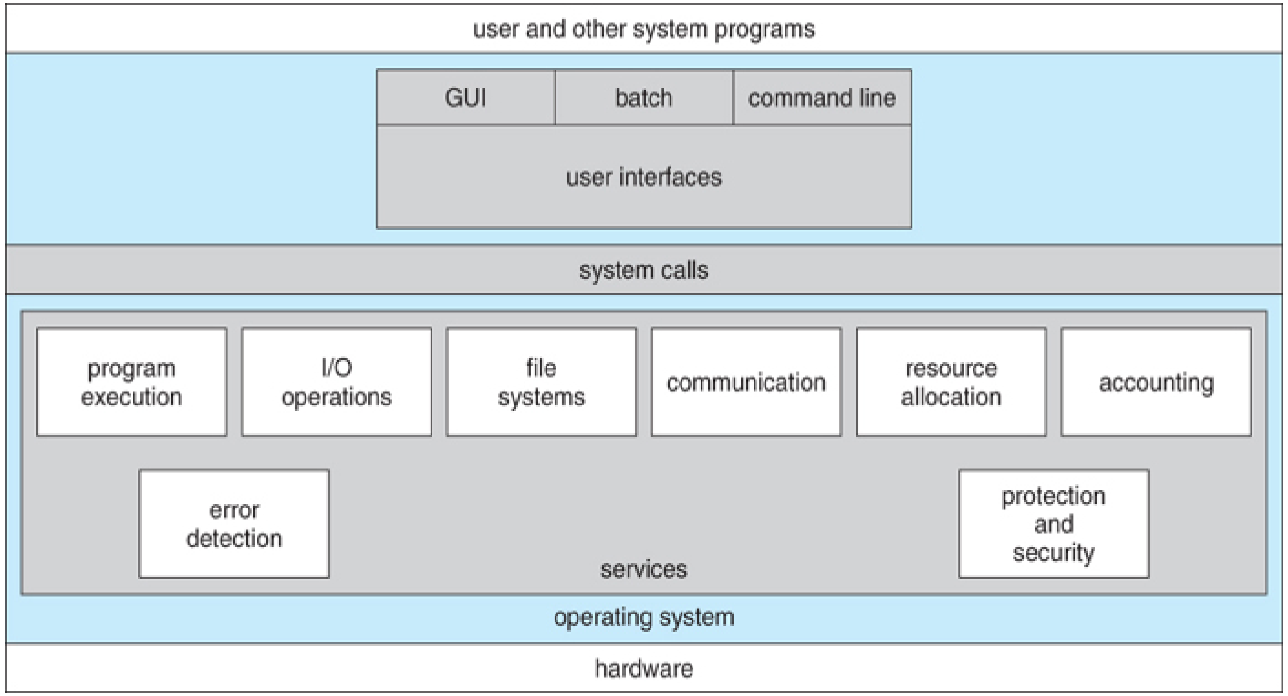
OS Structures

Spring 2022

Operating-System Services

- User interface
- Program Execution
- I/O operation
- File system manipulation
- Communication
- Error detection
- Resource allocation
- Accounting Protection and security

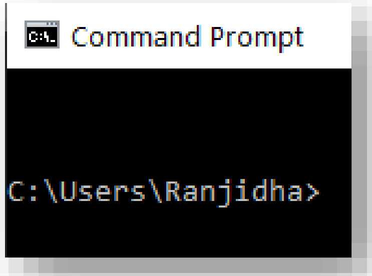


User interface

- Graphical User Interface (GUI)

- Touch screen

- Command-line interface



```
C:\Users\Ranjidha>
```

Program Execution

- Load the program into memory
- Run that program
- End its execution (normally or abnormally)

I/O operations

- Requirement for I/O from a program
  - File
  - I/O device
  - Reading from a network interface
  - Writing to a file system

File-system Manipulation

- Create file
- Read file
- Write file
- Delete file
- Search file
- Manipulate Directories
- Access Permissions

Communications

- Process to process communication
    - Same system
    - On a different system in the network
- Implemented
    - Shared Memory
    - Message passing

Error Detection

- Detecting errors
- Correcting errors

- Errors in
  - CPU & Memory
    - Memory error
    - Power failure
  - I/O devices
    - Parity error
    - Network failure
    - Lack of paper in printer
  - User Programs
    - Arithmetic overflow
    - Illegal memory access

Resource Allocation

- CPU cycles
- Main memory
- File storage
- I/O devices

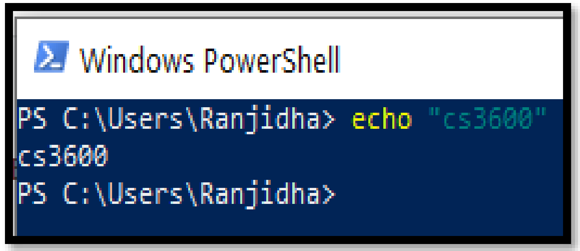
- CPU scheduling routines
  - Speed of CPU
  - Active process
  - Number of processing cores

