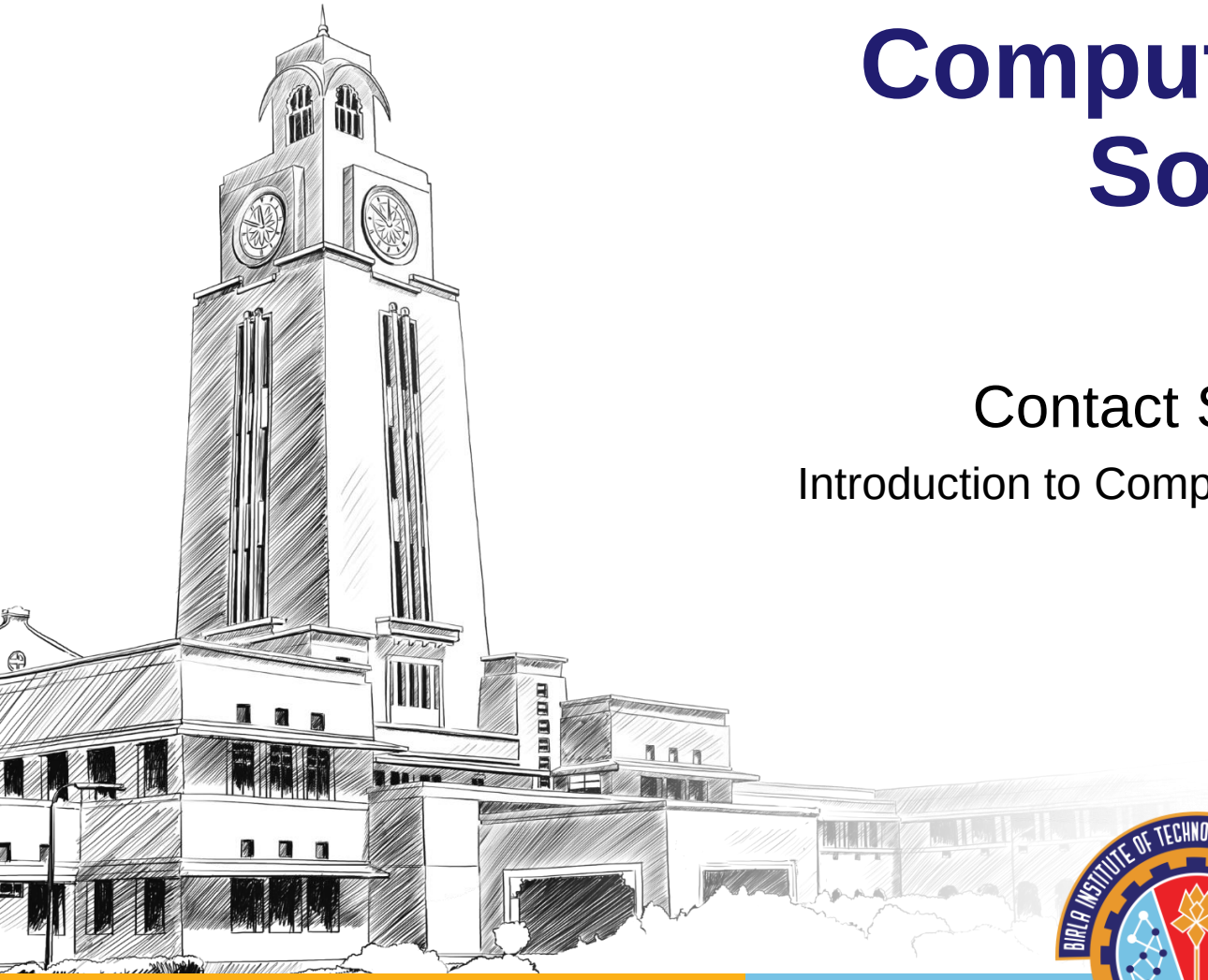


Computer Organization and Software Systems

Contact Session 1

Introduction to Computer Systems

Dr. Lucy J. Gudino

Team

Instructors:

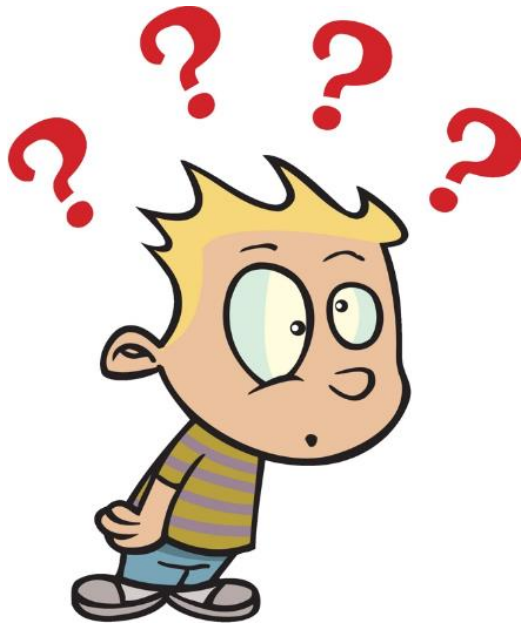
- Dr. Lucy J Gudino (IC)
- Prof. Pradeep H K

