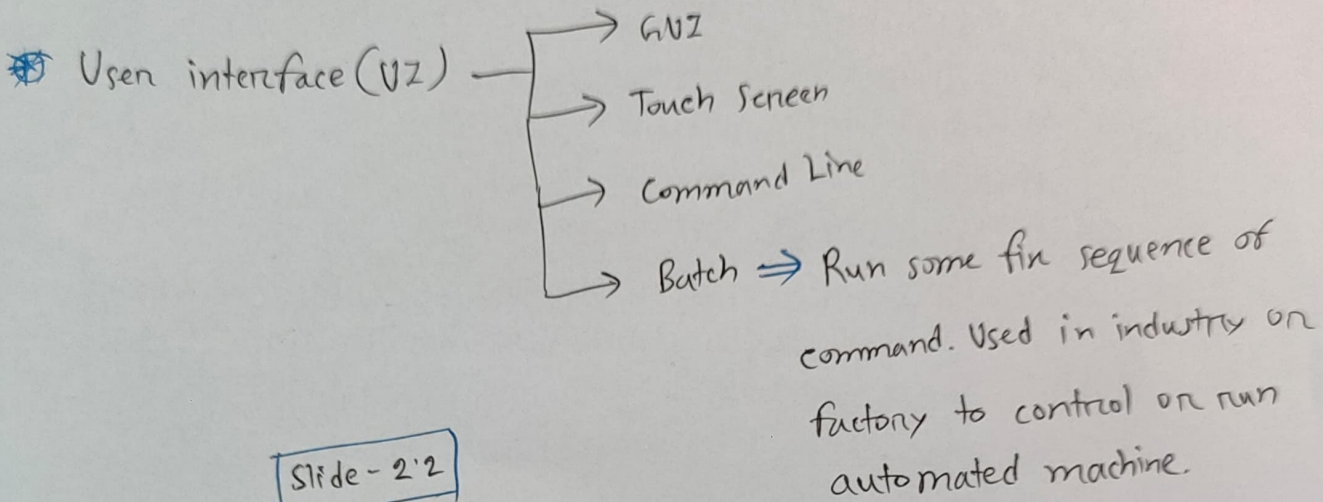


Chapter-2

Operating System Structure



* Features of interface of touchscreen devices.

- mouse not possible, not desire
- Action and selection based on gestures
- virtual keyboard for text entry
- voice commands

* Operating System Services:

(i) Program Execution: load and run program.
end normally or abnormally.

(ii) I/O Operation: control I/O devices for taking input or giving output.

iii) File System Manipulation:

- read, write file and directories
- create, delete, search, list all file info.
- permission management.

iv) Communication:

- control data movement computer to computer
- packets move by the OS

v) Error detection:

- always check for possible error.
- may occur in CPU, memory hardware, I/O devices

vi) Resource allocation:

- control resource usage by different process on Application program.
- CPU cycles, main memory, file storage, I/O devices etc.

vii) Accounting:

- control time limitation, count and warn
- keep track of resource usage time.

viii) Protection and security:

- ensure all access to system resources is controlled
- security for user authentication
- defends external I/O devices from invalid access attempts

* System Calls:

