

# CS & IT ENGINEERING

## OPERATING SYSTEM

Intro and Background

Lecture No. 2



By- Dr. Khaleel Khan Sir



# TOPICS TO BE COVERED

Architectural  
requirements for Multi  
Programmed OS

Mode of shifting in Multi  
Programmed OS

System Calls

# Operating system



Windows 8



ANDROID

Linux™



Mac OS



ubuntu



fedora



What is an O.S

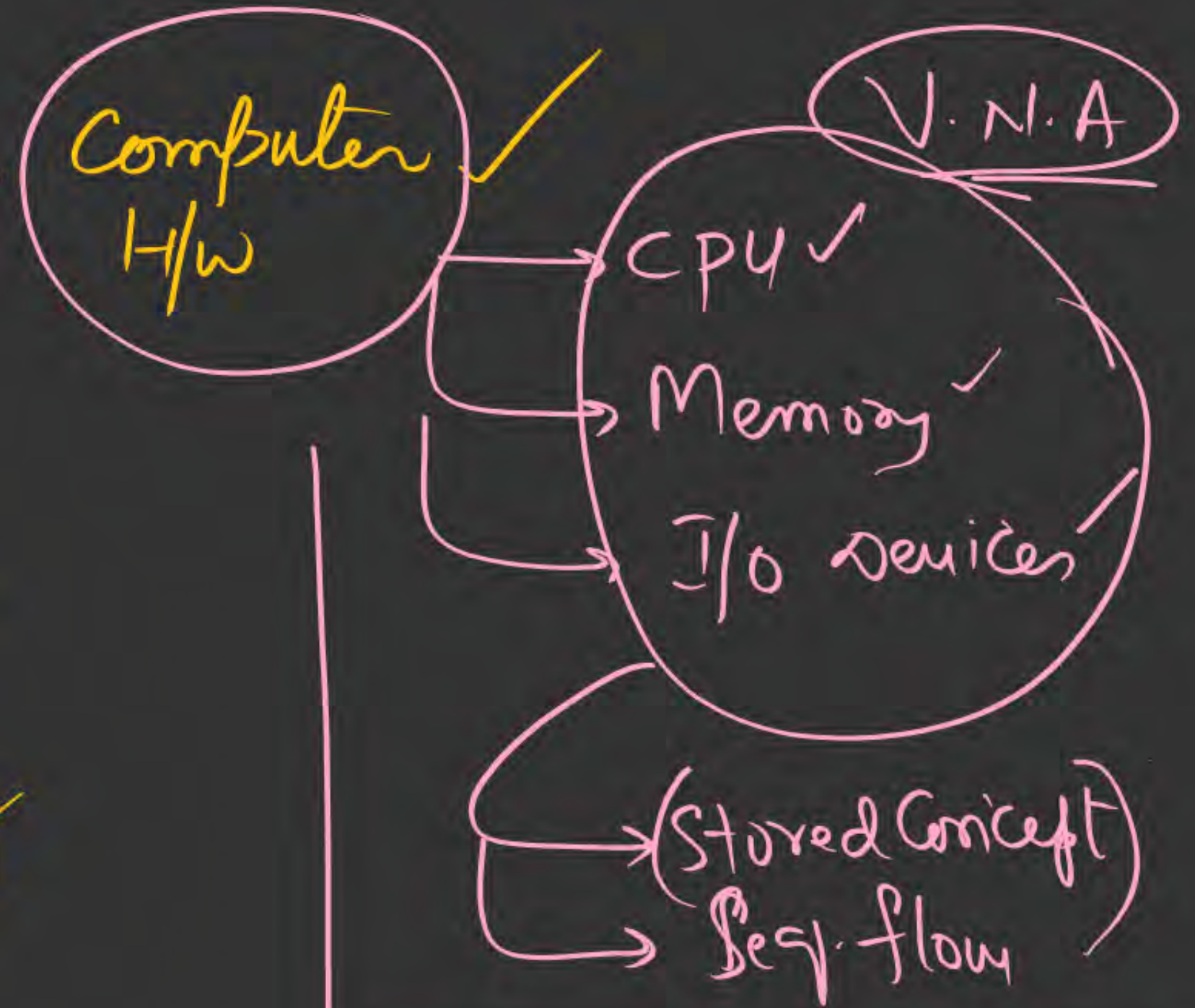