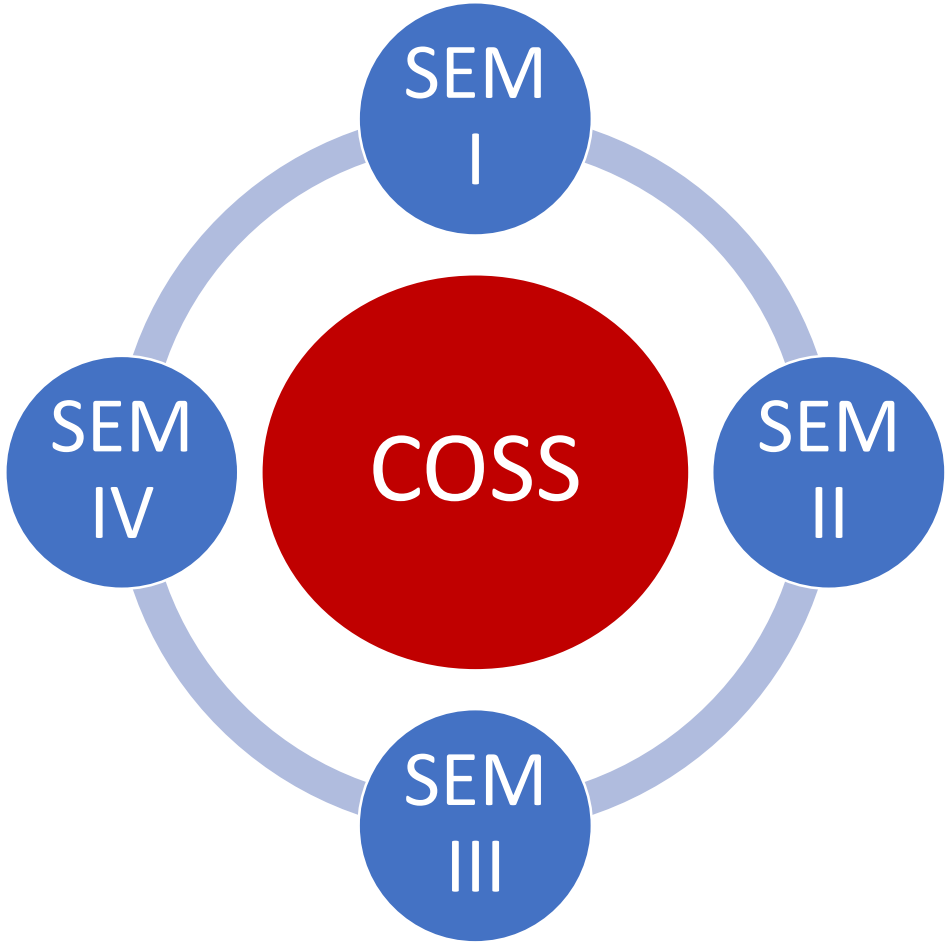
Teaching Assistants:

- VAIBHAV JAIN (Lead TA)
- Usha Govindaiah
- KADAM BHUSHAN VINAYAK
- Selva Kumar S
- H. Madhusudan Rao
- Puneet

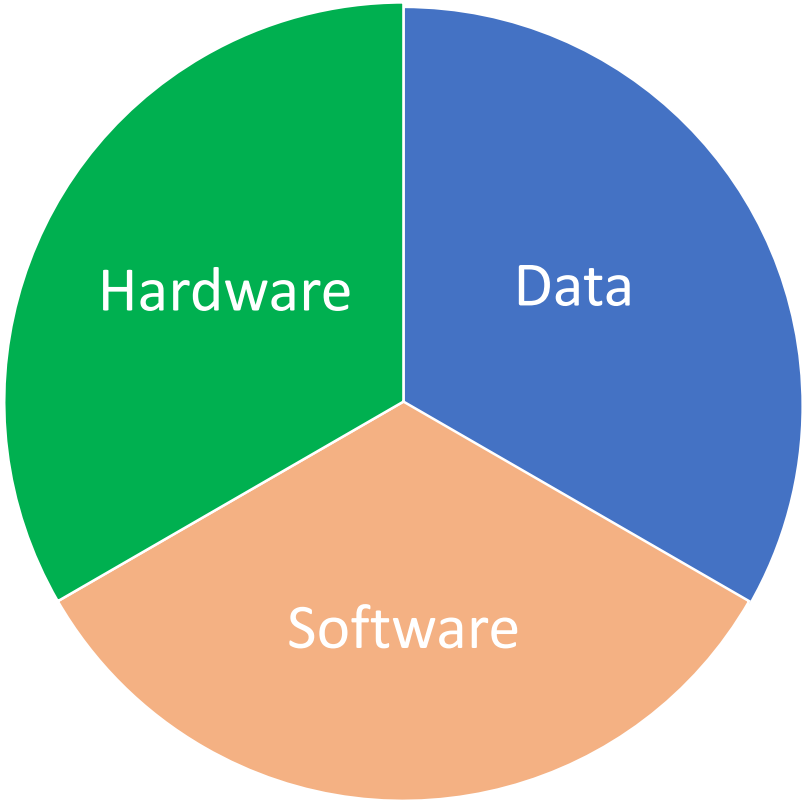
Introduction

- **Why Study *COSS*?**





Introduction



Data analytics: is the process of examining **data** sets in order to draw conclusions about the information they contain, increasingly with the aid of **specialized systems** and **software**.

Three courses

- Computer Organization and Software Systems (Core Course)
- Systems for Data Analytics (Elective)
- Big Data Systems (Elective)

Benefits

- Understanding System Architecture
- Efficient Code Implementation
- Performance Optimization
- Memory Management
- Parallel Computing
- System-level Troubleshooting
- Integration with Existing Systems
- Scalability Considerations
- Resource Utilization

reg
global
local
reg short int count;
4 bytes
1 byte

```
for (count = 1; count <= 10; count ++)
        printf("Count = %d", count);
```

Text Books and Reference Books

Text Books:

(T1) W. Stallings, Computer Organization & Architecture, PHI, 10th ed., 2010.

(T2) A Silberschatz, Abraham and others, Operating Systems Concepts, Wiley Student Edition, 8th Edition

Reference Books:

(R1) Patterson, David A & J L Hennenssy, Computer Organization and Design – The Hardware/Software Interface, Elsevier, 5th Ed., 2014.

