COMPUTER SCIENCE & IT

OPERATING SYSTEM

Functions & Goals

Lecture - 03

Dr. KHALEEL KHAN

Today's Goal

Fork System Call,
Problem Solving

Recap :

→ What is OS

→ Interface b/w User & H/W

Architecture changes

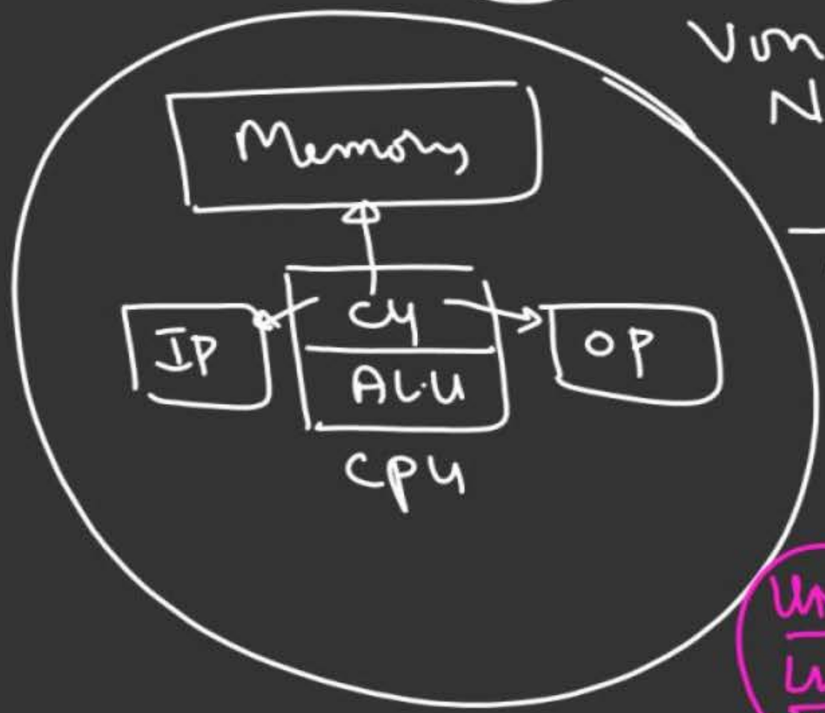
OS changes

IOS

Android/Win

Harvard Arch

Mobile Phone



Von-Neumann Arch.

Stored Prog. Concept

UNIX

LINUX

WIN

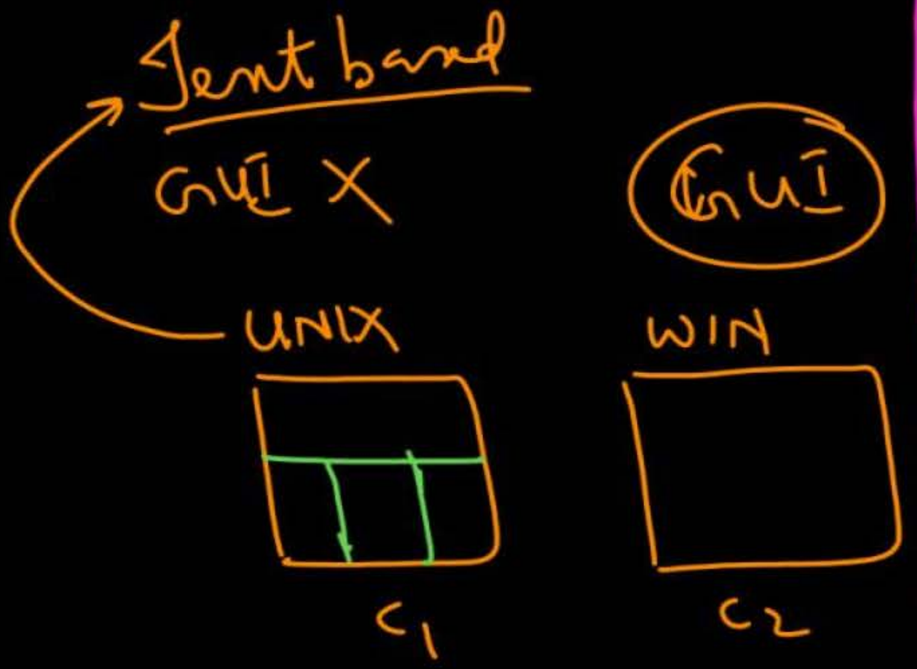
→ Control Program

→ Resource Manager

→ Set of utilities to simplify appl. develop.

