CS & IT ENGINEERING

Operating Systems

Introduction & Background

Lecture No. 1

By- Dr. Khaleel Khan sir

- Introduction to Course
- OS Definition
- Functions; Goals ; Types of OS

ABOUT ME

Hello, I'm Dr. Khaleel Ur Rahman Khan.

1. Ph.D. in Computer Science.
2. Professor in Computer Science.
3. Has more than 28 Years of Experience in Teaching at Engineering Colleges.
4. Published more than 50 journal articles in the areas of Wireless Networks.
5. Seven candidates have been awarded PH.D. under his Supervision.
6. Has more than 22 years of Educating and Mentoring the GATE Aspirants.

By- Dr. Khaleel Khan sir

Operating Systems

Lecture schedule

I. Introduction & Background

- 1.1 What is Operating System
- 1.2 Function & Goals of Operating System
- 1.3 Types of Operating system
- 1.4 Multiprogrammed Operating System
- 1.5 Architectural requirements for multiprogrammed OS
- 1.6 Mode Shifting in Multiprogrammed OS
- 1.7 System Calls
- 1.8 Fork System Call
- 1.9 Problem Solving

II. Process Management

- 2.Process Concepts
  - 2.1 program Vs Process
  - 2.2 Process as ADT
  - 2.3 Process State Transition Diagram
  - 2.4 Schedulers & Dispatchers
  - 2.5 Problem Solving

3.CPU Scheduling

- 3.1 Need For Scheduling & Scheduling Criteria
- 3.2 Process Times
- 3.3 Scheduling Algorithms
  - 3.3.1 FCFS
  - 3.3.2 SJF
  - 3.3.3 SRTF
  - 3.3.4 LRTF
  - 3.3.5 Priority
  - 3.3.6 Round Robin
  - 3.3.7 Multilevel Queue Scheduling
- 3.4 Problem Solving

4. Multithreading

- 4.1 Thread Concept & Benefits
- 4.2 Types of Threads
- 4.3 Thread Issues
- 4.4 Thread Libraries
- 4.5 Problem Solving

5.Process Synchronization/Coordination

- 5.1 What is IPC & Synchronization
- 5.2 Types of Synchronization
- 5.3 Critical Section Problem
- 5.4 Requirements of CS Problem

5.5 Synchronization Mechanism

- 5.5.1 Lock Variables
- 5.5.2 Strict Alternation
- 5.5.3 Peterson Solution
- 5.5.4 Synchronization Hardware
- 5.5.5 Semaphores
- 5.5.6 Monitors

5.6 Classical IPC Problems

- 5.6.1 Producer Consumer Problem
- 5.6.2 Reader-Writer Problem
- 5.6.3 Dining Philosopher Problem

5.8 Concurrency Mechanisms

- 5.8.1 Parallel Construct
- 5.8.2 Fork & Join Statement

5.10 Problem Solving

6. Deadlocks

- 6.1 Concepts of Deadlock
- 6.2 System Model
- 6.3 Deadlock Characterizations
  - 6.3.1 Necessary conditions
  - 6.3.2 Resource Allocation Graph

6.4 Deadlock Handling Strategies

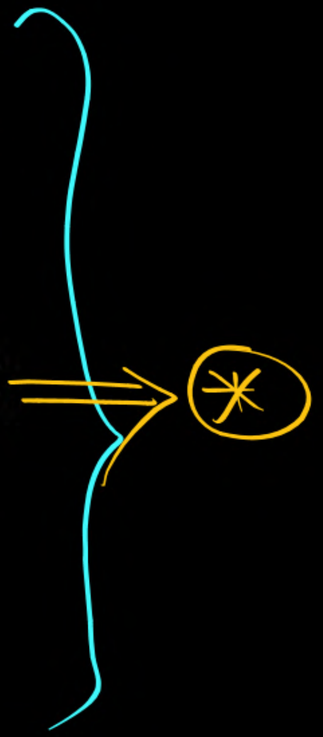
- 6.4.1 Prevention
- 6.4.2 Avoidance
  - 6.4.2.1 Bankers Algorithm
- 6.4.3 Detection & Recovery
- 6.4.4 Deadlock Ignorance
- 6.5 Problem Solving

