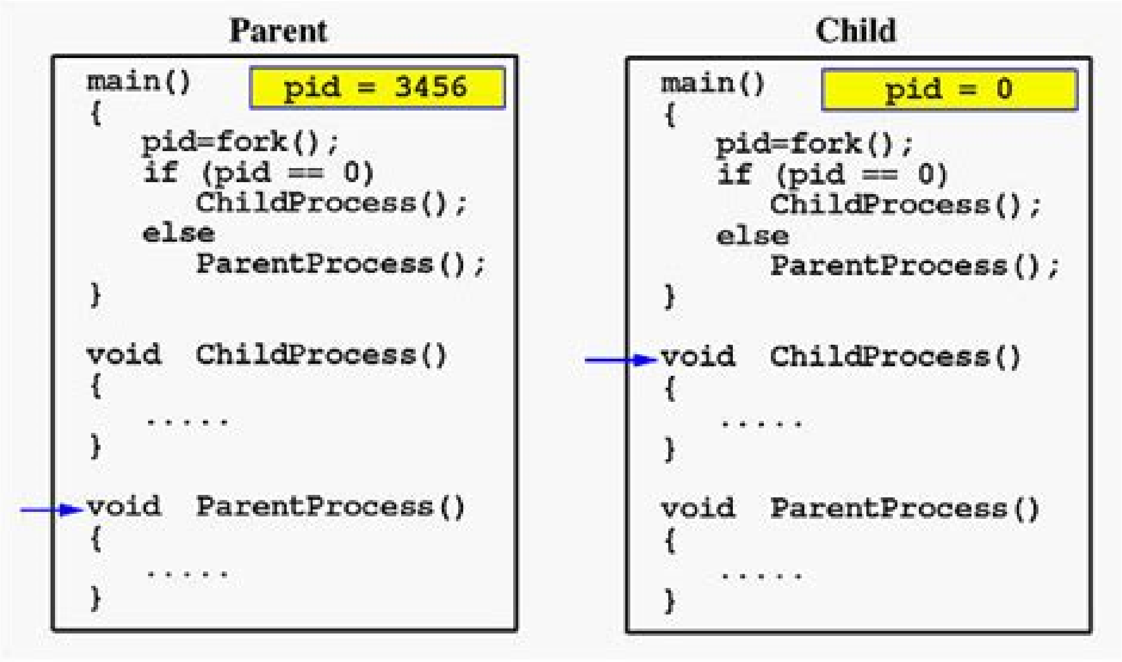
Illustration of Process Control Calls



https://www.geeksforgeeks.org/difference-fork-exec/

Processes - Programming

- *fork* system call
  - Creates a new process
  - The child process is a copy of the calling parent process
    - The parent process returns from the *fork* call with the process ID of its child process
    - The child process begins life as a return from the same *fork* call with a process ID of zero
  - The parent and child are now scheduled independently