1) Interface b/w user & Programmer

2) Resource Manager

3) Control Program(s) / Sys S/w

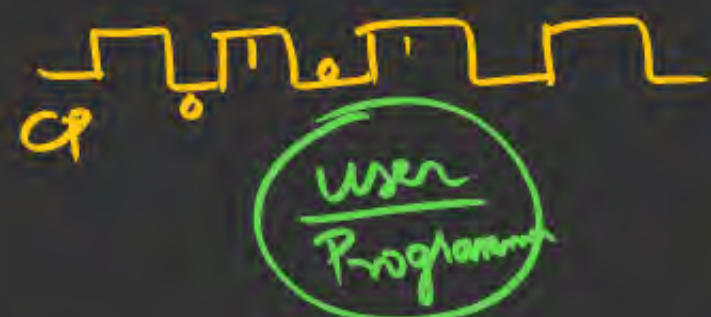
4) Set of utilities to simplify appl. development

5) Acts like a Government



- (ii) Harvard Arch
- (iii) R.T. Architecture
- (iv) Mobile Phone





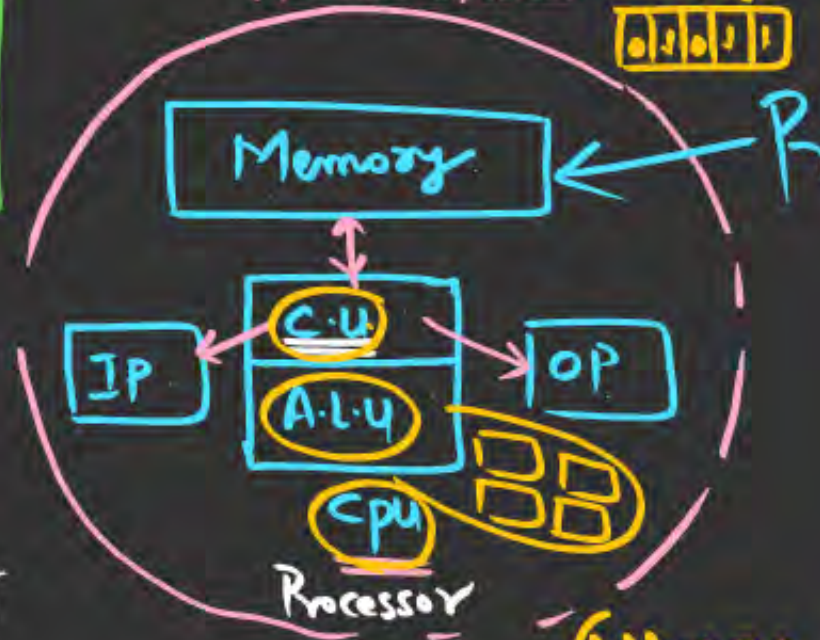
0  
\$

H/W

CU: Control unit

Architecture

Reg:  
0 1 0 1 1



VON-NEUMANN

(Address + Subtracting)