→ Acts like a Govt

Functions & Goals of OS



A → Convenience
B → efficiency
Primary Goal of OS

C → Robustness
D → Reliability
E → Scalability (Ability to evolve)
F → Portability

Sat: 5-9
Sun: 9am-1pm

1., 7os for Personal Computing Von-Neumann Arch. Primary Goals is Convenience

Computing Domain changes, Goals changes

1. Real time Computing:

RTOS (RTS) → Systems that operate in strict deadlines (Time Constraints)

VxWorks

Ex: Missile Control, Satellite, ATC, Military operations, Nuclear Sys...

Real time Systems

Hard

→ Deadlines are very tight (strict)

→ heavy losses / Destruction

Soft

Enquiry System : Bank ATM

→ Deadlines Not strict

→ No loss / Destruction

Primary Goal of RTOS : efficiency

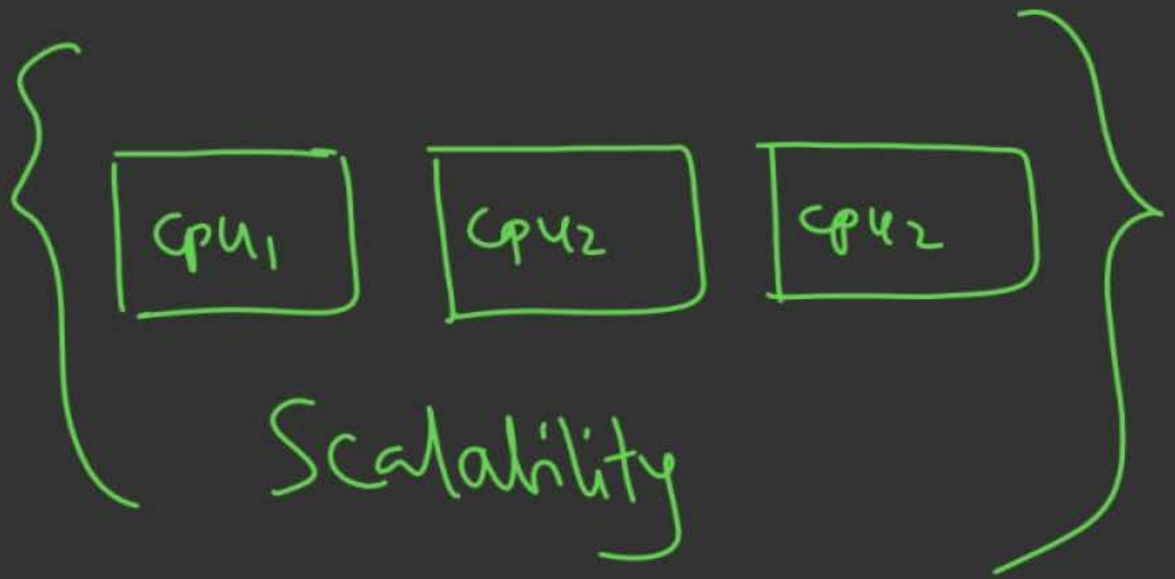
Mobile O.S (Mobile Computing)

→ Primary Goal ?

- Convenience
- efficiency
  - Battery Power

Parallel Computing ∞ main

∞ Distributed : Amoeba (Multi-processor Systems)



Types of O.S?

Disk Technology

- Uniprogrammed (U.Pr)
- Multiprogrammed (M.Pr)

1st Gen: 1930 - 40's
No OS
(Card-Reader)