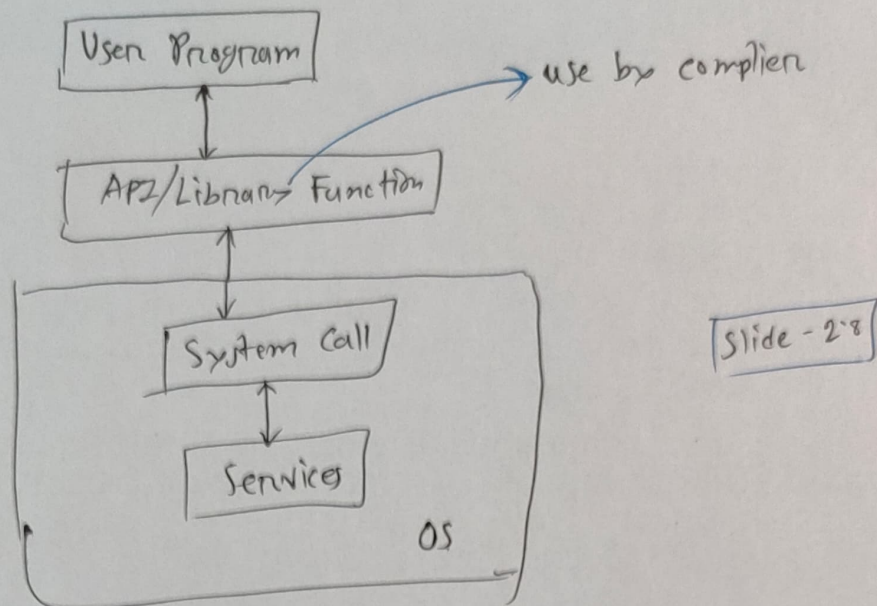
- Every service module has multiple functions. OS call these function for providing services. This is system call.
- Everytime we need to use system call to get a service.
- programming interface written in C or C++.
- programmers are not allowed to know the system call. They use API - Application Programming Interface.

* API

- Function name from system call need to call from user program.
- But for security, programmer don't know the function name used by system call.
- Mediator function, API is public for access.
- Win32, POSIX, UNIX, Java API.

Example:

~~API~~ ~~API~~
API calls \Rightarrow Create Process() \Rightarrow Create Process() {
Ntcreateprocess()
}
System Call \leftarrow



⊗ If two OS used same API, then user program can be run in both OS.

That's means same program can run in multiple OS if API is same.

Slide - 2.10

Loop
Until read fails } Not System Call

⊗ System-call interface

- ~~stored array~~
- a number associated with each system call.
- maintains a table indexed according to these numbers.

Slide - 2.12

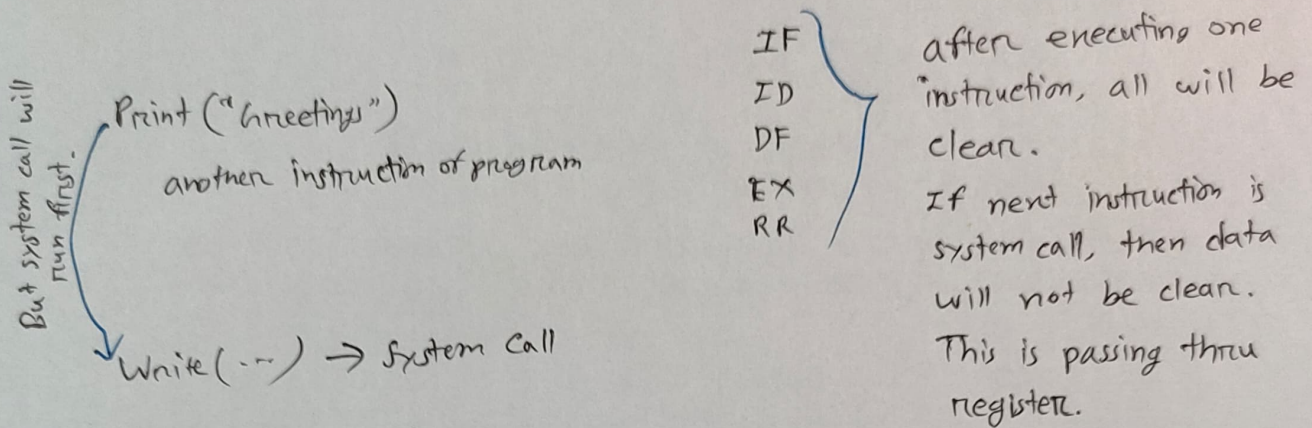
⊗ Standard C Library works as API.

L-2/09.03.2024/

Quiz-1

L-10/14.03.2024/

System Call Parameter Passing:



⇒ Passing thru block, table or array, starting address will store in the register. slide-2.15

- only if data size is larger than a register.

⇒ ~~Stack~~ Stack System

- whenever program loaded to memory a specified space by default reserved for the stack in the memory block of the program.

- But if data need to pop as per sequence as they are pushed then stack can't be used.