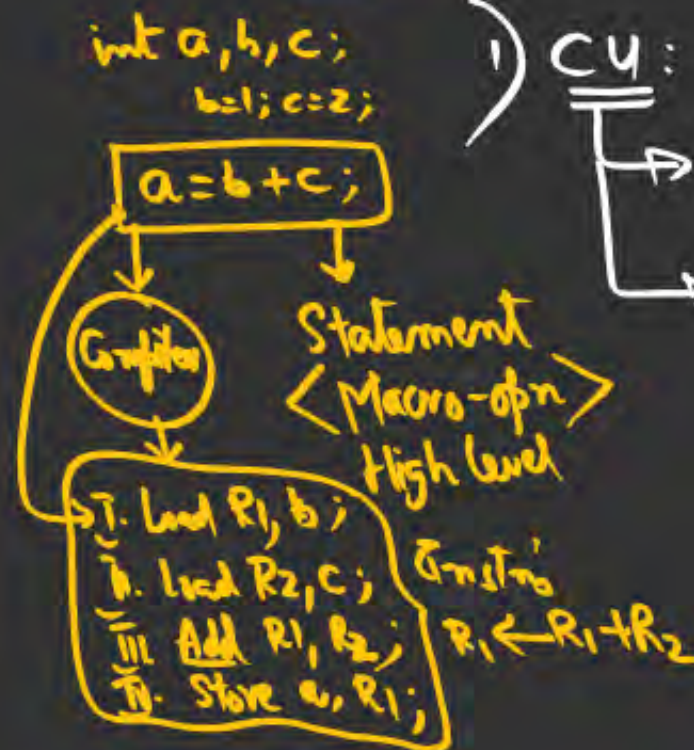
(ALU: Arithmetic Logic Unit  
<functional unit>)

Timing/Control Signals:

Sequencing/Execution of

Micro-ops:

Ops that are carried out on the data stored in registers (CPU)



Note: As per Von-Neumann Architecture, All Sec. Storage devices are part of IP/OP Devices

Memory

Primary

Primary

< Main Physical >

(RAM + ROM + Cache + Registers)

→ Volatile ✓  
→ Faster to access  
(ns)  $\times 10^{-9}$   
→ Smaller in size

Secondary

(Auxiliary Hard disk + Pen-drives + Tapes DVD's)

- Non volatile  
- Slower to access  
(ms)  $\times 10^{-3}$   
→ larger size



Stored Program Concept

How are Programs executed in Von-Neumann Computer

$a = b + c$

Test.c

Compiler

• .exe  
I<sub>1</sub>  
I<sub>2</sub>  
I<sub>3</sub>  
⋮  
I<sub>10</sub>

OS

Memory (RAM)