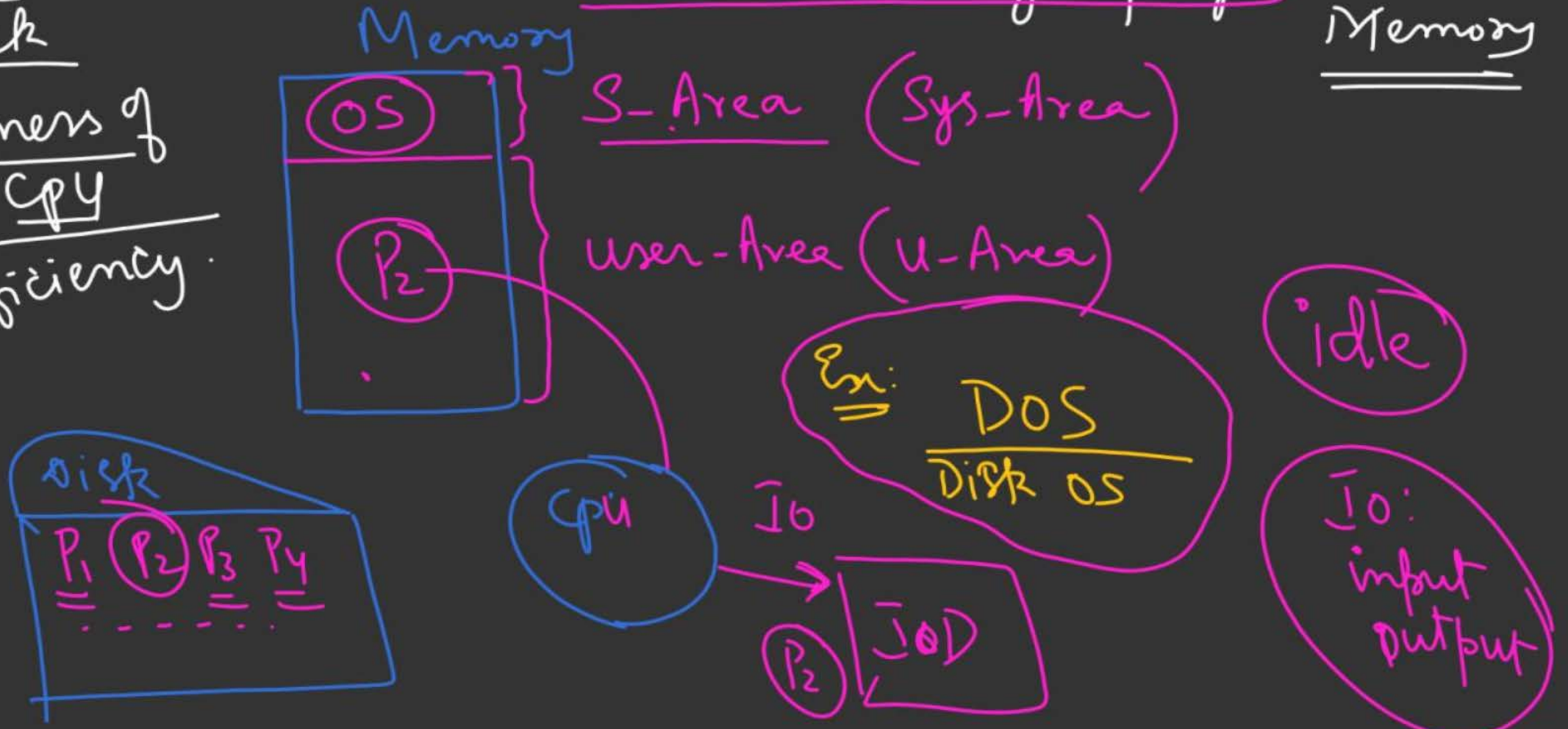
2nd Gen: (1940 - 50)
Mag. Tapes
No OS

3rd Gen.: 1950 - 60's
(Mag. Disk)

Uniprogramming: Ability of the O.S to Load and execute a Single program in Memory

