CS & IT ENGINEERING

Process Management

OPERATING SYSTEM

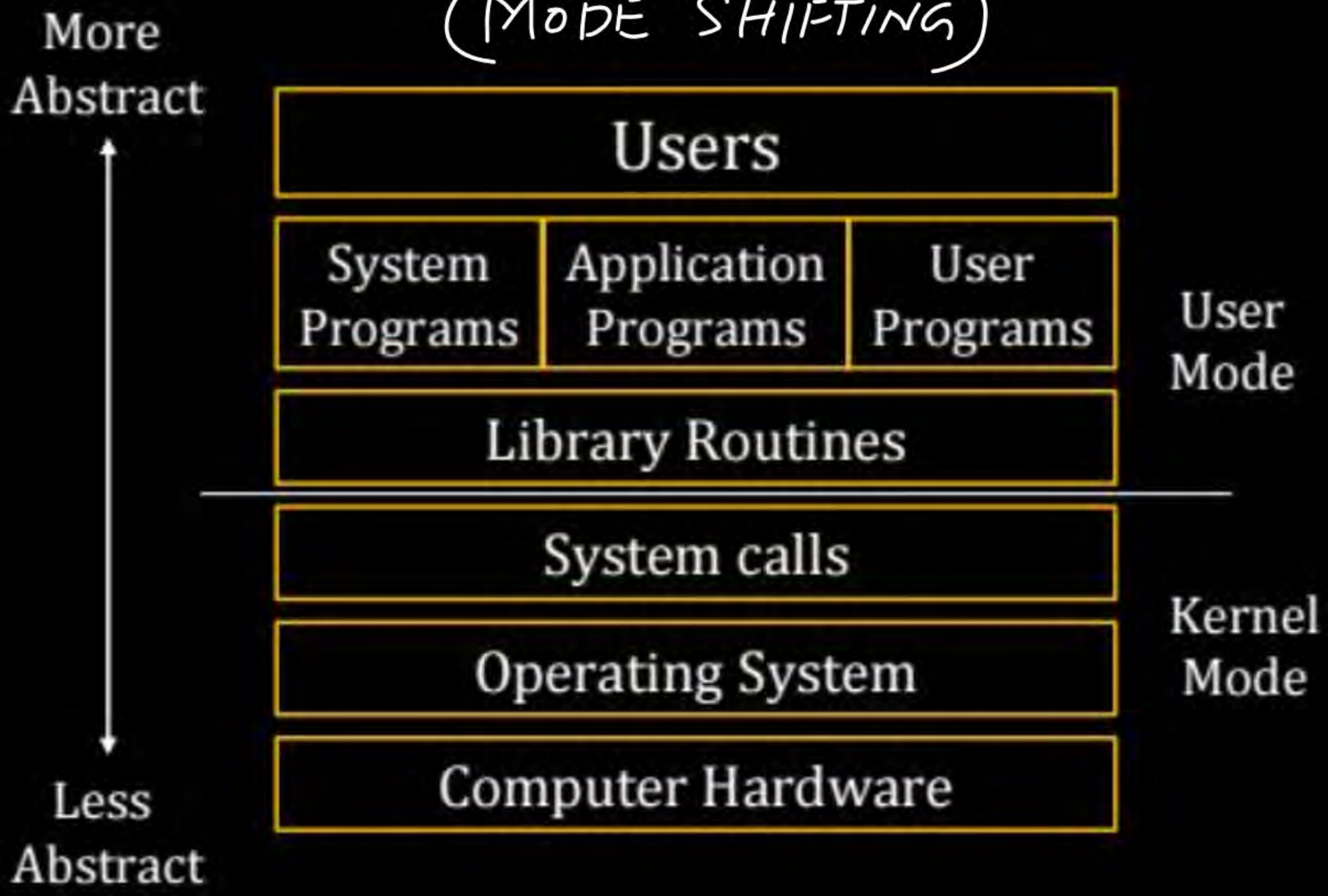
Lecture No. 01

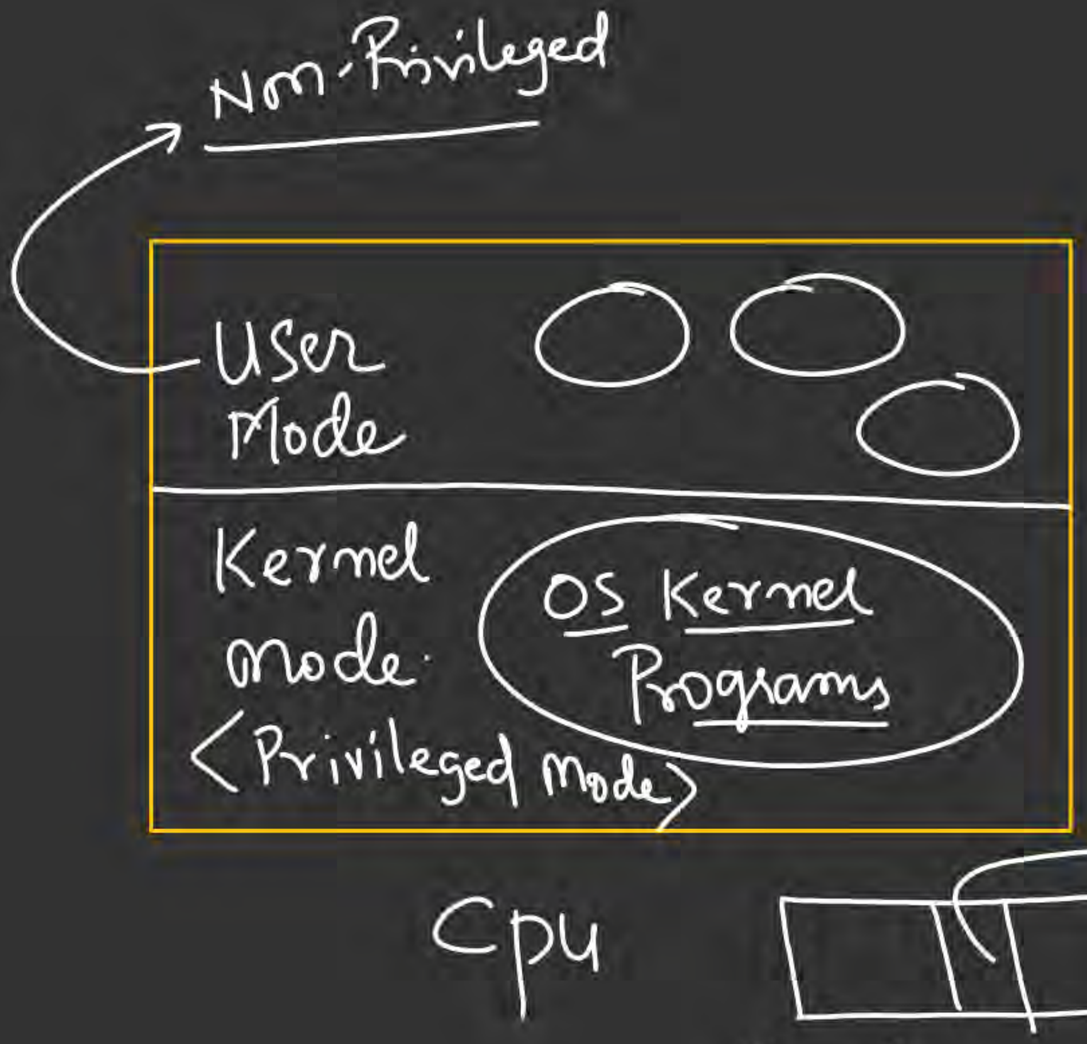
By- Dr. Khaleel Khan Sir

- Mode Shifting
- System Calls
- Problem Solving

User Mode vs Kernel Mode

(MODE SHIFTING)





→ U.M is PreEmptive (Non-Atomic)