Logging , Protection & Security

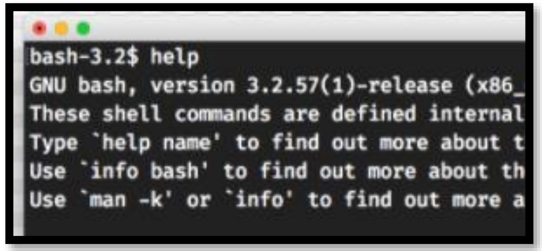
- Keep track of
  - Resources used by each program
  - Tools for system administrators
- Controlled access on system resources

User and Operating-System Interface

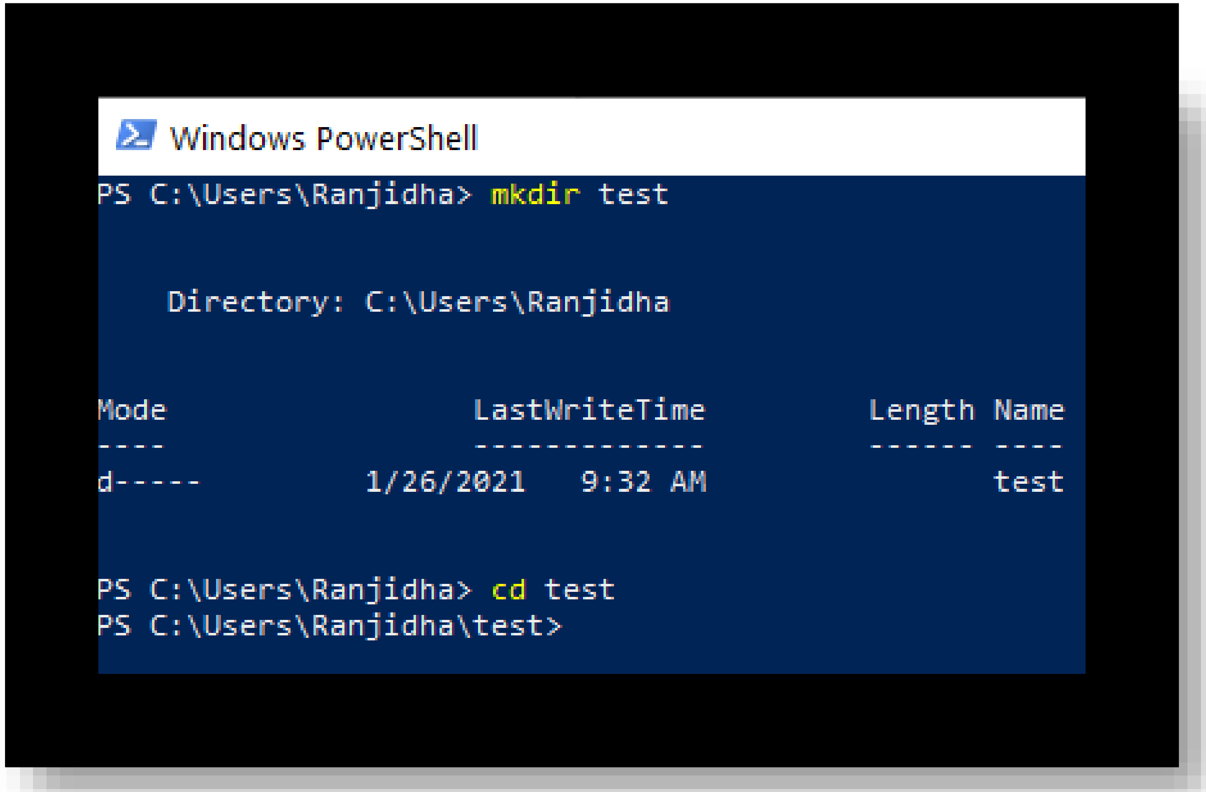
- Command-line interface



```
PS C:\Users\Ranjidha> echo "cs3600"
cs3600
PS C:\Users\Ranjidha>
```



```
bash-3.2$ help
GNU bash, version 3.2.57(1)-release (x86_
These shell commands are defined internal
Type `help name' to find out more about t
Use `info bash' to find out more about th
Use `man -k' or `info' to find out more a
```

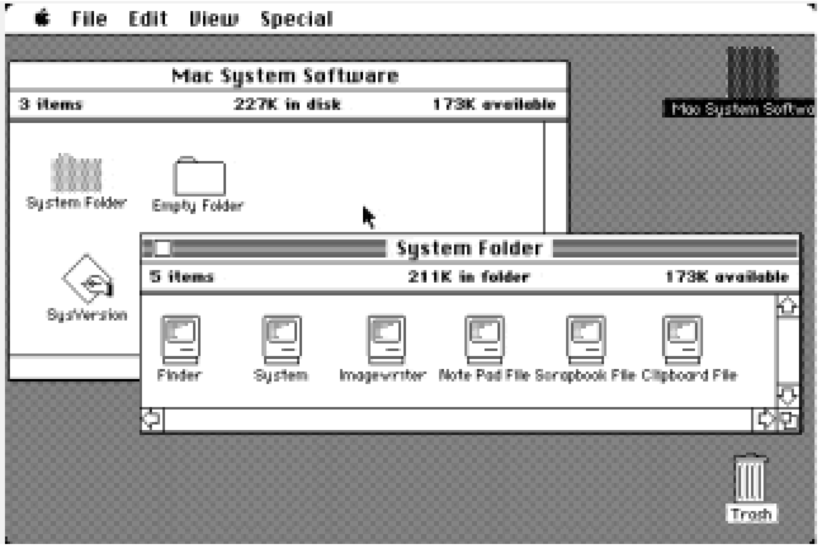


```
PS C:\Users\Ranjidha> mkdir test


    Directory: C:\Users\Ranjidha


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/26/2021   9:32 AM                test


PS C:\Users\Ranjidha> cd test
PS C:\Users\Ranjidha\test>
```