(R2) Randal E. Bryant, David R. O'Hallaron, Computer Systems – A Programmer's Perspective, Pearson, 3rd Ed, 2016.

(R3)Tanenbaum, Modern Operating Systems: Pearson New International Edition, Pearson Education, 2013 (Pearson Online)

(R4)Stallings, Operating Systems: Internals and Design Principles : International Edition, Pearson Education, 2013 (Pearson Online)

Evaluation Scheme

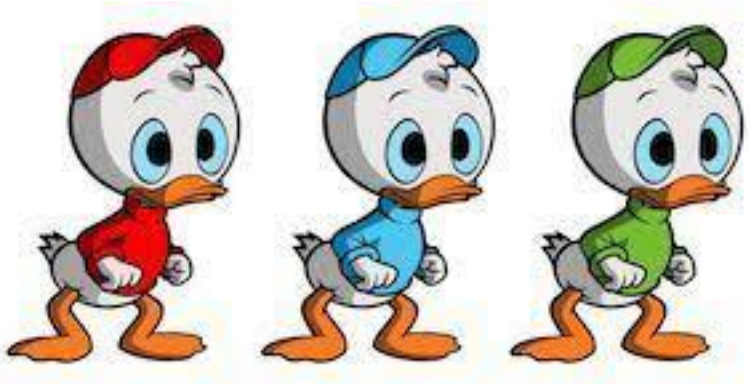
- 5 unit course.

Sl No.	Evaluation Component	Weightage %	Nature of Component
1	Mid Sem Exam	30%	CB
2	Comprehensive Examination	40%	OB
3	Quiz	5% (Two quizzes – Best of two)	OB
4	Assignments	25%	OB

Assignments

- Two assignments:
  - One pre-midsem exam : 10%
  - One post-midsem : 15%
- Lab based
- Simulator to be used : CPU-OS simulator
  - Open source tool https://drive.google.com/open?id=12YUK52RQ-JhP0ddj6CD_oifW4sTMbsBl
  - Virtual lab (Platifi)

Assignment should not be



General Instructions

1. Always use note book for writing important points and for solving problems
2. Use chat box for writing subject related questions
3. Do not repeat the questions on chat box. Questions will be answered during last 10 minutes of the session
4. Unanswered questions will be put up on the canvas forum

Today's Session

Contact Hour	List of Topic Title	Text/Ref Book/external resource
1-2	**Introduction to Computer Systems** • Hardware Organization of a computer • Basic uniprocessor architecture • Instruction Cycle State Diagram • Operating System role in Managing Hardware • Running a Hello Program	T1

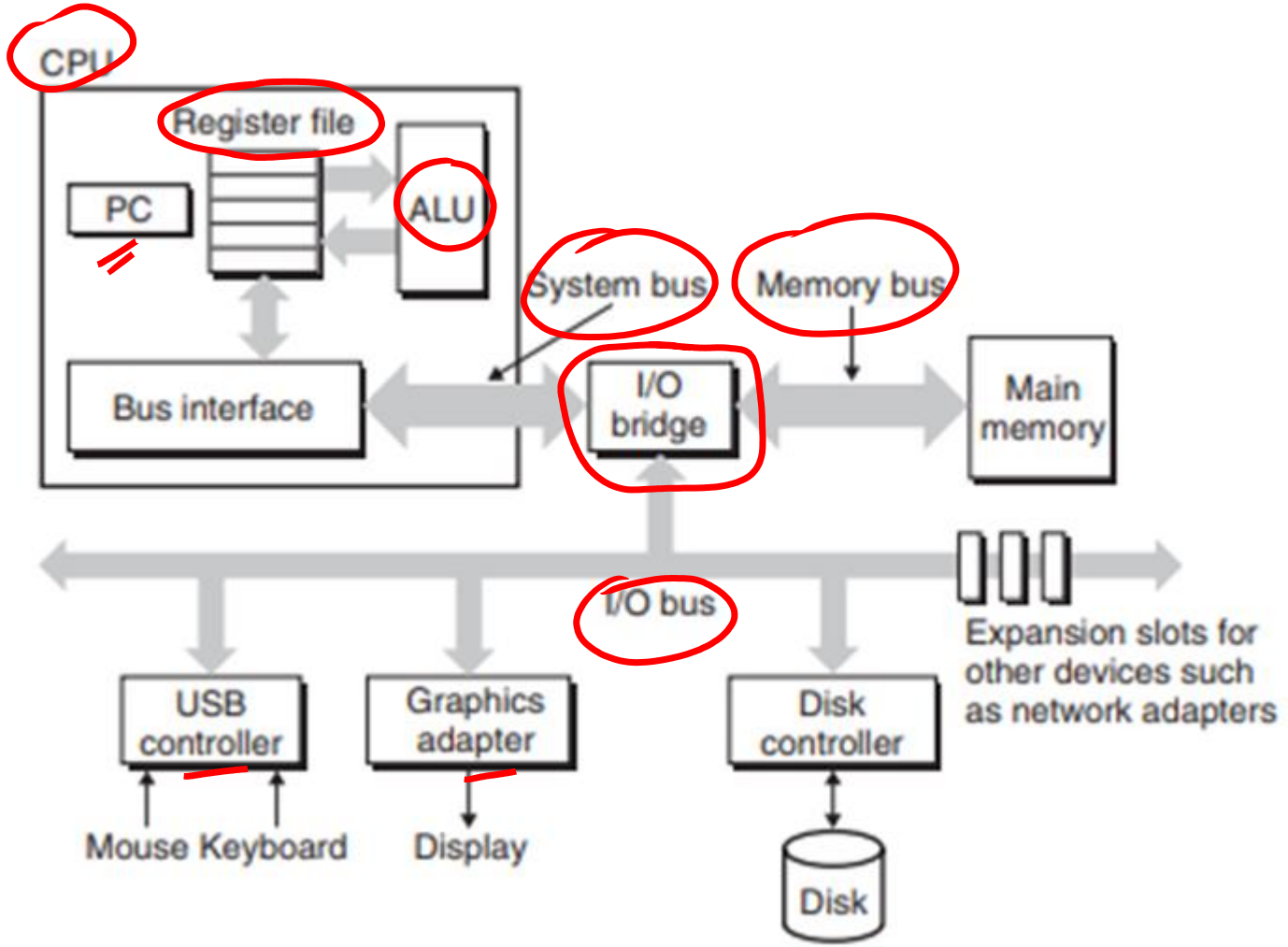
Definition of a Computer

- Is a complex system
- Is a programmable device
- Must be able to process data
- Must be able to store data
- Must be able to move data
- Must be able to control above three functions