III Memory Management

7. **Abstract View of Memory**
8. **Loading vs Linking**
9. **Address Binding**
10. **Memory Management Techniques**

- 10.1 Swapping
- 10.2 Partitioning
  - 10.2.1 Fixed Partitions
  - 10.2.2 Variable partitions

- Non Contiguous Allocation
  - 11.3.1 Simple Paging
  - 11.3.2 Paging With TLB
  - 11.3.3 Hashed Paging
  - 11.3.4 Multilevel Paging
  - 11.3.5 Inverted Paging
  - 11.3.6 Shared Paging
  - 11.3.7 Segmentation
  - 11.3.8 Segmented-Paging Architecture



13. Virtual Memory

Concept + Implementation + Performance

IV. File System & Disk Management

14. Physical Structure of Disk

15. Logical Structure of Disk

16. File System Interface

- 16.1 File & Directory Concept
- 16.2 File Attributes
- 16.3 File Operations
- 16.4 Types of Files
- 16.5 Directory Structure

1. Galvin ✓
2. Tanenbaum
3. Stallings

17. File System Implementation

- 17.1 Allocation Methods
- 17.2 Disk Free Space Management Algorithms

19. IO Scheduling(Disk Scheduling)

- 19.1 Need for Disk Scheduling
- 19.2 Disk Scheduling Techniques
  - 19.2.1 FCFS
  - 19.2.2 SSTF
  - 19.2.3 SCAN
  - 19.2.4 LOOK
  - 19.2.5 C-SCAN
  - 19.2.6 C-LOOK
- 20. Problem Solving

(6-8) : Min
(10-12) : Avg
(14-16) : Max

Pre-Requisites

(i) Programming exposure
(ii) Fundamentals of Computers
(iii) (C.O.A) | Data Structures

Operating system

Windows 8

ANDROID

Mac OS

ubuntu

fedora

What is Operating System:

→ Interface between user and H/w ; ✓

→ Resource Manager

Resources < H/w : CPU + Mem + IO
S/w : Files + Semaphores + Pipe

→ Control Program(s)

→ Set of utilities to simplify application development

→ Acts like a Government (Service Provider)

→ System software

Interface / Need :