I<sub>1</sub>  
I<sub>2</sub>  
I<sub>3</sub>  
⋮  
I<sub>10</sub>

Test.exe Program in Memory

Load Store

CPU  
I<sub>4</sub>

disk  
Test.c  
Test.exe

CPU cannot execute Programs (.exe) directly from disk

Instructions of the Program are executed Sequentially

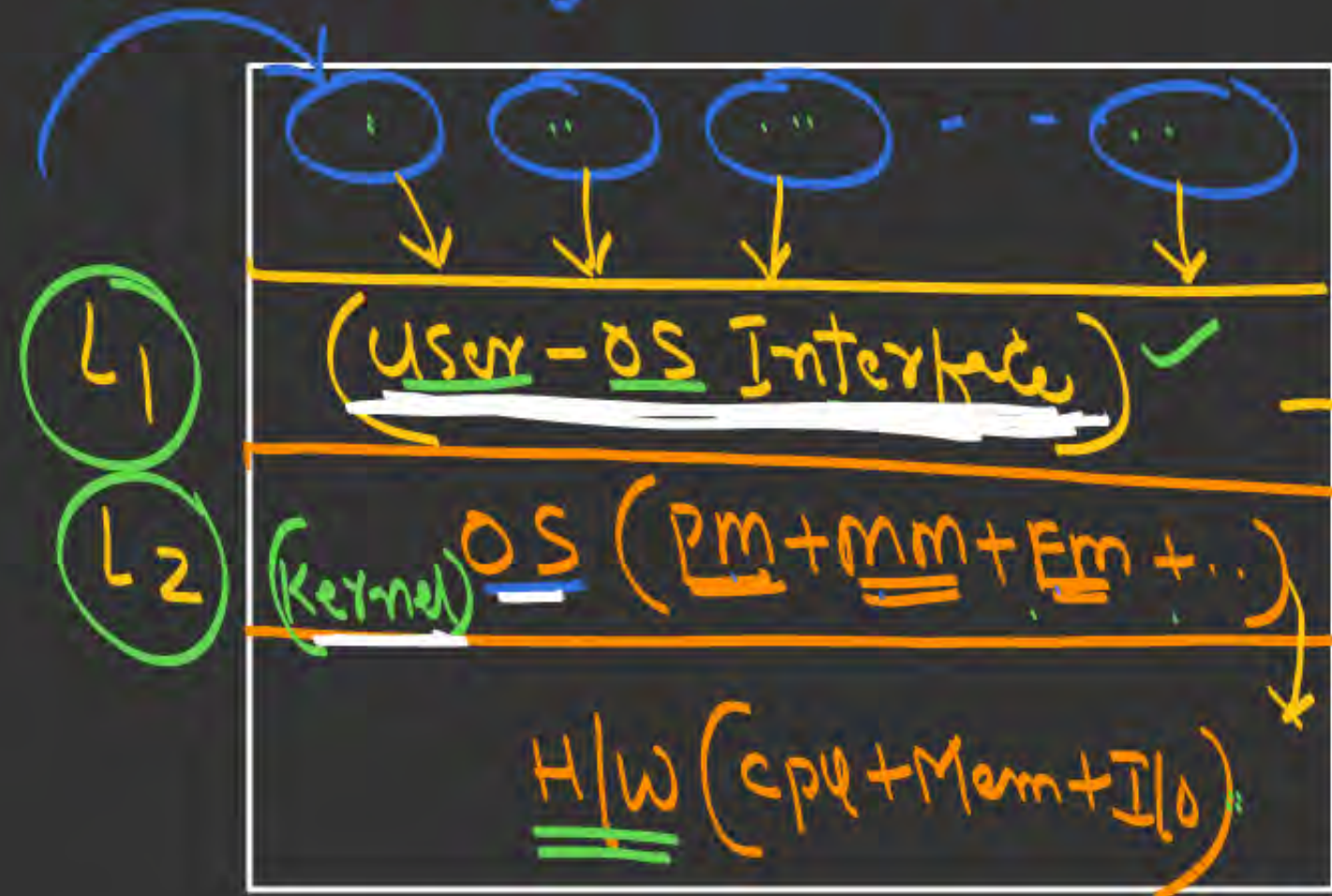
1. S.P.C  
2. Seq. Flow





$$L_1 = \sum (a - 3)$$

users / Programmers



Kernel  
Nucleus

Interface <  $L_1$   
 $L_2$



$$L = \sum (\overline{0}, 1)$$

010101...

Process Manager (cpu)

Memory " (memory)

File " } (devices)  
service "

Protection " (Resources)