Computer System

- **Hardware**
  - Central Processing Unit (CPU)
  - Memory
  - I/O devices
- **Software**
  - System Software
    - System Management Software
    - Tools and Utilities for Developing the software
  - Application Software
    - General Purpose Software
    - Specific Purposed Software

Hardware Organization of a computer

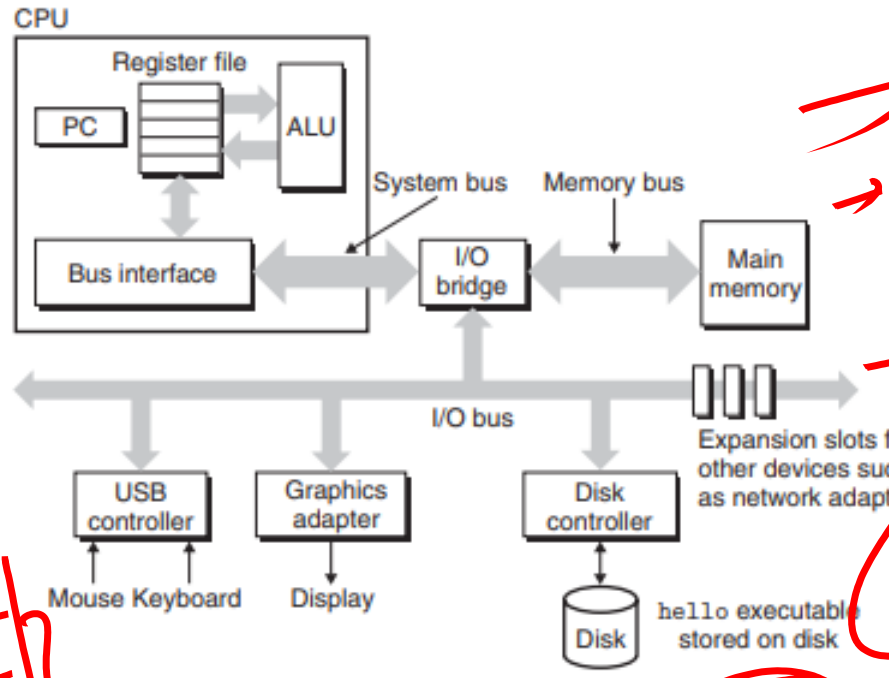


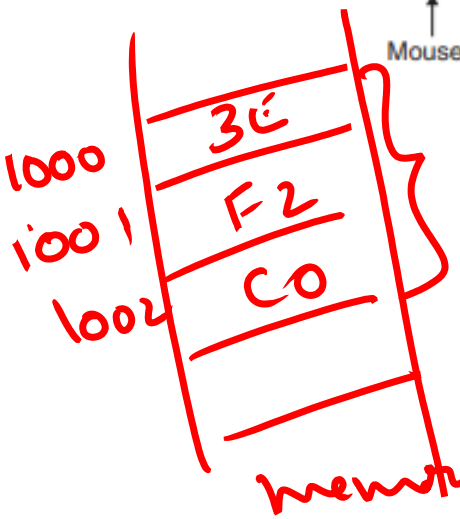
① Von Neumann Architecture

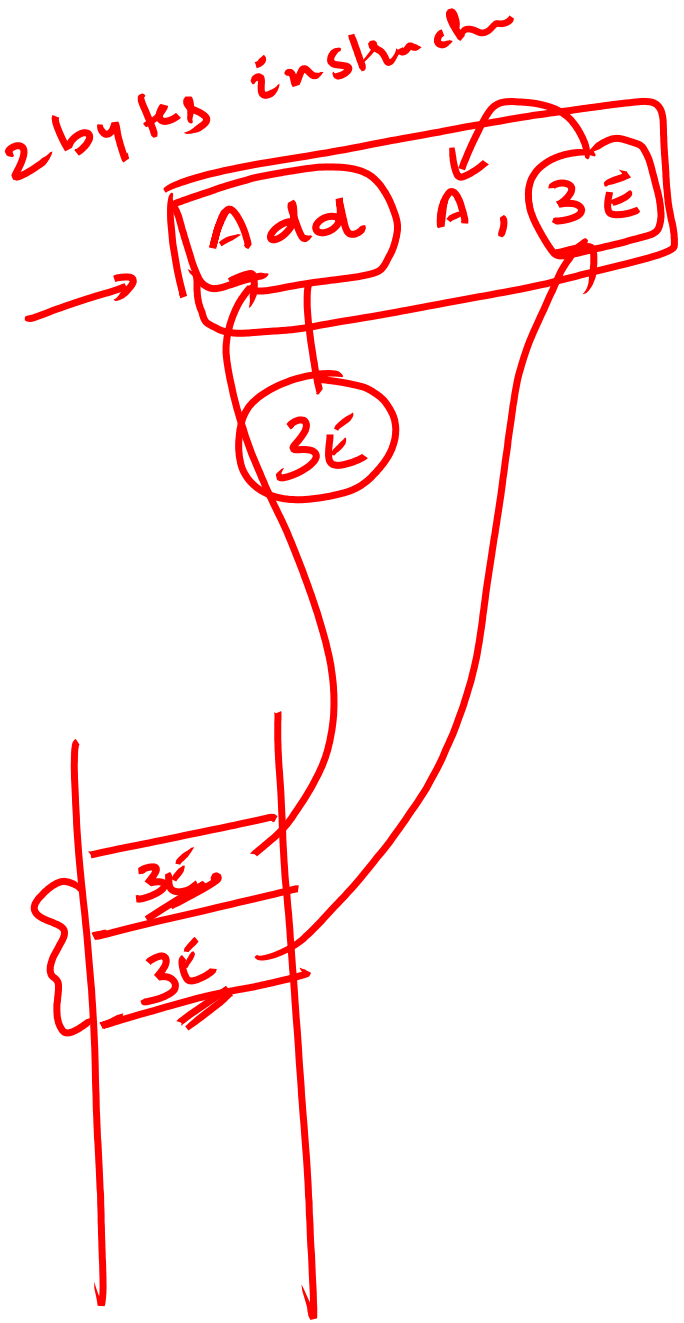
Harvard

└ Stored program Concept

- Three key concepts:
  - Data and instructions are stored in a single read – write memory
  - The contents of this memory are addressable by location, without regard to the type of data contained there
  - Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next





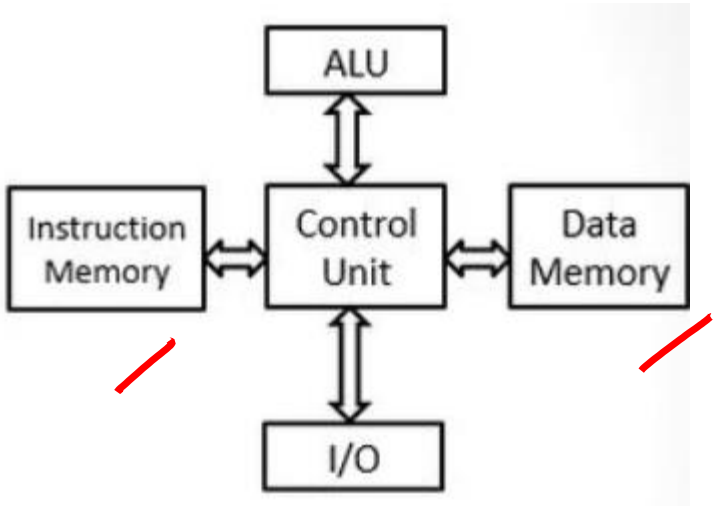

2 bytes instruction
Add
A, 3E
3E
3E
3E

Von Neumann Architecture…

- Stored-program computers have the following characteristics:
  - Three hardware systems:
    - A central processing unit (CPU)
    - A main memory system
    - An I/O system
  - The capacity to carry out sequential instruction processing.
  - A single path between the CPU and main memory.
    - This single path is known as the ***von Neumann bottleneck***.
    - Side effect : reduced throughput (Data Rate)

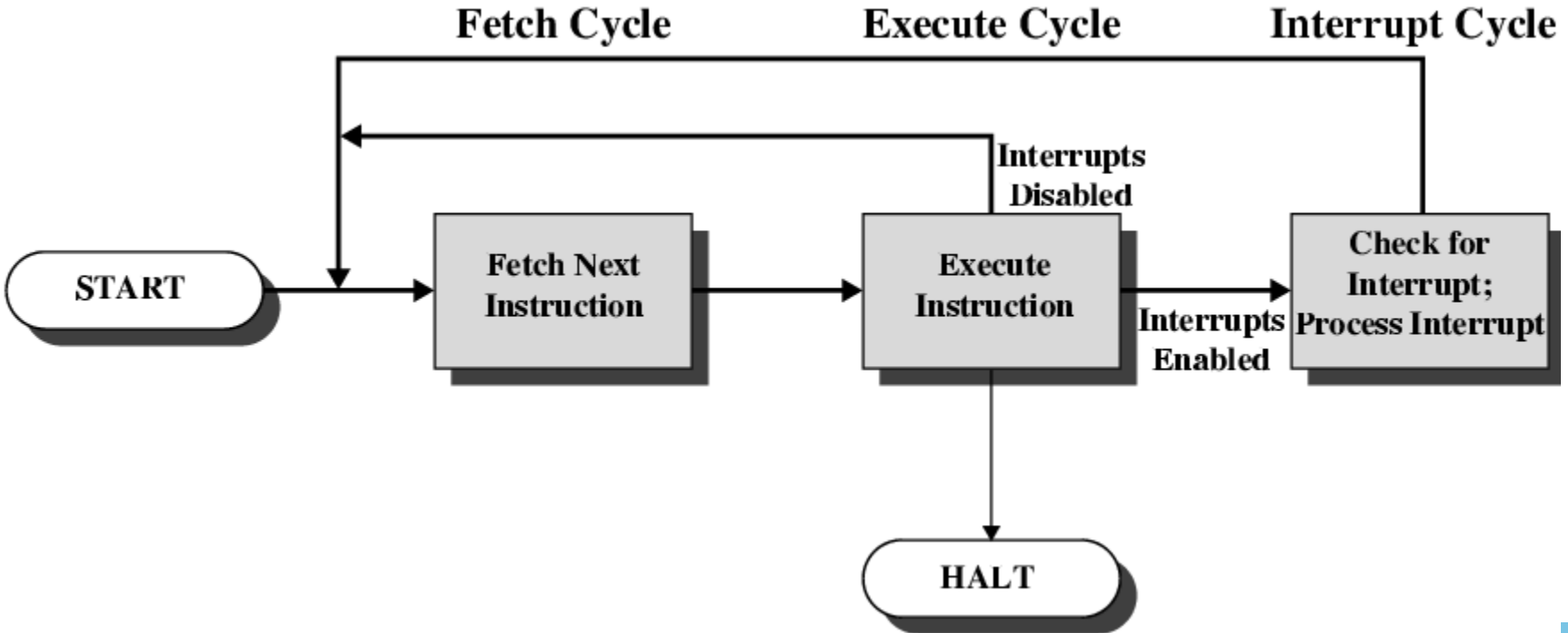
Harvard Architecture

- Uses two memory systems and two separate busses
  - Instruction Memory
  - Data Memory