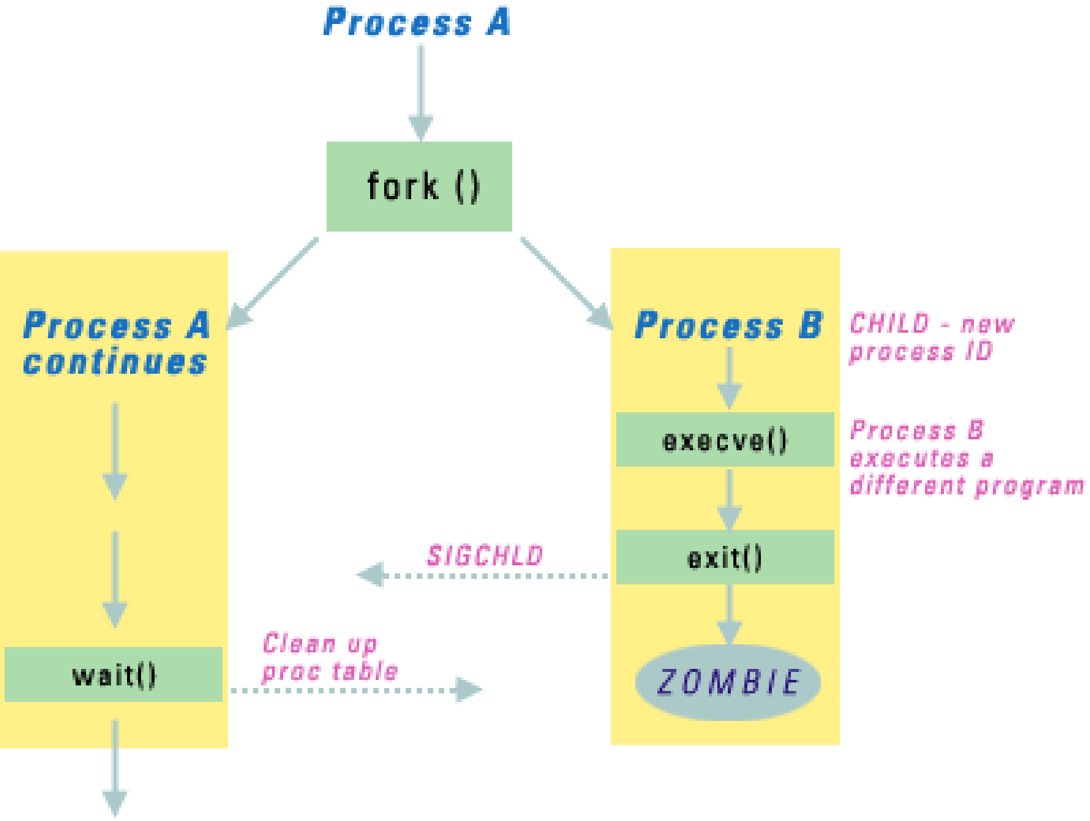
```
main()
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void  ChildProcess()
{
    .....
}

void  ParentProcess()
{
    .....
}
```

```
main()
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void  ChildProcess()
{
    .....
}

void  ParentProcess()
{
    .....
}
```

Processes - Programming

exec system call

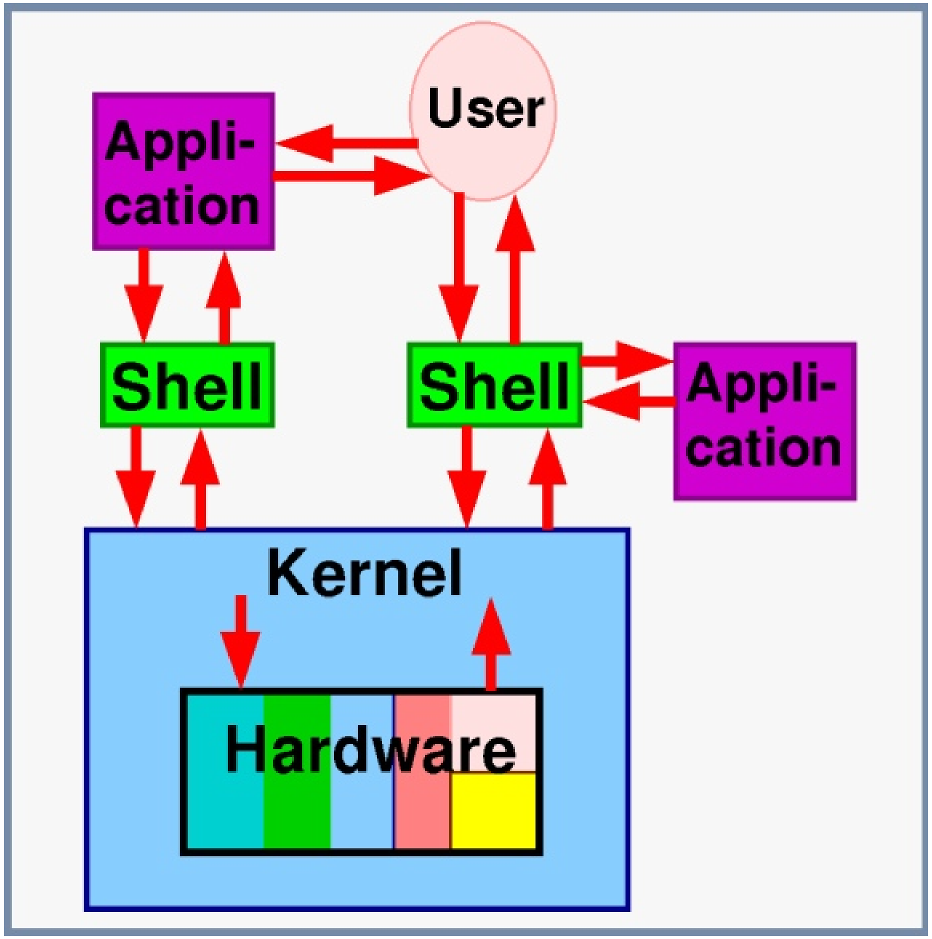
- Overlays the calling program with a new program and starts execution of the new program
- The calling program does not return from this call unless there is an error
- This all occurs within a single process