- If not supported, then use the block method (2nd option)

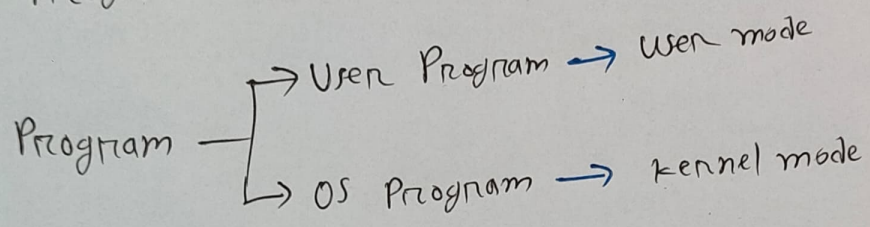
⊗ Types of System Call → Slide-2.16-2.19

- Process Control
- File management
- Device management
- Information maintenance
- Communication
- Protection

→ Just a brief idea, no need to memorize.

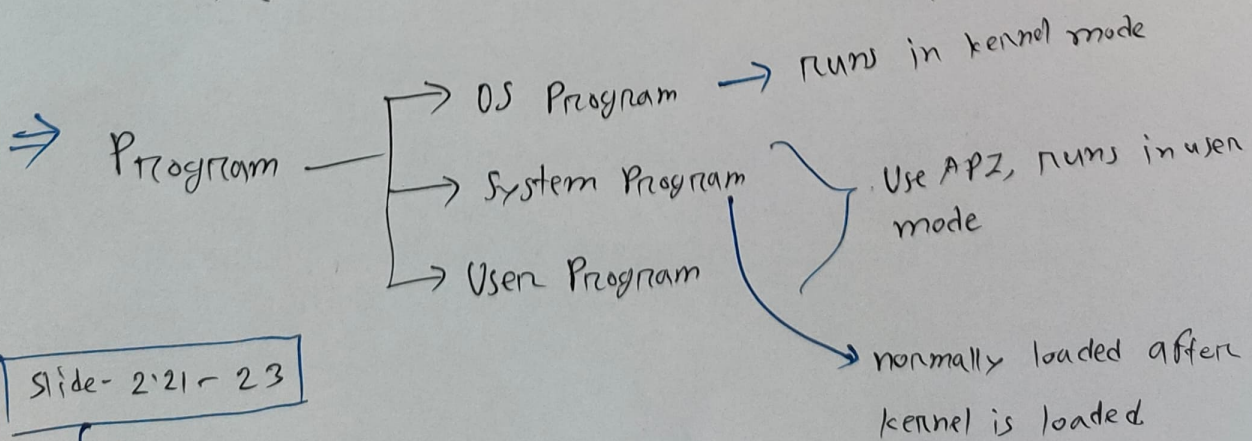
Example of system call, slide-2.20

⊗ System Programs:



⇒ Program comes with OS, Like calendar, Task Manager, File Explorer, ~~runs in~~ runs in user mode.

- These program used API for system call, that's why we can say, it runs in user mode.
- And these are system program.



Slide-2.21-2.23

→ Just a reading to take a idea of the software

* Operating System Structure:

- Simple Structure - MS-DOS
- More complex - UNIX
- Layered - an abstraction
- microkernel - Mach
- modular
- Hybrid

* Simple Structure - MS-DOS

Diagram - Slide-2.25

- program can access device drivers, as well as hardware.
- 8088, don't have dual mode.
- functionality not separated, not divided into module.

* Monolithic Structure - Original UNIX

- all of the functionality of the kernel into a single, static binary file that runs in a single address space.
- known as a monolithic structure.

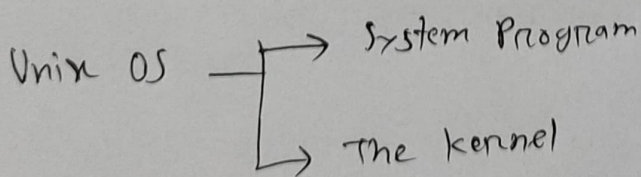


Diagram
slide-2.27

- ↳ all below system call interface and above physical hardware
- file system, CPU scheduling, mmu, etc.