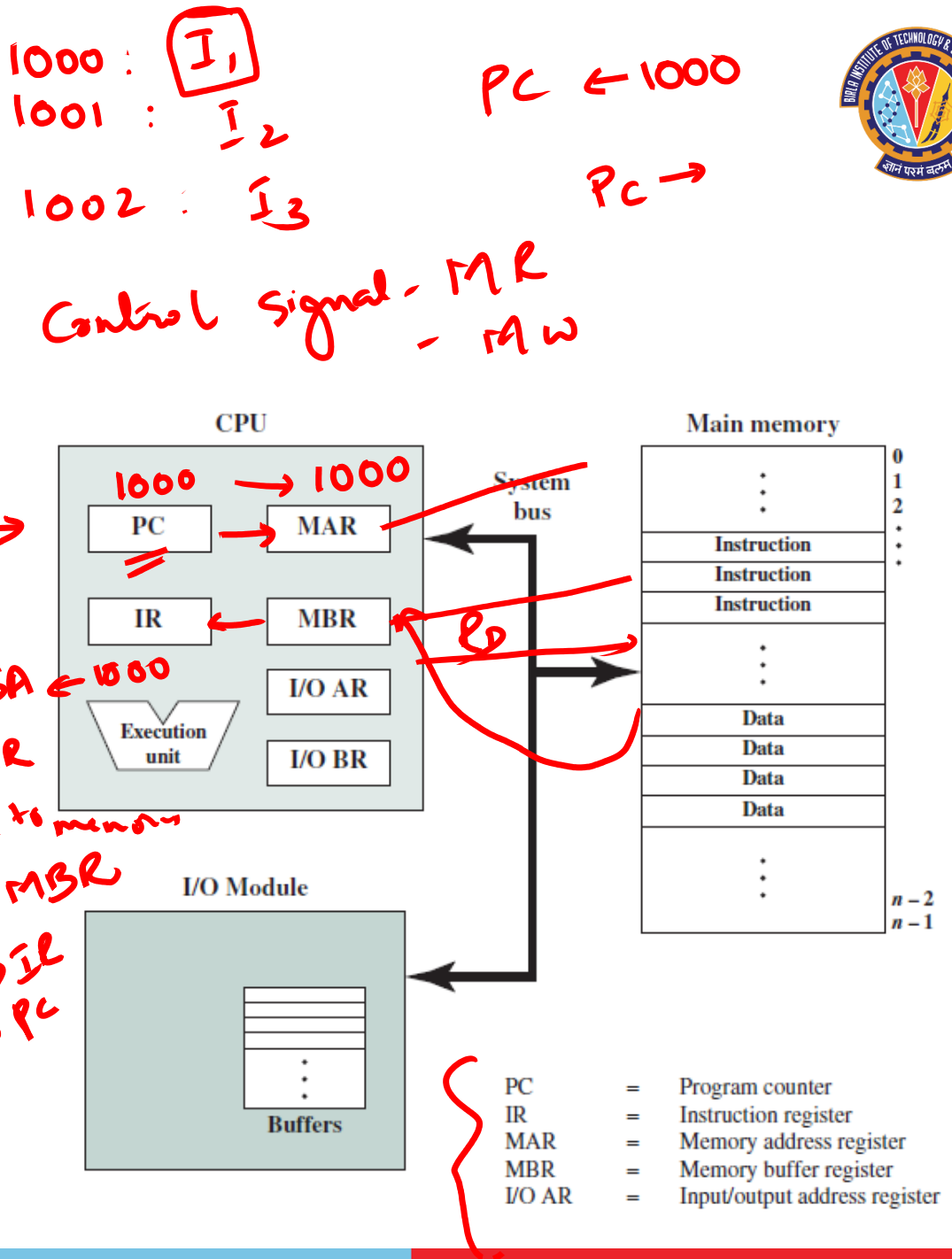
Instruction Cycle Diagram

- Instruction execution : Two steps:
  - Fetch ✓
  - Execute ✓
- Interrupt: Interrupt is checked at the end of Instruction cycle



Fetch Cycle

- Program Counter (PC) holds address of next instruction to be fetched
- Processor fetches instruction from memory location pointed to by PC
- Instruction loaded into Instruction Register (IR)
- Processor interprets instruction and performs required actions during execution cycle
- Increment PC
  - Unless told otherwise