Shell

A program (a.k.a. command line interpreter) that allows the user to interact with the UNIX/Linux system.

- Reads user’s input.
- Parses it (evaluates special characters if any).
- Works with the kernel to execute the command.



Run some commands on CL?

Why use command line when we have GUI?

- Achieve complex tasks
- Get things done quickly and efficiently
- Perform tasks originally not thought of by authors
- Automate tasks easily

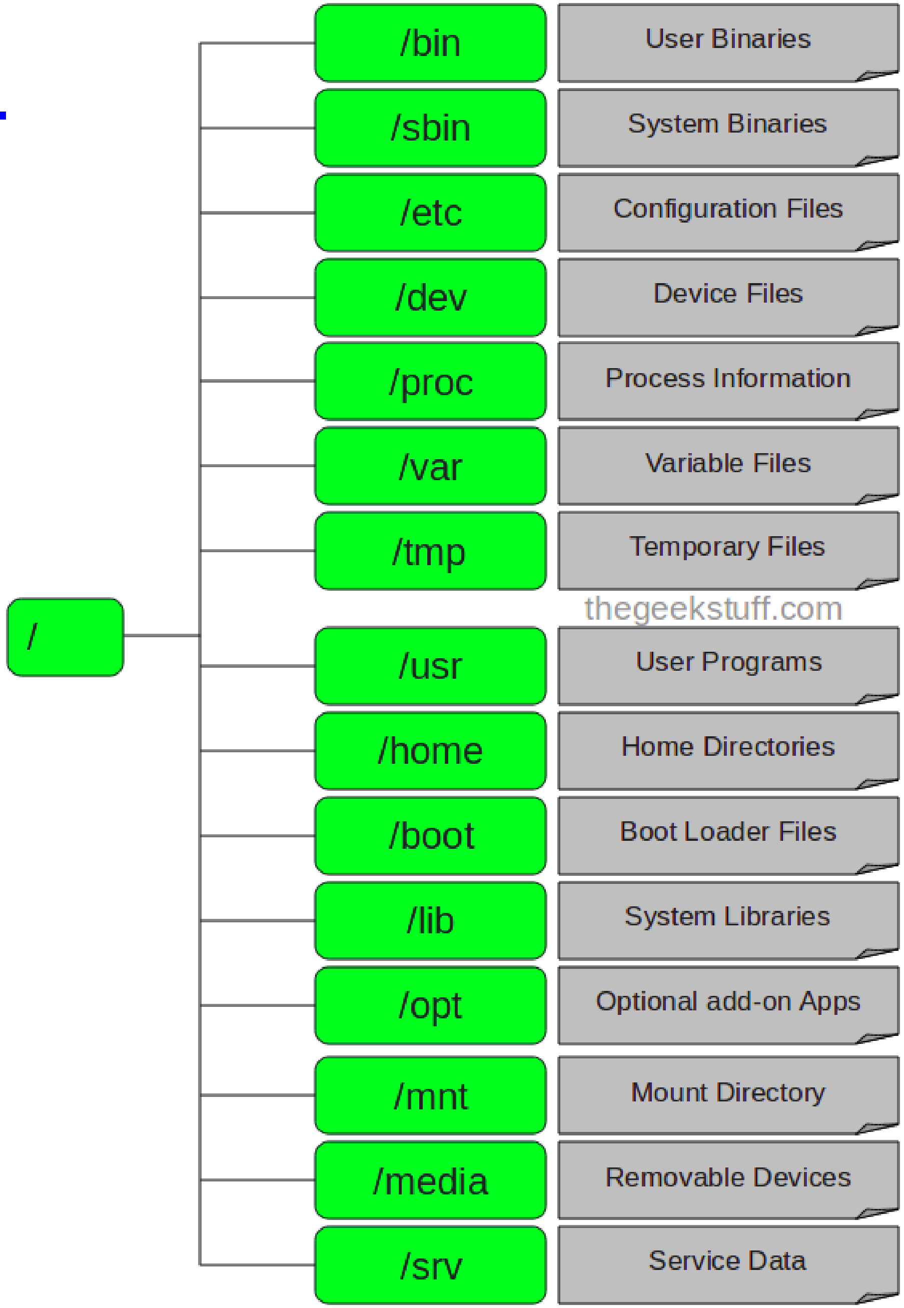
File System

- In Linux, everything is a file!
- Hierarchical organization
- Absolute vs Relative paths

➔ • ~ (tilde) - the home directory

➔ • . (a dot) - the current directory

➔ • .. (double dot) - the parent directory



/

/bin	User Binaries
/sbin	System Binaries
/etc	Configuration Files
/dev	Device Files
/proc	Process Information
/var	Variable Files
/tmp	Temporary Files
/usr	User Programs
/home	Home Directories
/boot	Boot Loader Files
/lib	System Libraries
/opt	Optional add-on Apps
/mnt	Mount Directory
/media	Removable Devices
/srv	Service Data

thegeekstuff.com

File System - File Types

- Regular
  - This is the normal file type: text, binary, programs, …
- Directory
  - Contains the information about files it houses e.g name and inode number
  - ‘.’ and ‘..’ are always present, current and parent directories
- Special/device
  - Connected to devices, printers, terminals, etc
  - Normally located in /dev
- Symbolic links
  - Provides an alias for an existing file
- Named pipes and sockets
  - Implement pipes and network connections

Basic Linux Commands

General Syntax

```