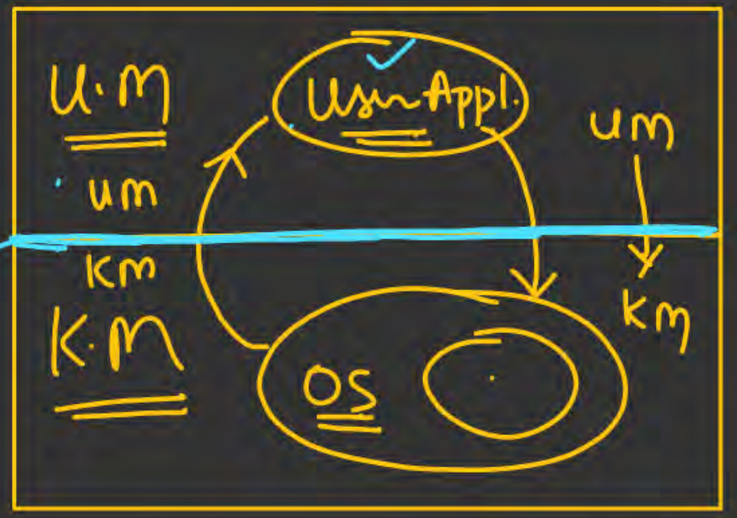
→ K.M is Atomic (Non-PreEmptive)

→ O.S is a Service Provider ;

→ User Appl's are Service users ;

→ Mode Shifting (Process) is needed to avail OS Services

Mode Shifting



→ Interface (API)

System call Interface

API : Appl. programmer's Interface

Impl. of Mode Shifting Process via API/S.C.I

```
main()
{
    int a,b,c;
    1. b=1;
    2. c=2;
    3. a=b+c;
    4. f(a);
}
```

```
f(int k)
{
    5. k++;
    6. printf("%d",k)
}
```

Predefined
- Library fn
- built-in

Impl. is in Library file (.lib)

```
f(1) ;
```

User-defined

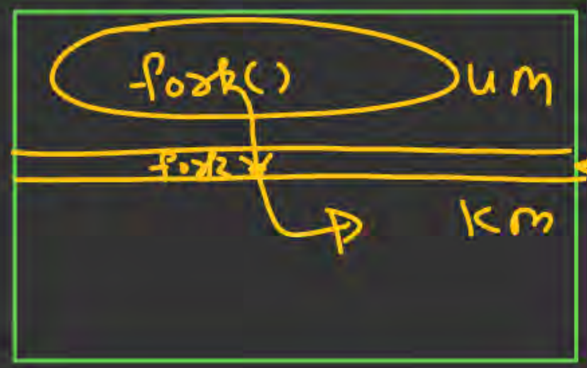
(.h) files Contain function Prototype → (Type-checking)

declaration

```
main()
{
    int a, b, c;
    a = 1;
    b = 2;
    c = a + b;
    f(c);
    fork();
    printf("%d", c);
}

f(int k)
{
    k++;
}
```

um



→ fork() is a fn

System_Call

(OS)

OS routine

→ fork() is Interface

defined

Impl.

in OS Kernel

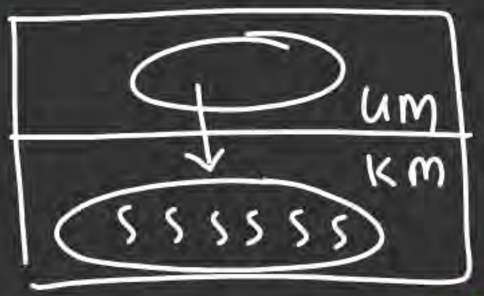
(Exec. of fork results in creating a child Process)

C·T = Compile Time ; BSA : Branch & Save Address

N·Privileged_Instn (um)

Run Time

R.I



SVC : Supervisory Call

(Privileged Instn)

```
main()
{
    .
    .
    f();
    .
    .
}
```

Instn → C·T → BSA -,-; → u·m

um: f() → C·T → fork() → SVC -,-;

(S/w Interrupt Instn)