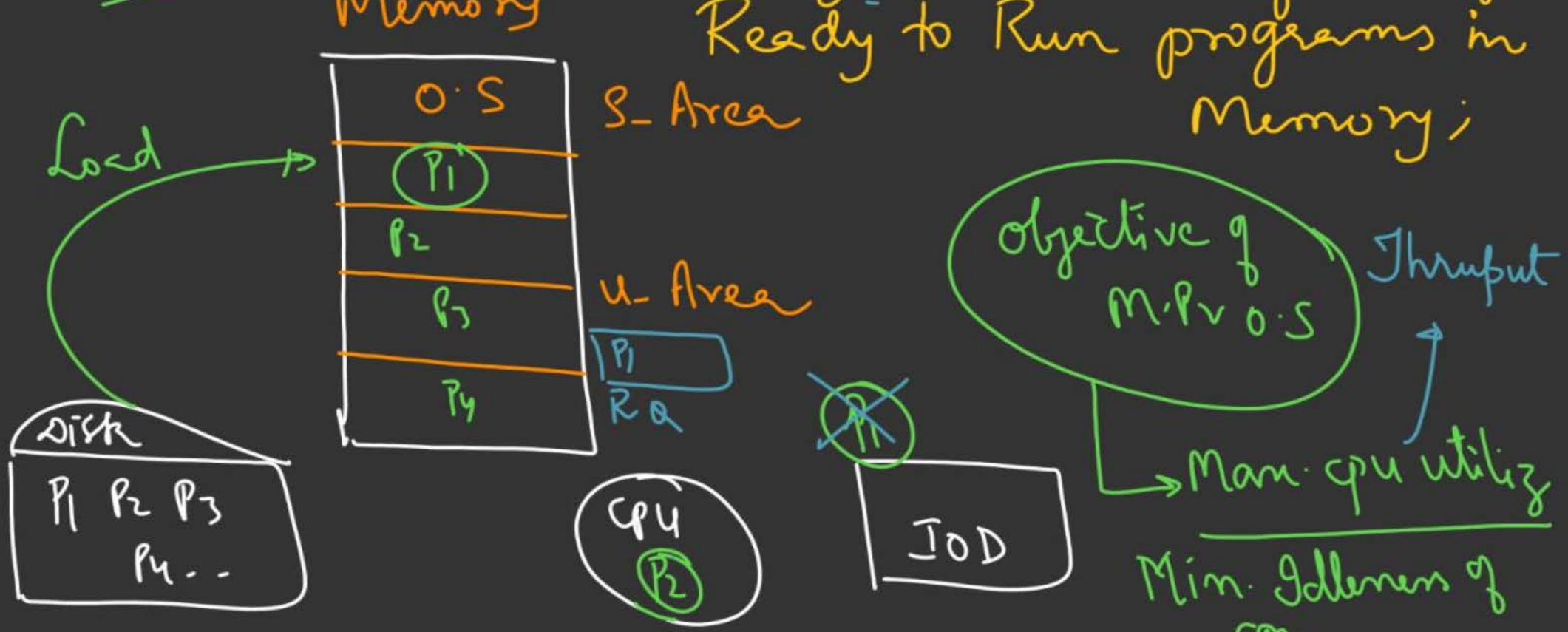
Drawback: Idleness of CPU

Inefficiency.



Multiprogramming: Ability of O.S to Load and manage execution of Multiple Ready to Run programs in Memory;

M.Pr



Throughput: No. of Programs/Tasks/appl. Completed per unit time;

10 programs — 60 m
? — 1 m

$\frac{10}{60} = \frac{1}{6}$

9-1

4 x 60
= 240 m

15 topics — 240 m
? — 1 m

$\frac{15}{240}$

Types of Multiprogramming :

Pr. M.Pr OS : UNIX ; LINUX ; WINDOWS ; MACH

Non Preemptive (N.Pr)

- WIN 3.x
- No forceful deallocation
- Program releases CPU voluntarily
- Starvation (responsiveness / Interactiveness)
- Completion
- I/O
- System Call

IOS

Preemptive (Pr)

- Forceful deallocation
- Function call to O.S for availing OS service

Enforce Preemption:
- Priority
- Time

Pr. M. Pr is Multitasking

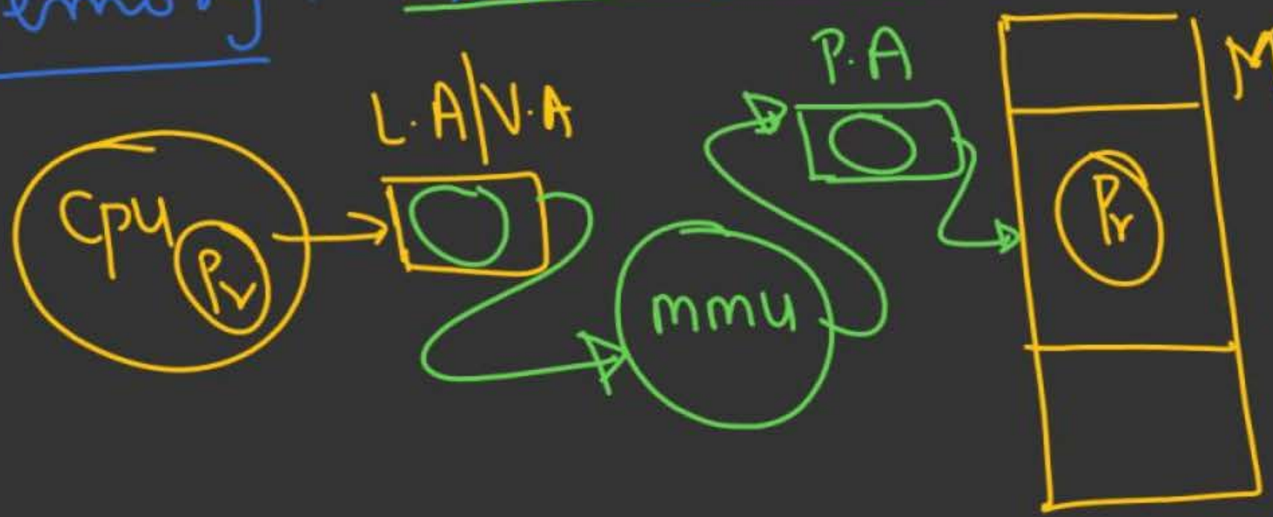
* Architectural Requirement for Implementation of a PreEmptive based M.Pr.O.S