Execute Cycle

ADD

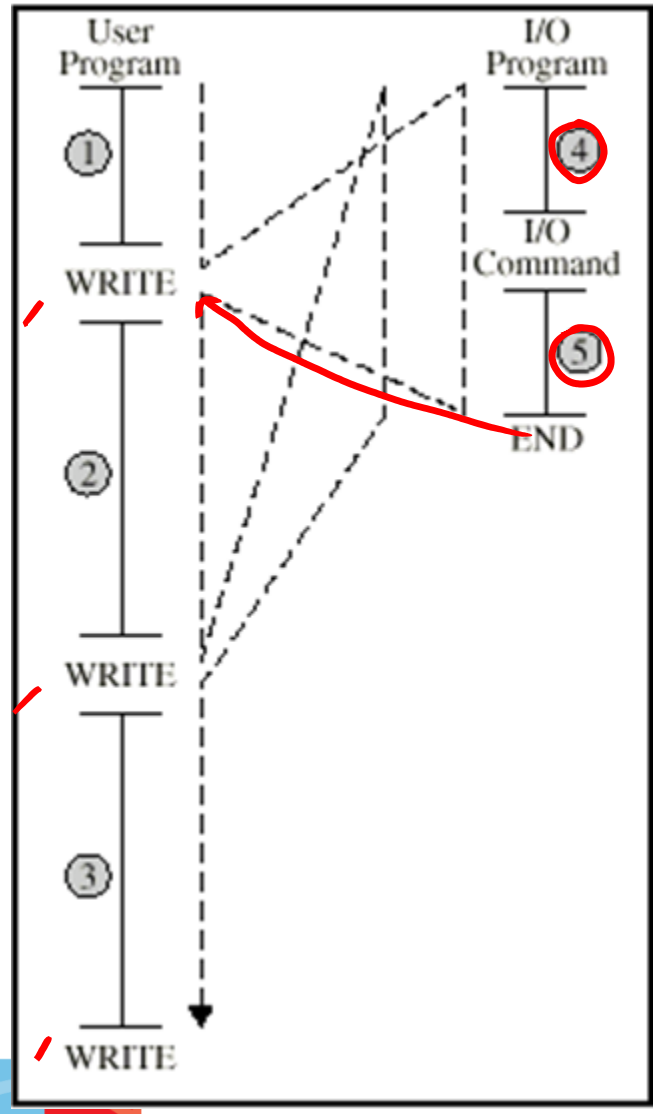
JMP

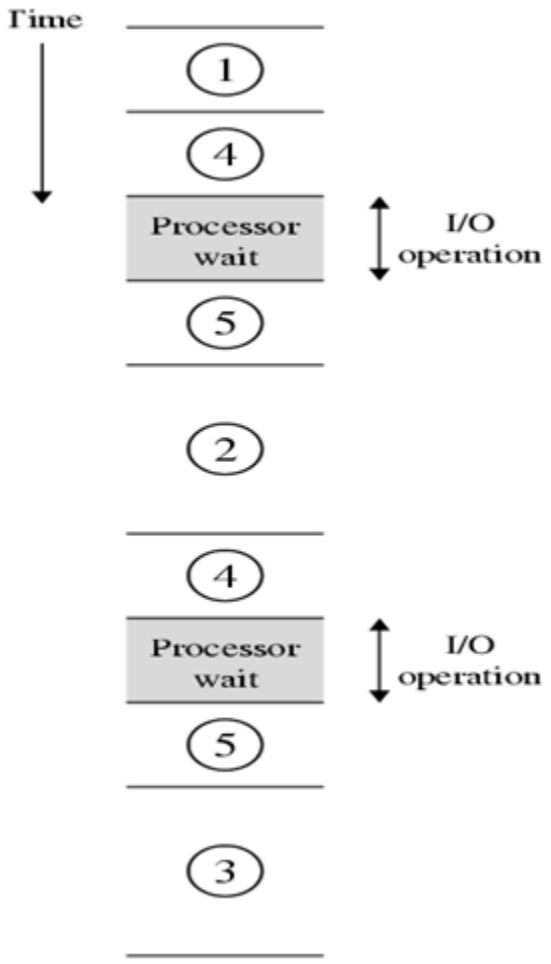
- Processor - memory
  - Data transfer between CPU and main memory
- Processor - I/O
  - Data transfer between CPU and I/O module
- Data processing
  - Some arithmetic or logical operation on data
- Control
  - Alteration of sequence of operations
  - e.g. jump
- Combination of above

Interrupt Cycle

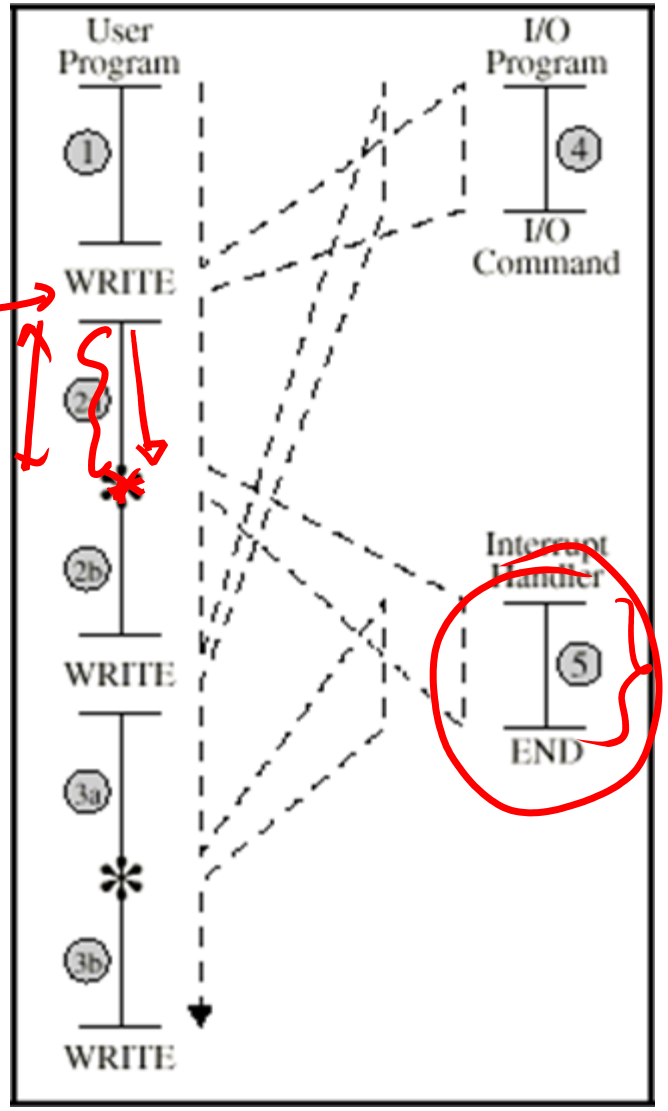
- Interrupts: Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing
- Interrupts enhances processing efficiency