(S/w Interrupt) → I·S·R

PSW: x 0 → mode bit

Int

```
fork int a,b,c;
     x
     .
     .
     Iv
     .
     .
c = a/b;
```

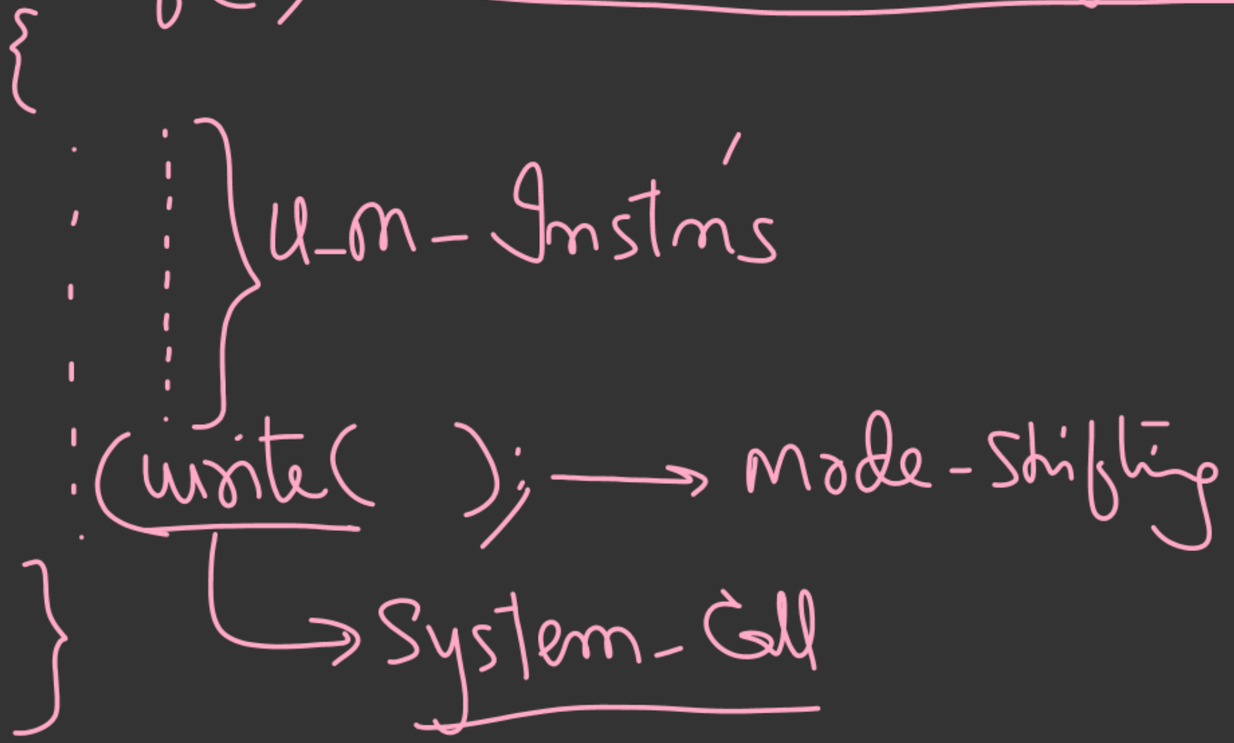
Pv

If b=0 int Divide overflow

(ISR: Interrupt Service routine)