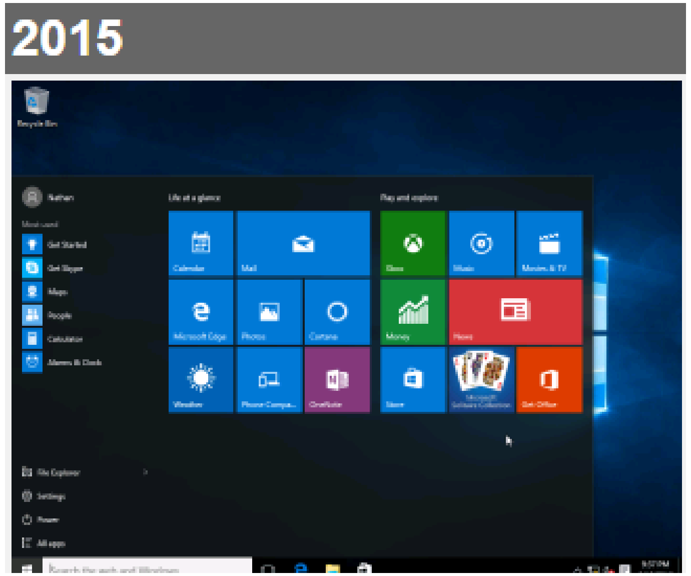
User and Operating-System Interface

- Graphical User Interface

April 1973, the first operational Alto computer is completed at Xerox PARC.



January 1984: Apple introduces the Macintosh.

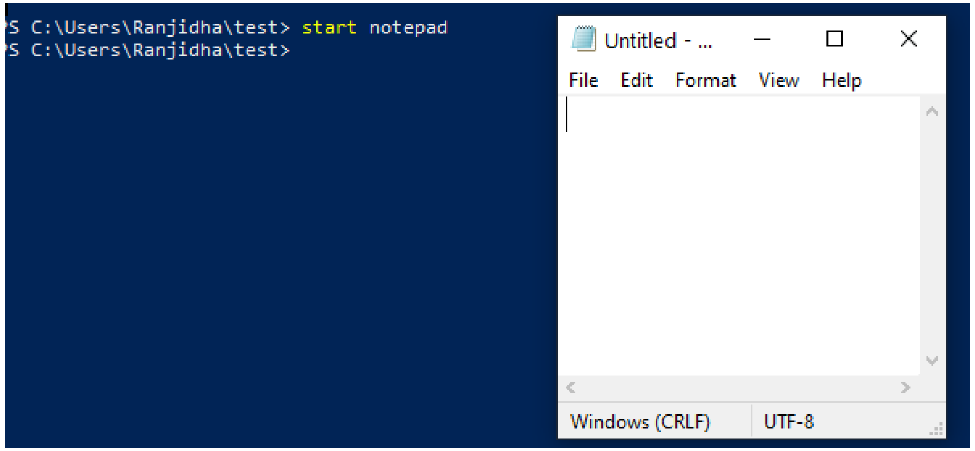


July 2015. Microsoft releases Windows 10

System Calls

- System calls provide an interface to the services made available by an operating system. These calls are generally available as functions written in C and C++.

What happens when we use an interactive desktop for the same process?



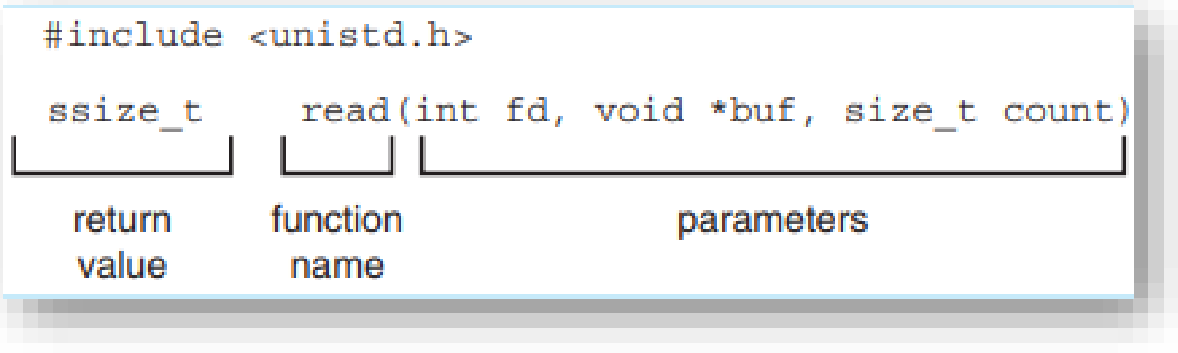
```
S C:\Users\Ranjidha\test> start notepad
S C:\Users\Ranjidha\test>
```

Application Programming Interface (API)

- The API specifies a set of functions that are available to an application programmer.
  - Windows API
  - POSIX API (UNIX, Linux & macOS)
  - Java API(Java Virtual Machine)

Why would an application programmer prefer programming according to an API rather than invoking actual system calls?