User / Programmer

$L_1 : \Sigma(a \cdots z)$

O.S

H/W

$L_2 : \Sigma(0,1)$

Primary

Secondary

Memory

C.U

I/P

O/P

A.L.U

Sec Storage

CPU

```
c = a+b;
```

```
Load R1, a
Load R2, b
Add  R1, R2
Store c, R1
```

C.U

Kernel

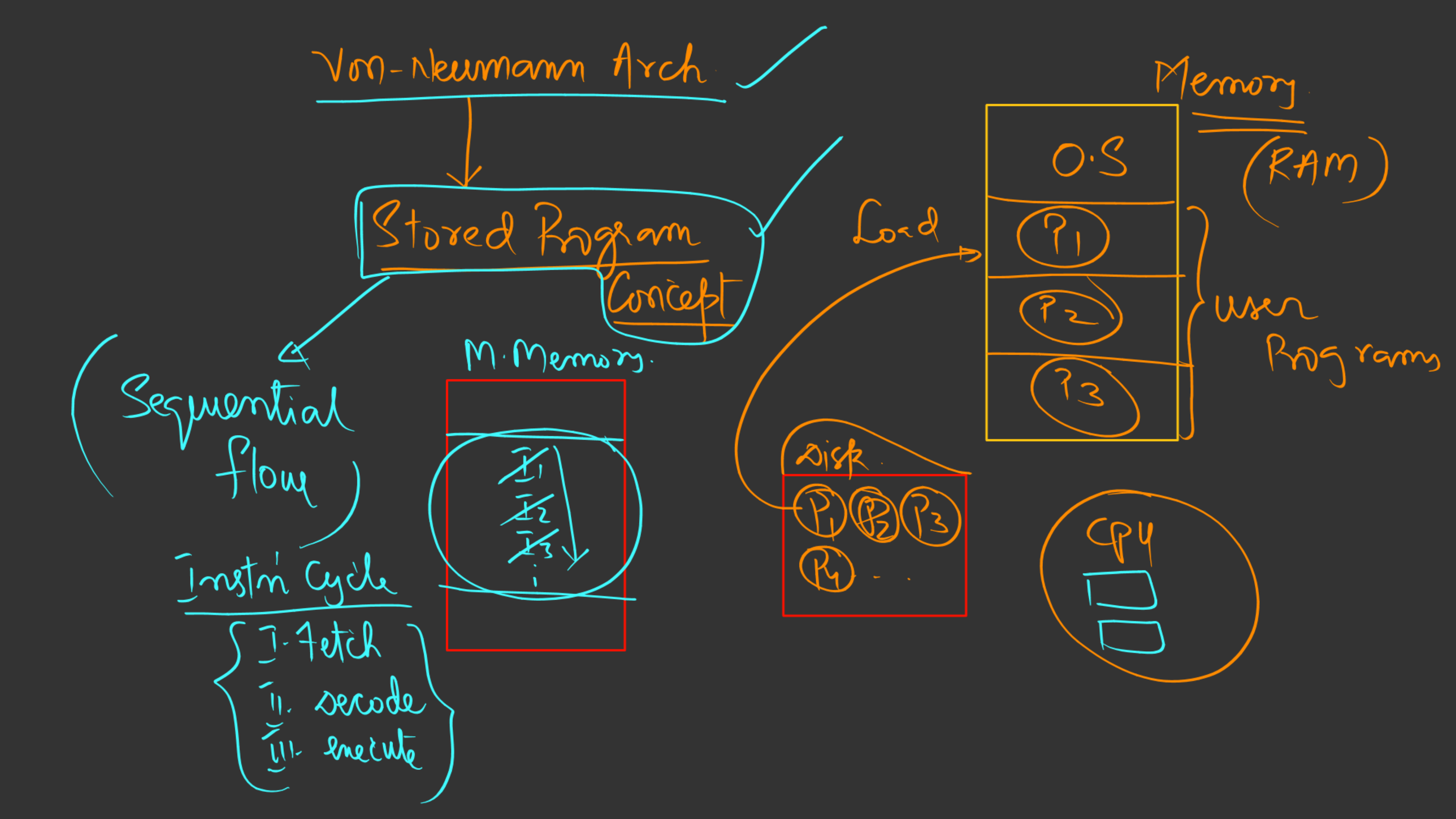
Timing Signal