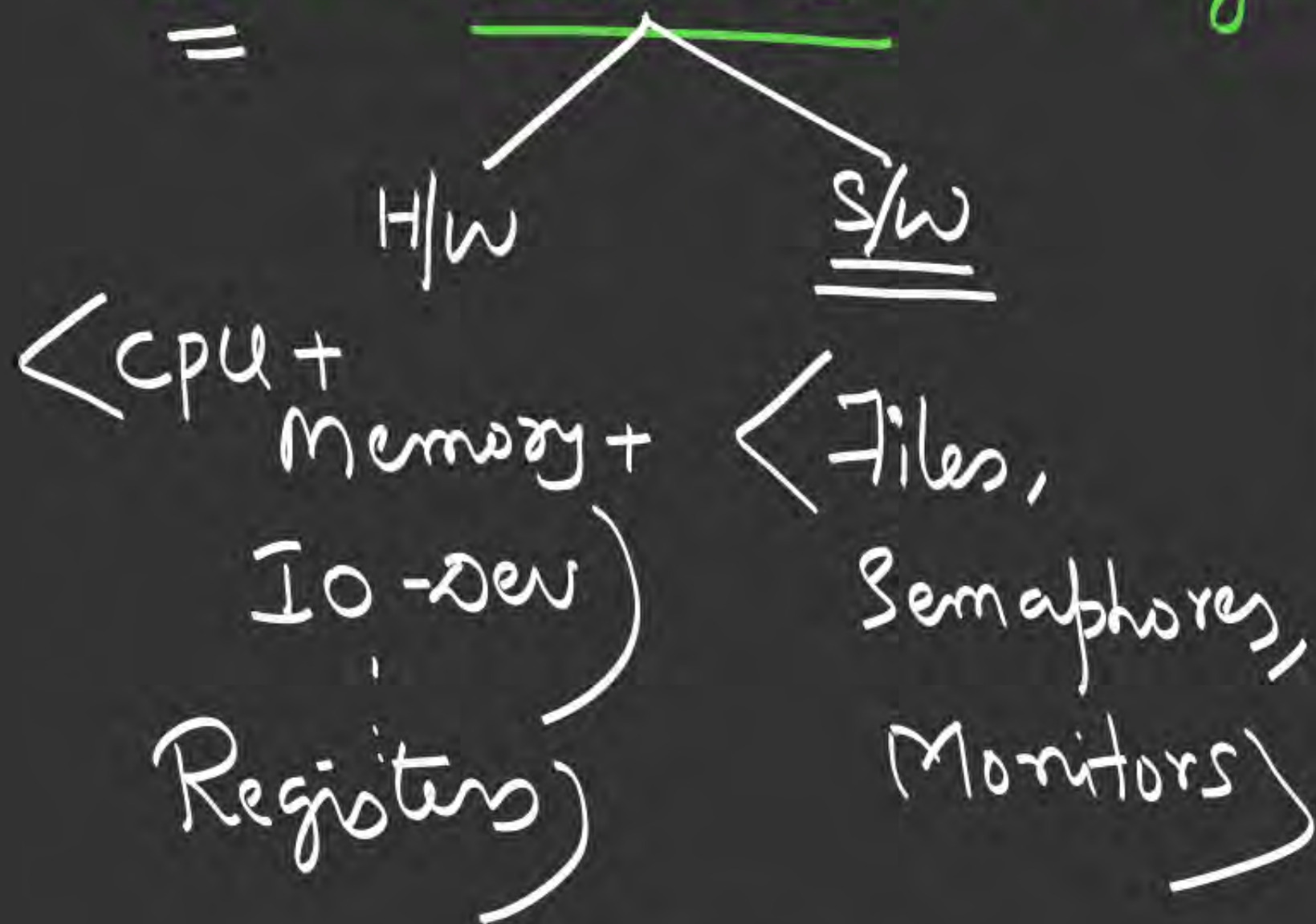
Command  
Interpreter

Shell (c: >)

GUI (desktop)