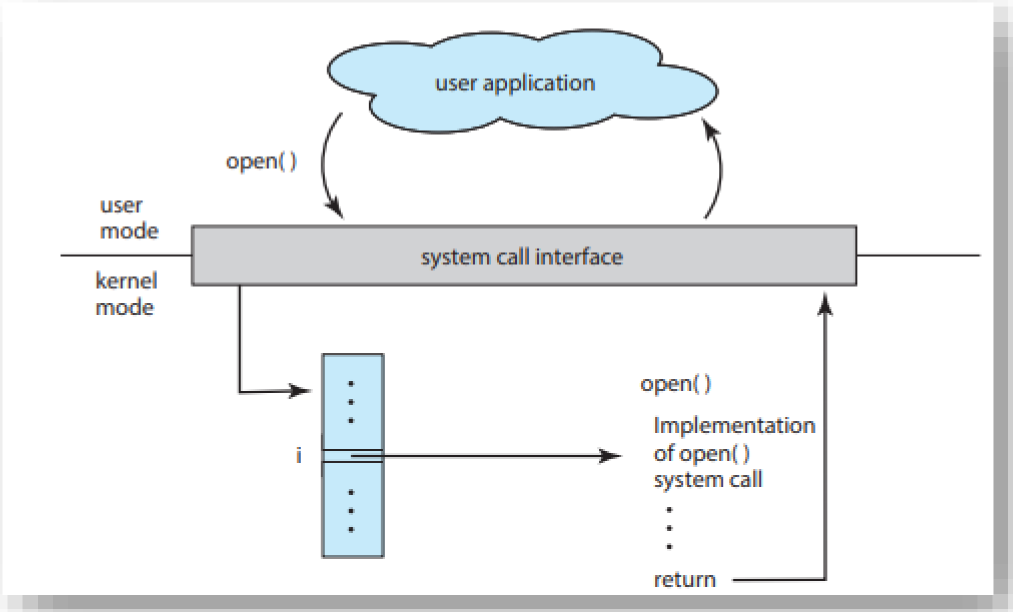
- A programmer accesses an API via a library of code provided by the operating system.



```
#include <unistd.h>

 ssize_t     read(int fd, void *buf, size_t count)
```

Application Programming Interface (API)

- Run-time Environment (RTE) : Provides a system call interface



Types of System Calls

- Process control
  - create process, terminate process
  - load, execute
  - get process attributes, set process attributes
  - wait event, signal event
  - allocate and free memory
- File management
  - create file, delete file
  - open, close
  - read, write, reposition
  - get file attributes, set file attributes
- Device management
  - request device, release device
  - read, write, reposition
  - get device attributes, set device attributes
  - logically attach or detach devices

Types of System Calls

- Information maintenance
  - get time or date, set time or date
  - get system data, set system data
  - get process, file, or device attributes
  - set process, file, or device attributes
- Communications
  - create, delete communication connection
  - send, receive messages
  - transfer status information
  - attach or detach remote devices
- Protection
  - get file permissions
  - set file permissions