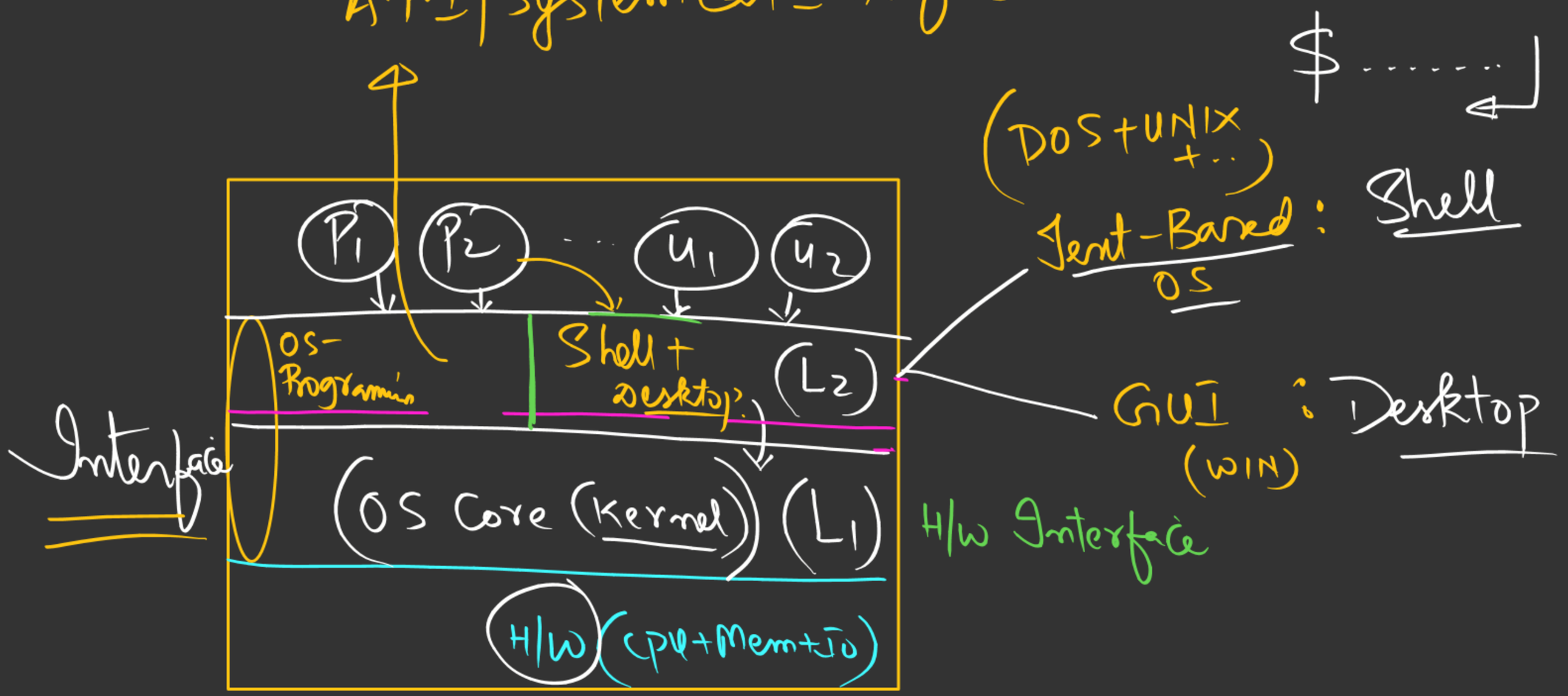
μ-oprns

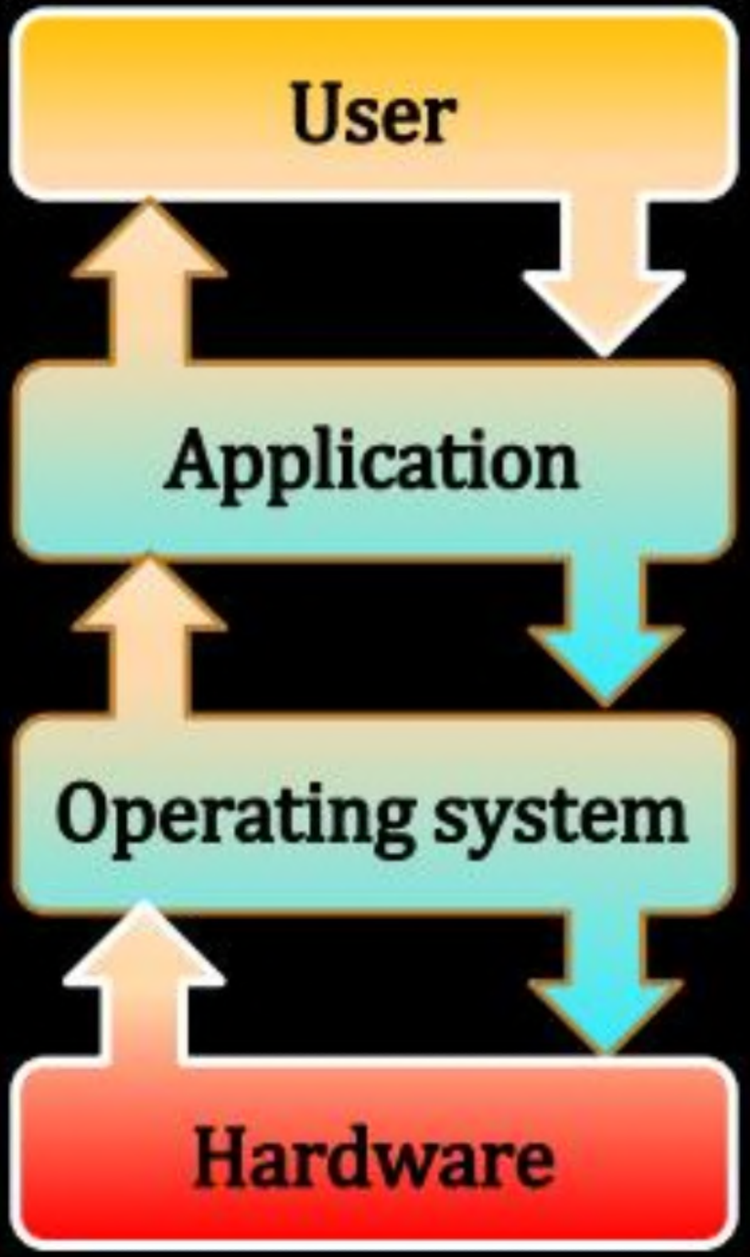
(oprns carried out on the data in Registers)