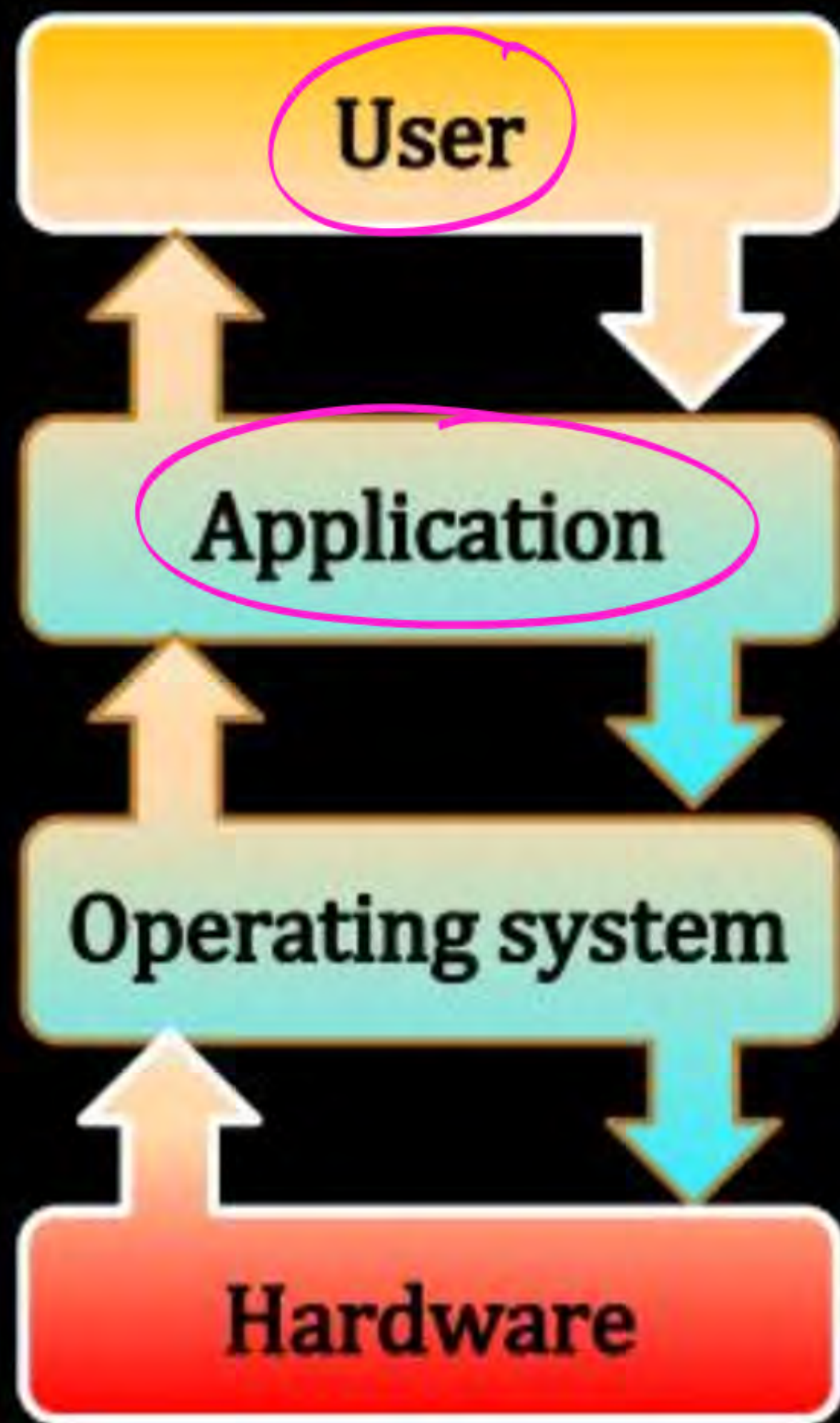
OS → Resource Manager (Allocation, deallocation,  
Protection, Free resources)