System Calls Examples

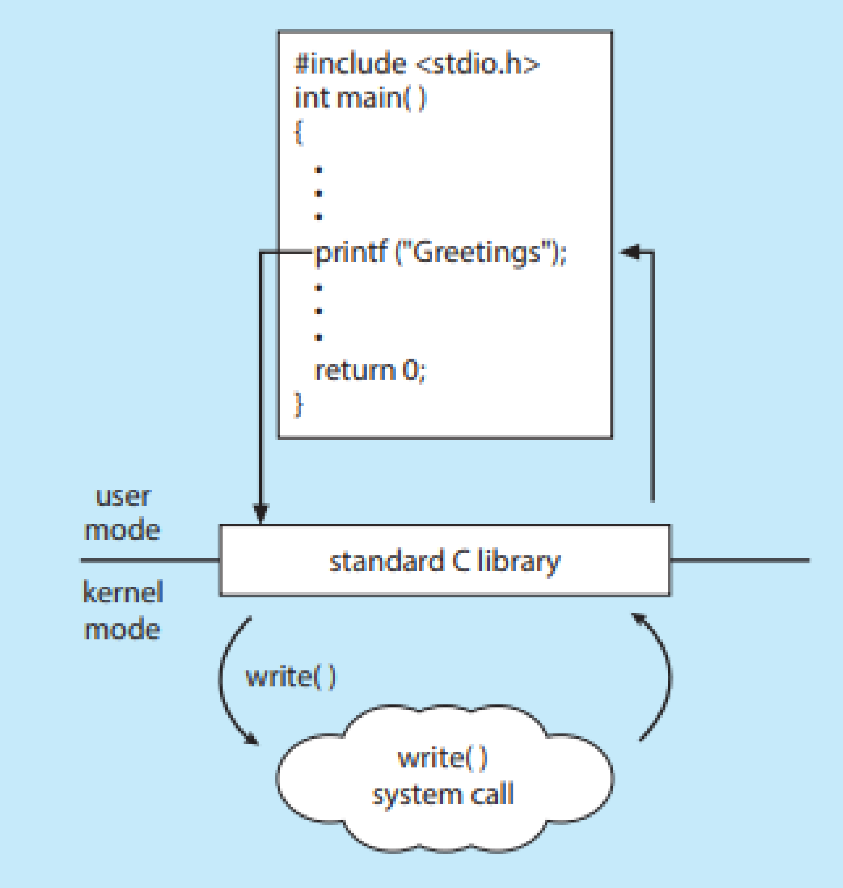
	Windows	Unix

Process Control	CreateProcess()
	ExitProcess()
	WaitForSingleObject()
		fork()
		exit()
		wait()

File management	CreateFile()
	ReadFile()
	CloseHandle()
		open()
		read()
		close()

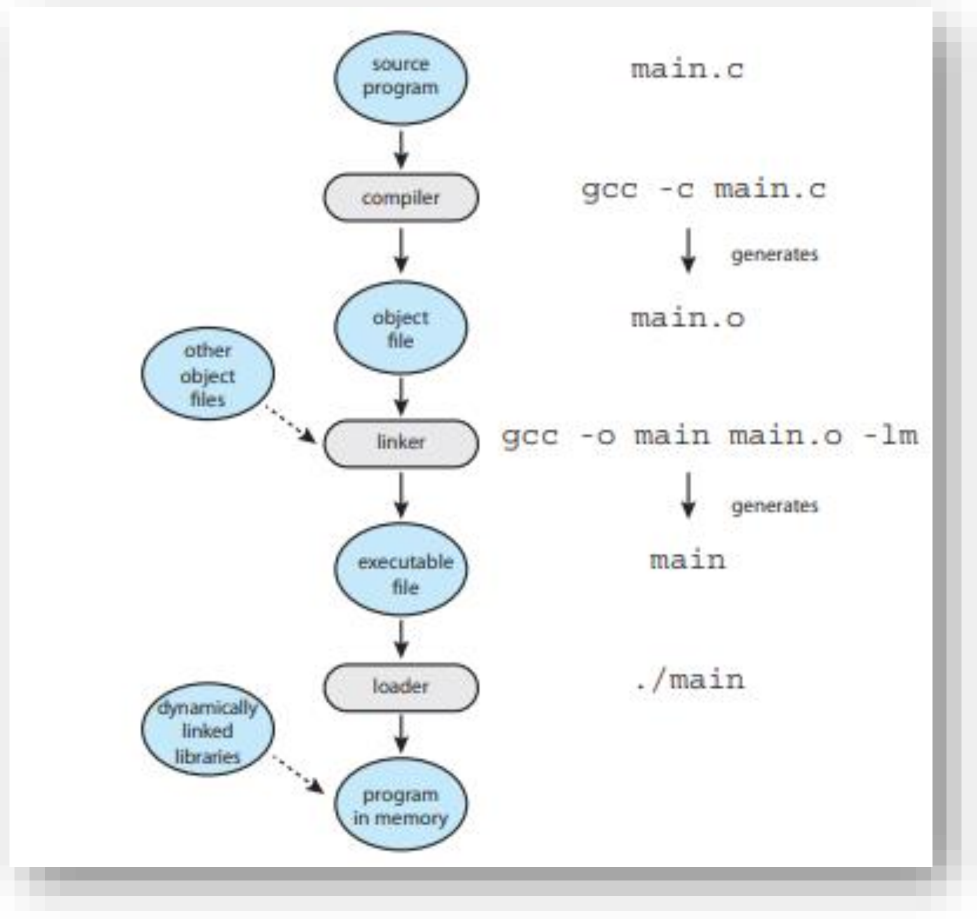
Communications	CreatePipe()	pipe()
Protection	SetFileSecurity()	chmod()

Example standard C library



```
#include <stdio.h>
int main( )
{
  .
  .
  .
  printf ("Greetings");
  .
  .
  .
  return 0;
}
```

Linkers and loaders



Object files & Executable files

- Compiled machine code
- Symbol Table (Metadata)

- Unix – Executable & Linkable Format (ELF)
- Windows – Portable Executable(PE)
- macOS – Mach-O format

Why Applications Are Operating-System Specific?

- An application compiled on one operating system are not executable on other operating systems ?
    - Each operating system provides a unique set of system calls.
- An application can be made available to run on multiple operating systems. How?
    - Python/Ruby?
    - Java?

Summary

- Unless an interpreter, RTE, or binary executable file is written for and compiled on a specific operating system on a specific CPU type (such as Intel x86 or ARMv8), the application will fail to run.