H/W Support

1) IO (Secondary Storage devices) : {DMA} (Direct Memory access)

MMU: Mem Mgmnt unit

2) Memory : Address Translation Support



LA: Logical Addrs

VA: Virtual "

Imp

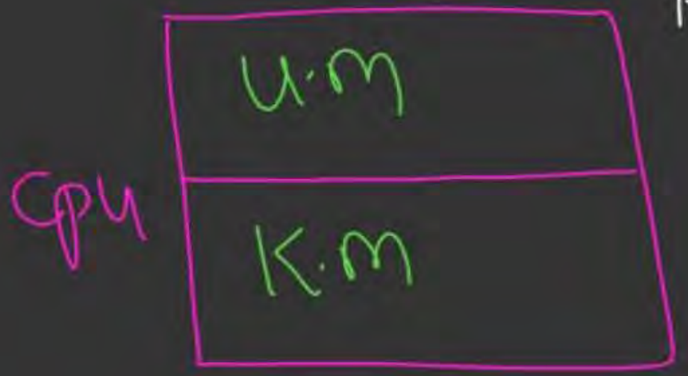
3) CPU : Dual Mode operation

cpu: central processing unit

(**)

: Every Processor in a M.Pr O.S must Support Imp Modes (atleast) for M.P. OS

- (UM) User Mode — Non Privileged
- Kernel Mode (KM) — Privileged — System



User Mode

1. User Programs/appl's run in User Mode;
2. User Mode execution is PreEmptive (Non-atomic)
3.

Kernel Mode

1) O.S programs/routines run in Kernel Mode;
2) Kernel Mode execution of OS programs is Non-PreEmptive (atomic)

1. Road Transport

- Ambulance (km) (OS)
- Common Man Vehicle (um)

(user appl.)

2. Convoy of P.m/c.m

⑧ Mode Shifting (MSB)

→ O.S is a Service Provider

→ Users & Programmers are Service users

→ Many a times during the execution of user Programs, it is reqd to access/avail OS Service

→ Mode Shifting from UM → KM, whenever an User appl needs an OS Service;