<SomeCommand> [option 1] [option 2] ...[option n]
```

The List command

```
ls [flags] [file]
```

- Lists directories

```
ls -ld */
```

```
ls -ld Videos
drwxr-xr-x  3 rekha rekha    4096 Jan 27  2018 Videos
ls -l foo1
-rwxrwxrwx 1 rekha rekha 2095 Jan  3  2018 foo1
ls -l /dev/zero
crw-rw-rw-  1 root root      1,     5 Jul 30 00:53 zero
```

Basic Linux Commands:help

The manual command

man <section> <command>

- Displays the manual page (manpage) of <command>.
- Use /<keyword> to do a keyword search in a manpage
- Make man your best friend!

Search/Locate commands

apropos <keyword>

Finds commands by keyword.

which <command>

whereis <command>

whatis <command>

Basic Linux Commands

General Syntax

<SomeCommand> [option 1] [option 2] ...[option n]

The List command

```
ls [flags] [file]
```

Lists directories

```
ls -ld */
```

```
ls -ld Videos
drwxr-xr-x  3 rekha rekha    4096 Jan 27  2018 Videos
ls -l foo1
-rwxrwxrwx 1 rekha rekha 2095 Jan  3  2018 foo1
ls -l /dev/zero
crw-rw-rw-  1 root root      1,     5 Jul 30 00:53 zero
```

Basic Linux Commands

Change Directory

cd [dir]

Changes directory to [dir].
Defaults to user’s home directory
if <dir> not given.

Current working directory

pwd

Know Your System

- echo $SHELL
- uname [-a]
- Whoami
- ps
- w(ho)
- ifconfig [-a]
- route
- df -h, du -h, free -m

File and Directory command: creation

Creating Files

touch [flags] <file>

- If the file exists, timestamp modified.
- If not, the file is created.

Creating Directories

mkdir [flags] <dir name>

- Creates a directory with the name <dir name>.