Supervisor call

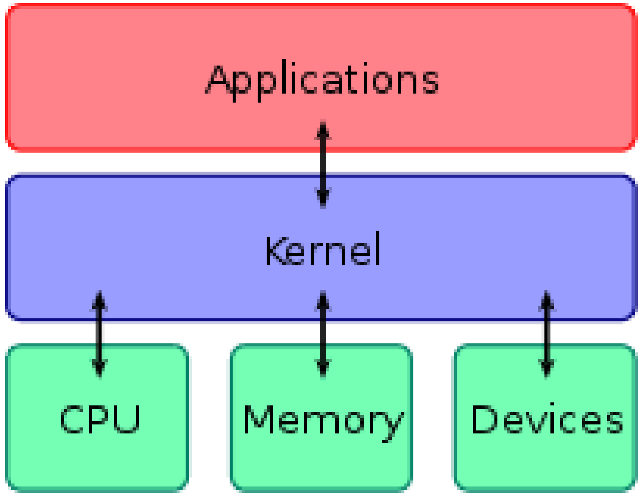
- A supervisor call (kernel call) is a privileged instruction that automatically transfers execution control to a well-defined location within the OS kernel.

Q5 in Worksheet

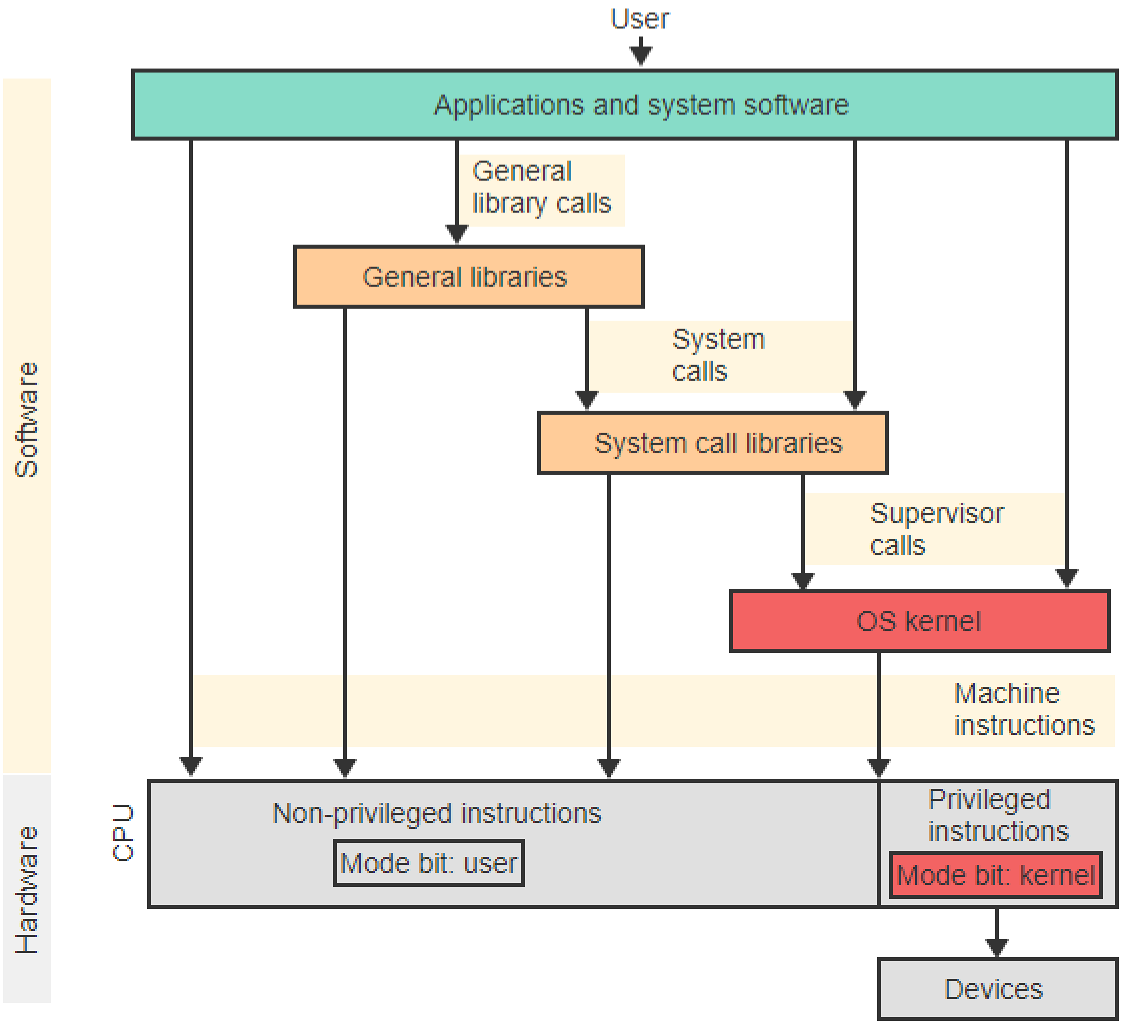
- Two concurrent applications, a1 and a2, execute the sequences of instructions (j1, j2, j3) and (k1, k2, k3), respectively. Execution switches between the applications whenever a timeout interrupt occurs or when one application terminates. If a2 starts, and interrupts occur after instructions k2 and j2, then what is the order in which the 6 instructions will execute?