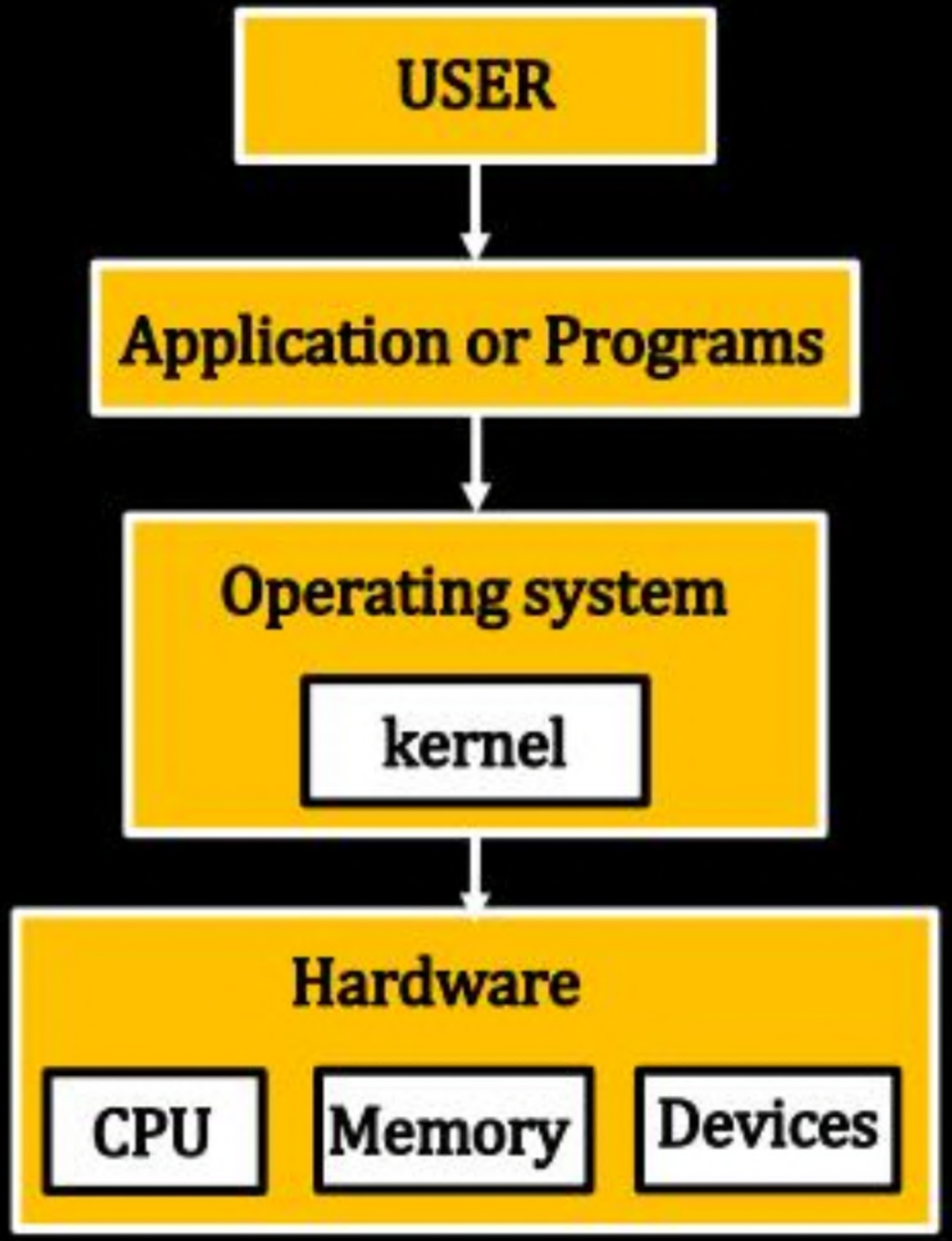
Von-Neumann Architecture

Harward Arch.


Von-Neumann Arch
Stored Program Concept
Sequential flow
Instn Cycle
I. fetch
II. decode
III. execute
M. Memory
I1
I2
I3
Load
Memory
(RAM)
O.S
P1
P2
P3
User Programs
Disk
P1
P2
P3
P4
CPU


A.P.I | System Call Interface
P1
P2
U1
U2
OS-Programs
Shell + Desktop
L2
OS Core (Kernel)
L1
H/W (cpu+Mem+Io)
Interface
H/W Interface
$ .....
(DOS+UNIX+..)
Text-Based OS : Shell
GUI (WIN) : Desktop


User
Application
Operating system
Hardware


USER
Application or Programs
Operating system
kernel
Hardware
CPU
Memory
Devices

Functions of Operating system

Functions & Goals of O.S : (user-centric) Primary

Computing domain → Goals

Distributed Systems

1. Convenience [Easy to use]
2. Efficiency [effective utiliz. of resources]
3. Robustness
4. Reliability

---

5. Portability
6. Scalability [Ability to evolve]

Shell → UNIX / C_1

GUI → WIN / C_2

Real Time System : Deadline Constraints (Vxworks)

Types of O.S:

1. 1st Gen : 1930-40's : (Punch Cards) No O.S [Manual]
2. 2nd " : 19-40-50's : (Mag Tape) [No O.S]
3. 3rd Gen : 1950-60's :