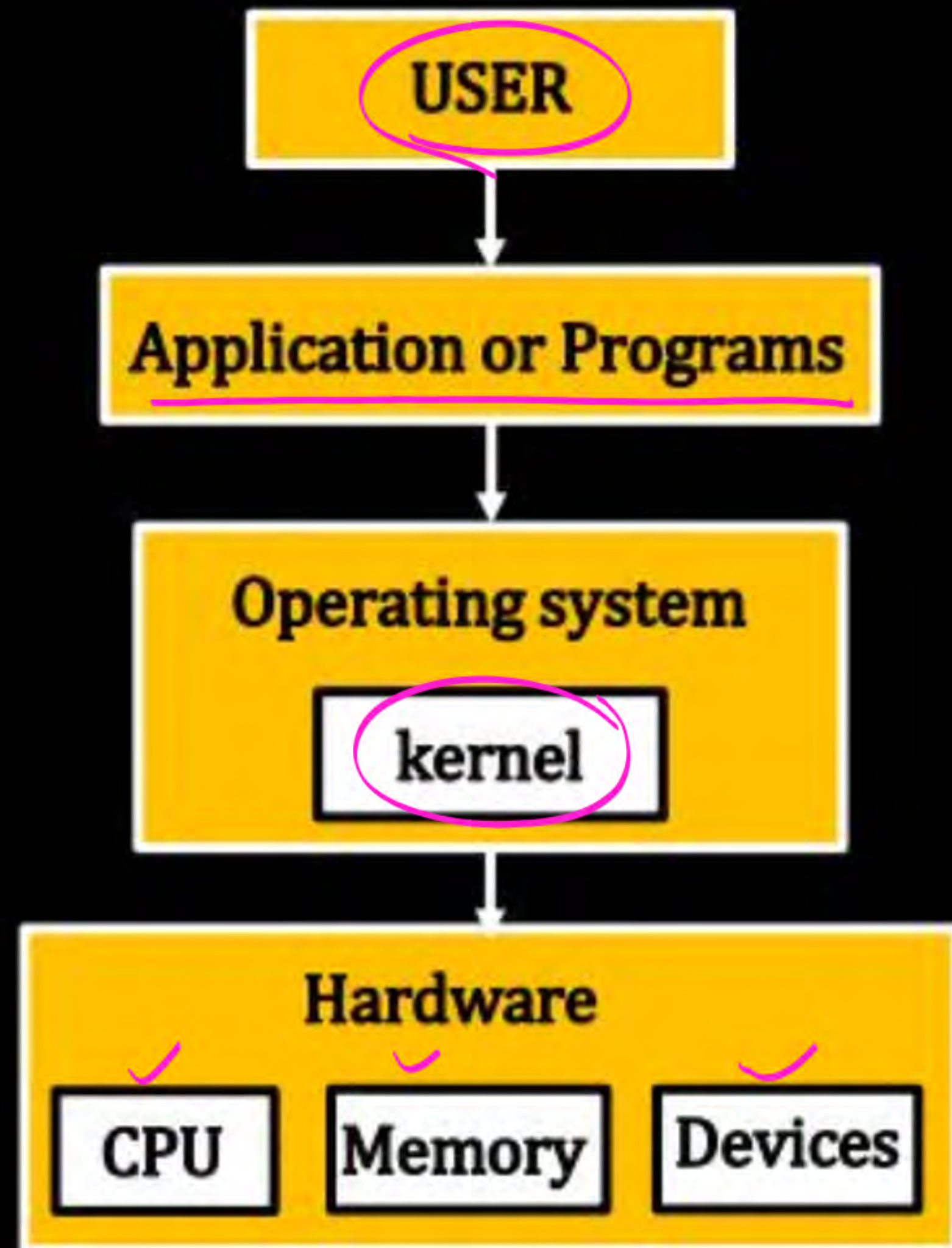
Functional

- Control Program(s)/S/w
- Set of utilities to simplify application development
- Acts like a Govt.





Kernel.





## Functions & Goals \*

→ is something that is to be carried out to achieve the Goal

- Convenience (user friendly) ✓
- Efficiency (Effective utiliz. of Resources)
- Reliability ✓
- Robustness (strong enough to bear errors)
- Scalability (Ability to evolve)
- Portability

(UNIX vs WINDOWS) ✓  
↓  
(Easy to use)

→ Primary (Main)  
Goal of an  
O.S. ?

Convenience  
for (Personal Computing  
desktop " )



→ Convenience is NOT Primary Goal of all other Computing Environment.

## Other Computing Environment

a) Real Time System (RTS) : (efficiency)

→ operates in Real time (Strict deadlines)

Destructions

Ex:

Missile Control, Satellite Control, Nuclear System, A.T.C

Hard-R.T

→ ATM

Soft R.T.S

NOT Strict

No loss



# Functions of Operating system