Operating-System Structure

- Kernel is the core of operating system which is the interface to the hardware.
- OS kernel is allowed to execute privileged instructions.

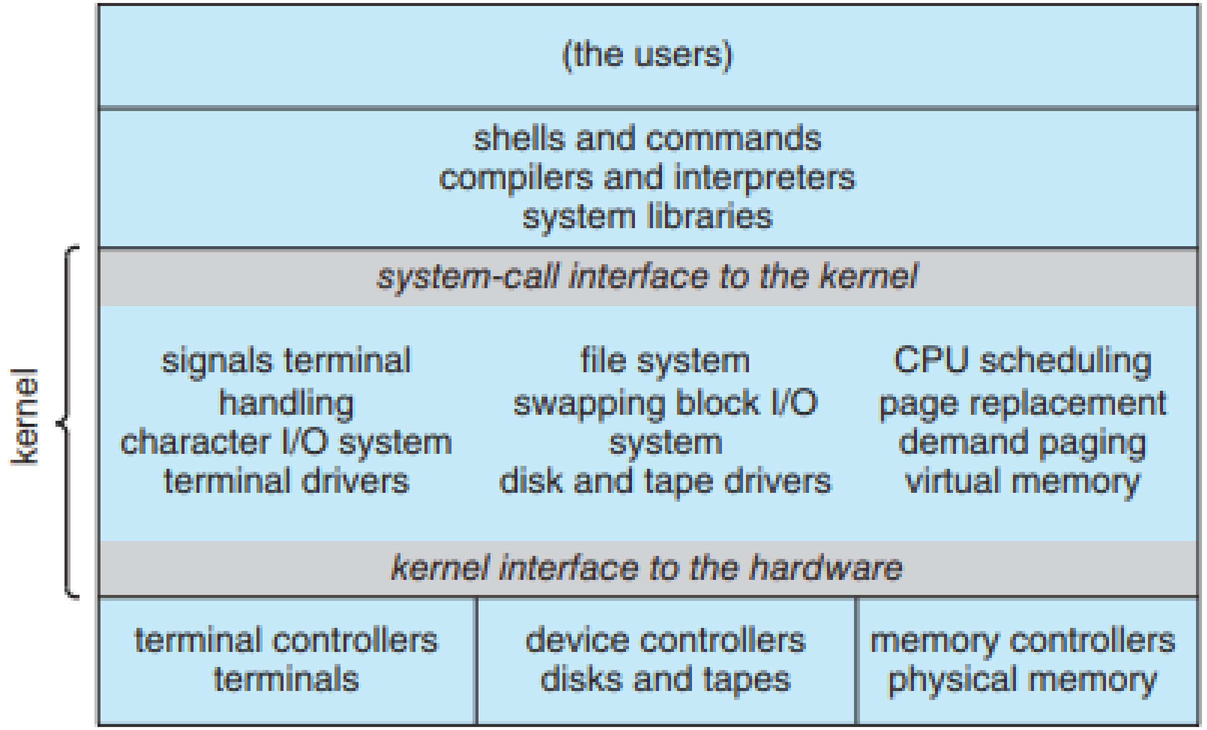


OS Hierarchy



Monolithic Structure

The functionality of the kernel into a single, static binary file that runs in a single address space.



(the users)

shells and commands
compilers and interpreters
system libraries

system-call interface to the kernel

signals terminal handling
character I/O system
terminal drivers

file system
swapping block I/O system
disk and tape drivers

CPU scheduling
page replacement
demand paging
virtual memory

kernel interface to the hardware

terminal controllers
terminals

device controllers
disks and tapes

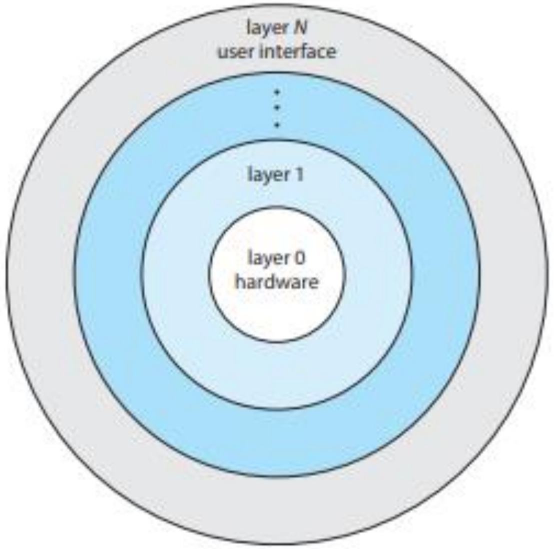
memory controllers
physical memory

kernel

Traditional UNIX system structure.