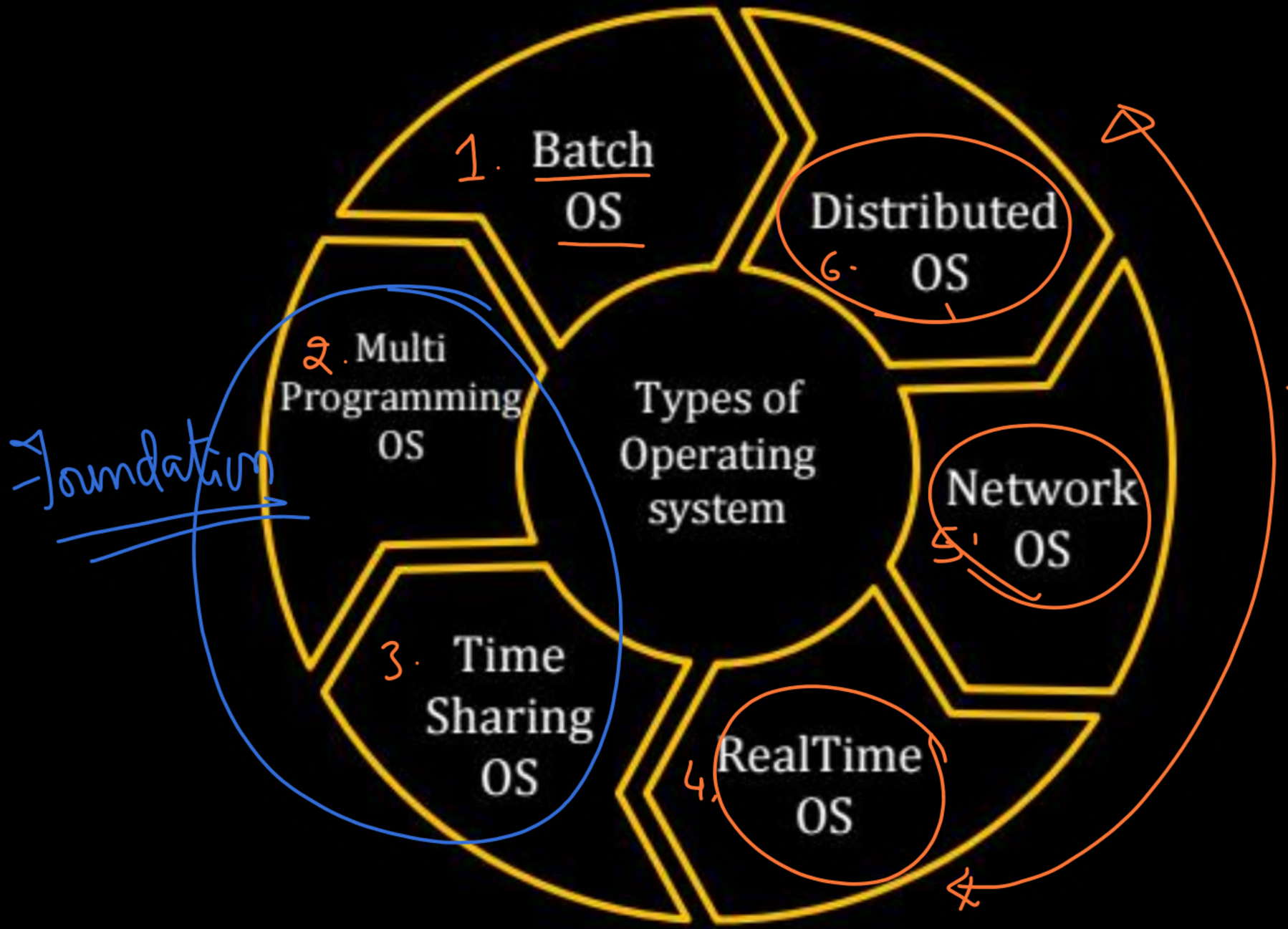
- UNI-PROGRAMMED
  - DOS
- MULTI PROGRAMMED
  - (UNIX + WIN)
  - MACH

↓

Mag. disk

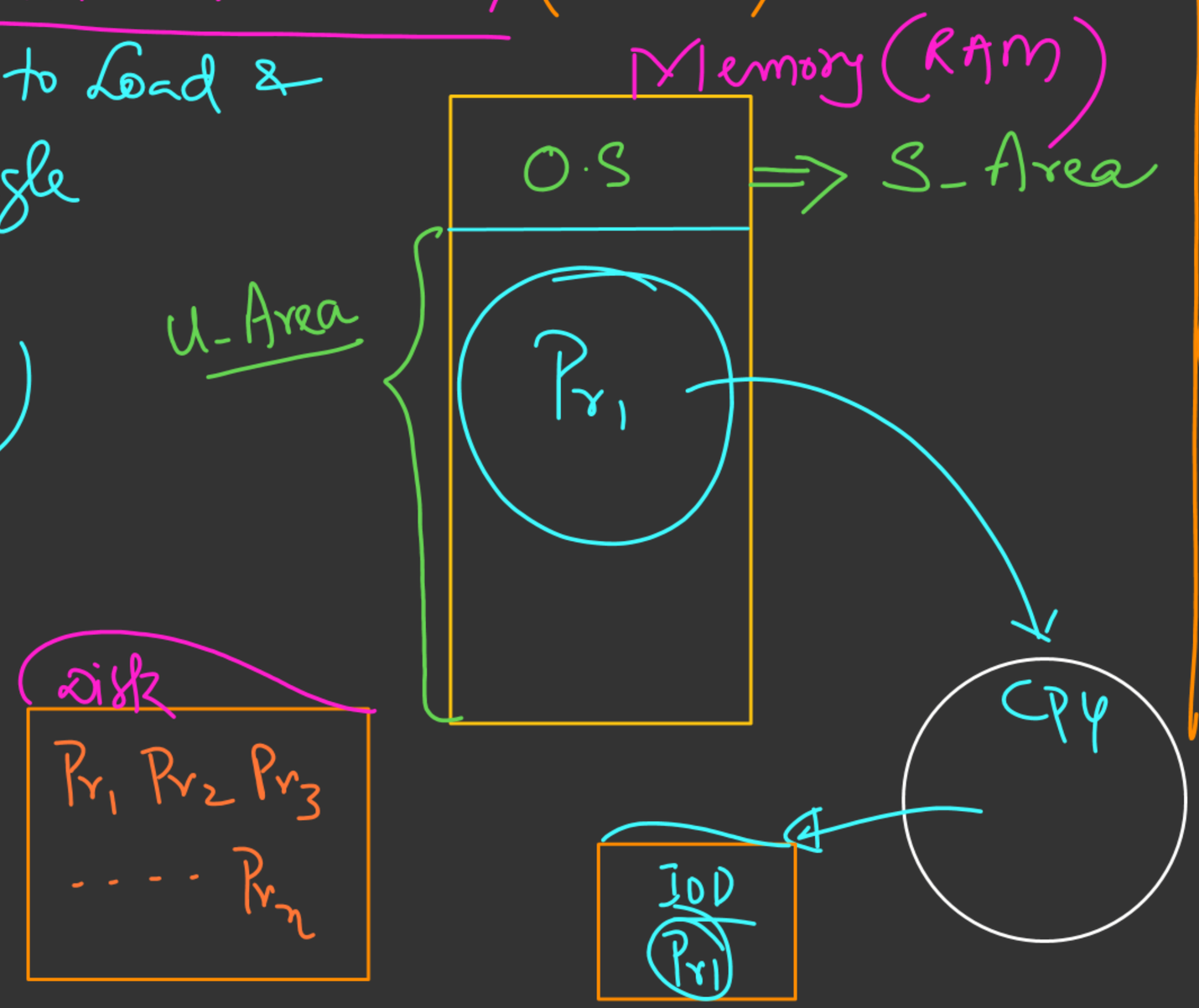
- Hard disk
- Floppy disk
  - $3\frac{1}{2}$
  - $5\frac{1}{4}$

:


1. Batch OS
2. Multi Programming OS
3. Time Sharing OS
4. RealTime OS
5. Network OS
6. Distributed OS
Types of Operating system
Foundation
Adv. O.S
Modern OS

1. UNIPROGRAMMING (U-Pr)

(: Ability of O.S to Load & Manage a Single Program in Memory)



Limitation of U-Pr

(is Idleness of CPU)


THANK YOU
GW
SOLDIERS !