Implementation: Library file

Printf( ): can also use Sys-Call

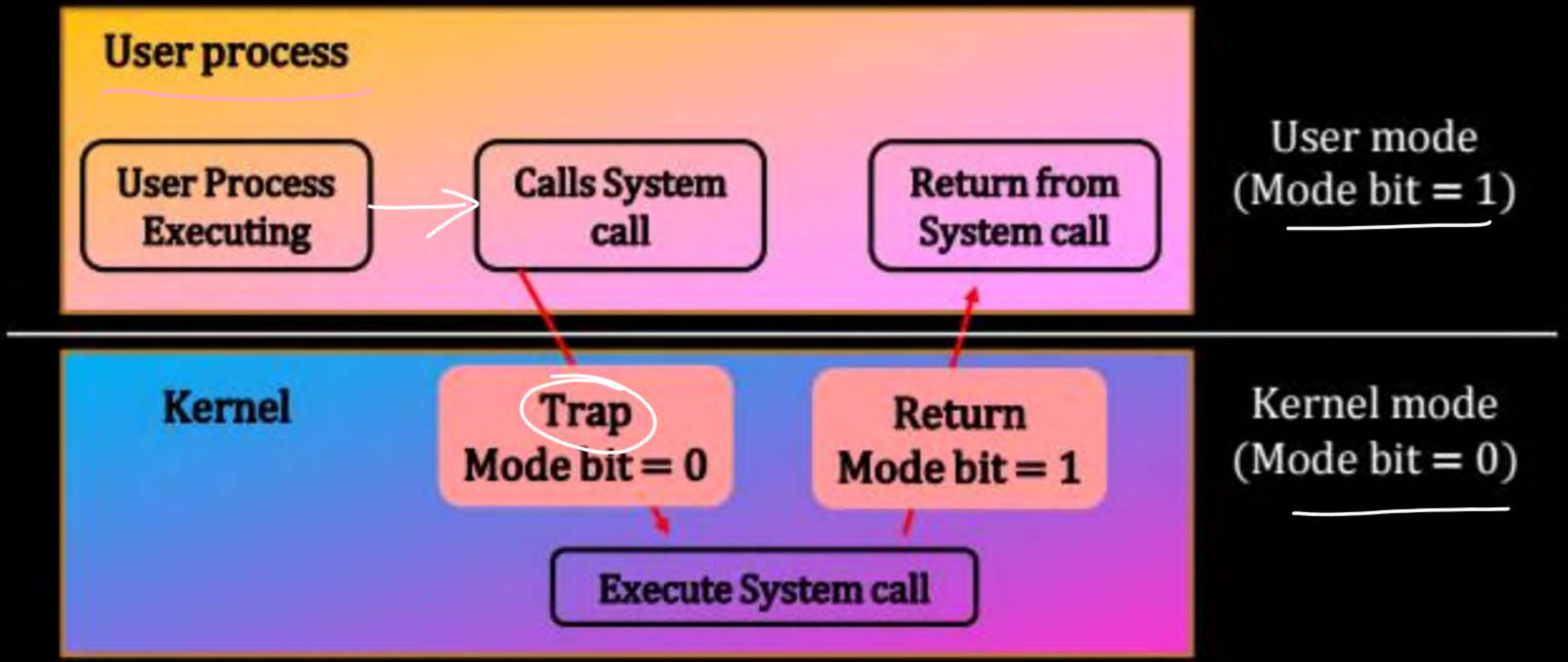


1. Kernel Mode

   In Kernel mode, the executing code has complete and unrestricted access to the underlying hardware. It can execute any CPU instruction and reference any memory address. Kernel mode is generally reserved for the lowest-level, most trusted functions of the operating system. Crashes in kernel mode are catastrophic; they will halt the entire PC.

2. User Mode

   In User mode, the executing code has no ability to directly access hardware or reference memory. Code running in user mode must delegate to system APIs to access hardware or memory. Due to the protection afforded by this sort of isolation, crashes in user mode are always recoverable. Most of the code running on your computer will execute in user mode.


User process
User Process Executing
Calls System call
Return from System call
User mode (Mode bit = 1)
Kernel
Trap Mode bit = 0
Return Mode bit = 1
Execute System call
Kernel mode (Mode bit = 0)

Q.1

A Processor needs Software Interrupt to

Sys_Call

A Test the Interrupt System of the Processor.

B Implement Co-Routines.

C Obtain system services which need execution of privileged instructions. ✓

D Return from subroutine.

Q.2 A CPU has two Modes-Privileged and Non-Privileged. In order to change the mode from Privileged to Non - Privileged.

(A) A Hardware Interrupt is needed.

(B) A Software Interrupt is needed.

(C) A Privileged Instruction (which does not generate an interrupt) is needed.

(D) A Non - Privileged Instruction (which does not generate an interrupt) is needed.

Q.3

System Calls are usually invoked by using:

A. A Software Interrupt ✓

B. Polling

C. An Indirect jump

D. A Privileged Instruction.

Pr. Instn

fork() → S.V.C → s/w Interrupt

Every Privileged Instn does not generate Interrupt

Q.4

A computer handles several interrupt sources of which the following are relevant for this question:

- Interrupt from CPU temperature sensor (raises interrupt if CPU temperature is too high)
- Interrupt from Mouse (raises interrupt if the mouse is moved or a button is pressed)
- Interrupt from Keyboard (raises interrupt if a key is pressed or released)
- Interrupt from Hard Disk (raises interrupt when a disk read is completed)

Which one of these will be handled at the HIGHEST priority?

A Interrupt from Hard Disk

B Interrupt from Mouse

C Interrupt from Keyboard

D ✓ Interrupt from CPU temperature

→ UNI – MULTI (M.Pr)

DOS

N.Pr

<Lack of Inter.>

WIN-3.0

Pr

Improving Interactiveness

→ Priority

→ Time

→ Timesharing

→ WIN|UNIX|LINUX

Architectural / H/w Requirements

(i) IO (Sec-Storage): DMA

(ii) Memory: Address Translation (mmu)

(iii) cpu: Dual Mode of cpu


THANK YOU
GW
SOLDIERS !