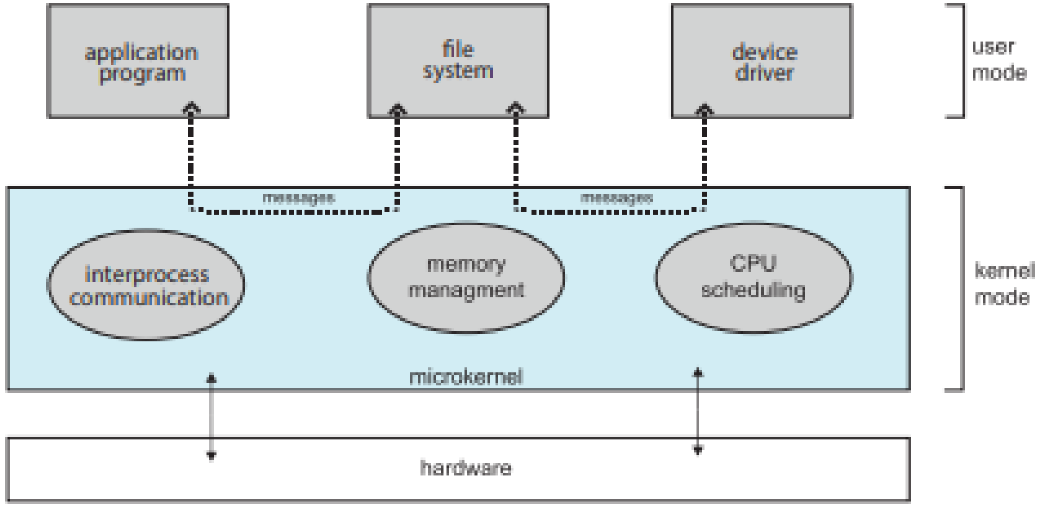
Layered approach

Modular Approach



Microkernel

- Nonessential components from the kernel and implementing them as user level programs that reside in separate address spaces.



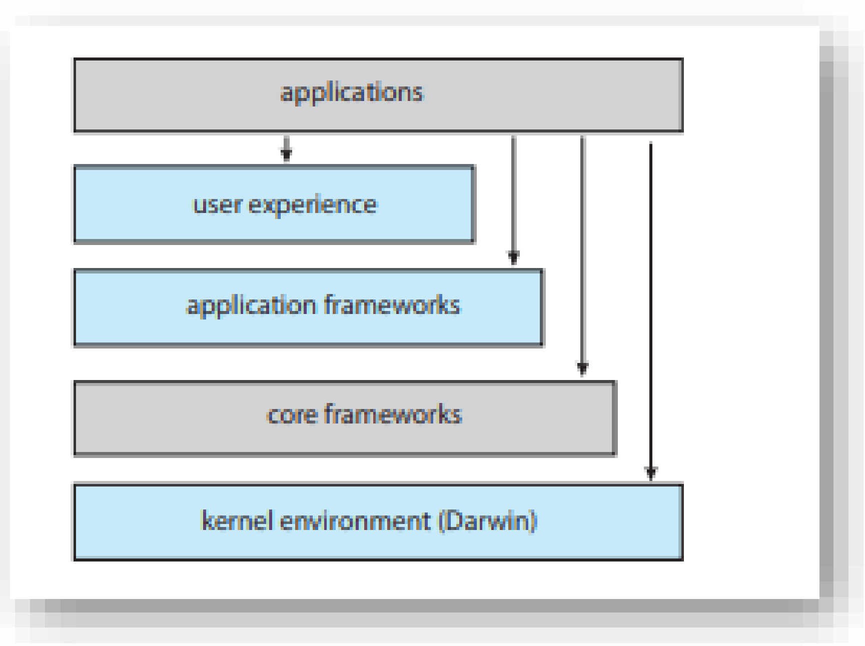
Modules

- Loadable Kernel module (LKM)
  - The kernel has a set of core components and can link in additional services via modules, either at boot time or during run time.

- Hybrid system
  - Combine different structures, resulting in hybrid systems

Mobile OS

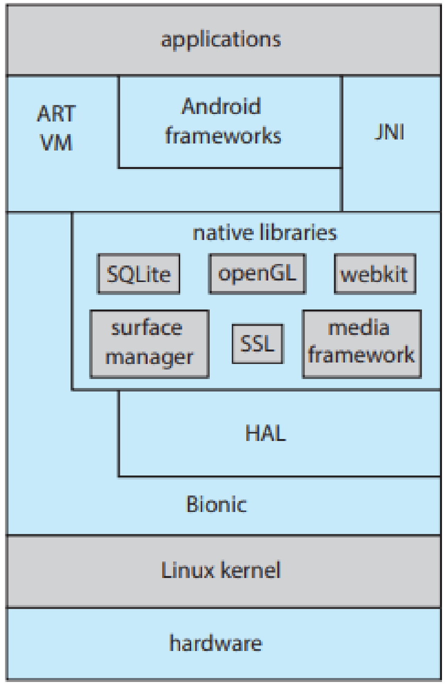
- User experience layer -This layer defines the software interface that allows users to interact with the computing devices.
    - iOS Springboard designed for touch devices
- Application frameworks layer - This layer includes the Cocoa and Cocoa Touch frameworks, which provide an API for the Objective-C and Swift programming languages.
- Core frameworks- This layer defines frameworks that support graphics and media including, Quicktime and OpenGL.
- Kernel environment - This environment, also known as Darwin



iOS operating systems

Mobile OS

- Architecture of Google’s Android.



Announcements (01/25/22)

- Read 1.1 and 1.2 in Module 1
- Next class
  - Please bring your laptop to class for software installations required for lab.