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

User-level Instns / UM

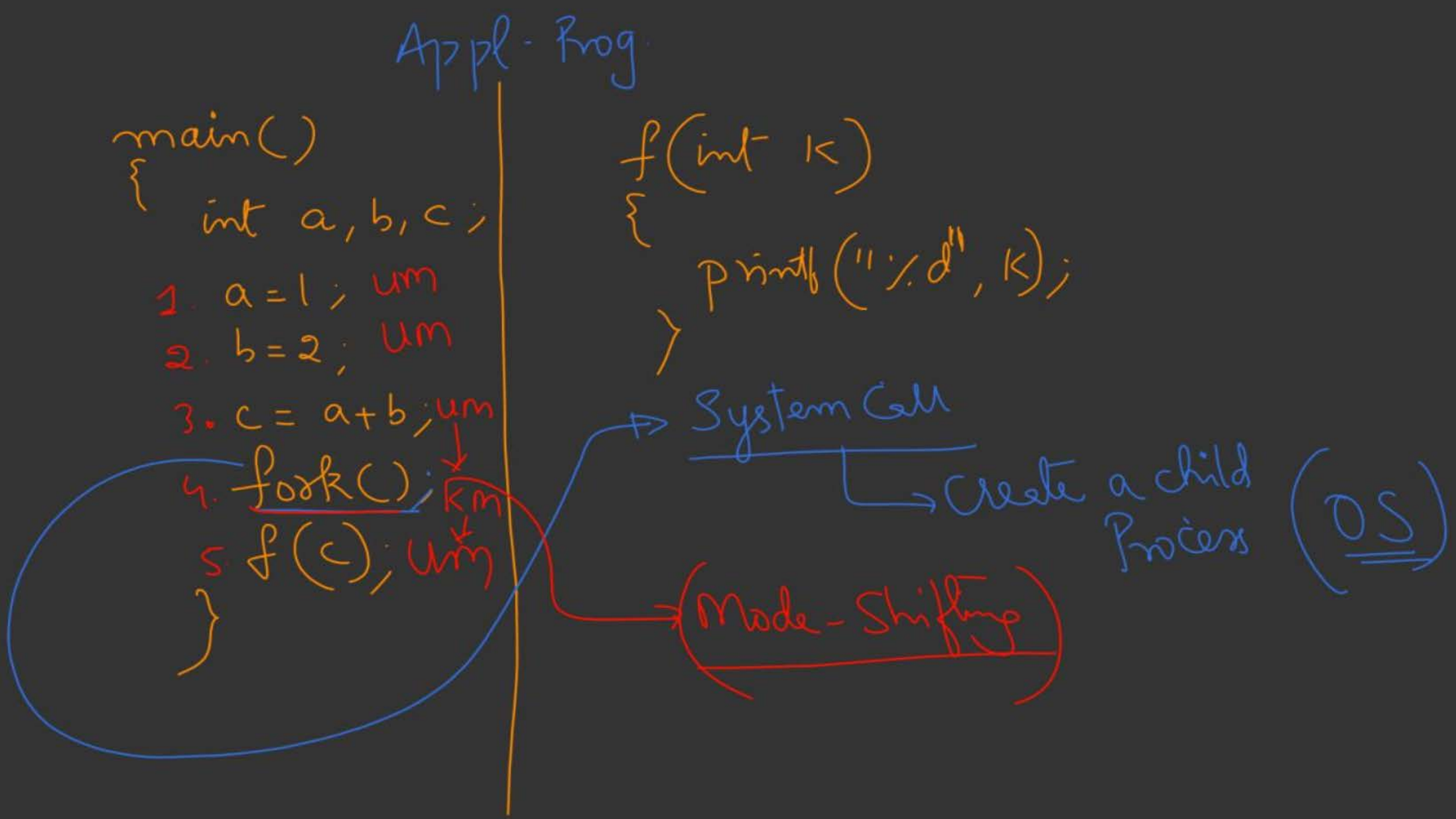
↓

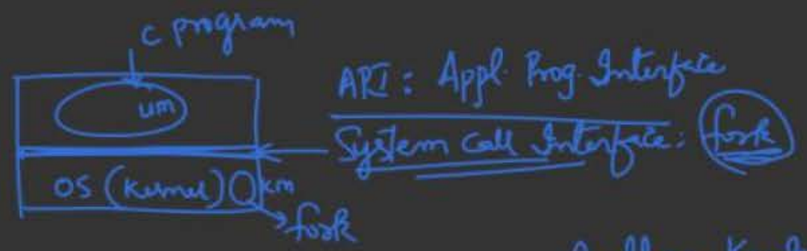
Predefined / In-built / User Defined Library

```
f(int k)
{
UM printf("%d",k);
```

→ All user-defined & Predefined fn's execute (Start) in UM.

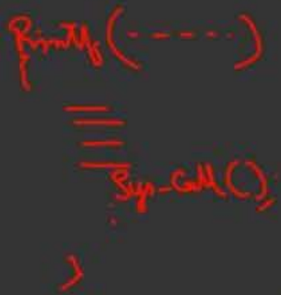
(*)


Appl. Prog
main()
{
int a, b, c;
1. a=1; um
2. b=2; um
3. c= a+b; um
4. fork(); km
5 f(c); um
}
f(int k)
{
printf("%d", k);
}
System Call
Create a child Process (OS)
Mode-Shifting



API : Appl. Prog. Interface

System Call Interface : fork



" User-Defined & Pre-Defined functions may also activate/use System Calls "

1) Goals of O.S

2) Types of OS
   - U. Pr
   - M. Pr
     - N. Pr
     - Pr

3) Arch. Support
   - → IO } DMA
   - → Disk }
   - → Memory : Addr. Translation
   - → CPU : Dual Mode opn

4) Ideology of mode shifting:

THANK YOU!