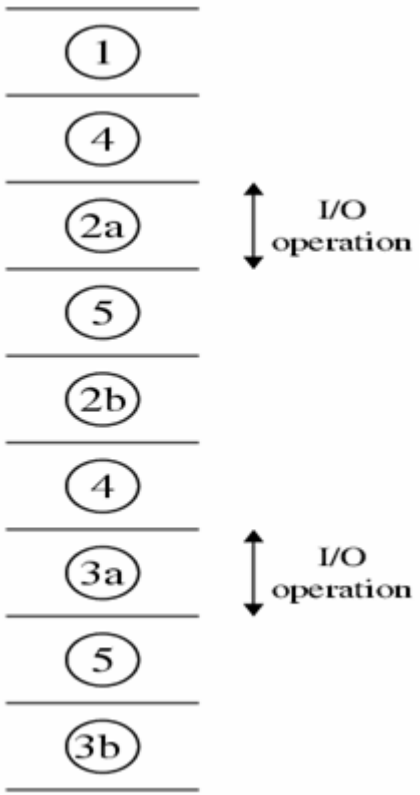
Program Flow Control (No Interrupts)





Program Flow Control (With Interrupts)





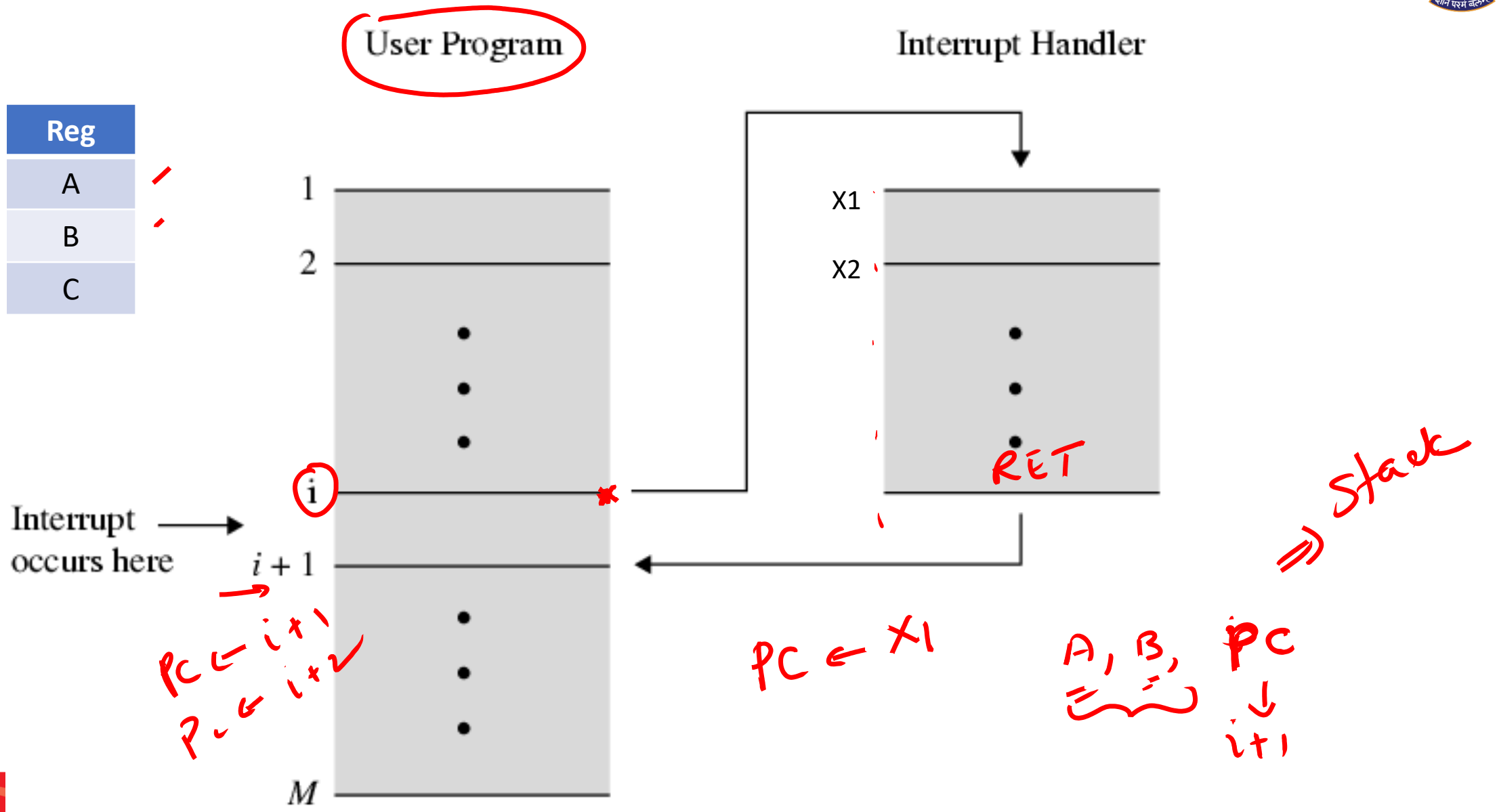
Types of Interrupts

- Types of interrupts:
  - Program
    - e.g. overflow, division by zero
  - Timer
    - Generated by internal processor timer
    - Used in pre-emptive multi-tasking
  - I/O
    - from I/O controller
  - Hardware failure
    - e.g. memory parity error

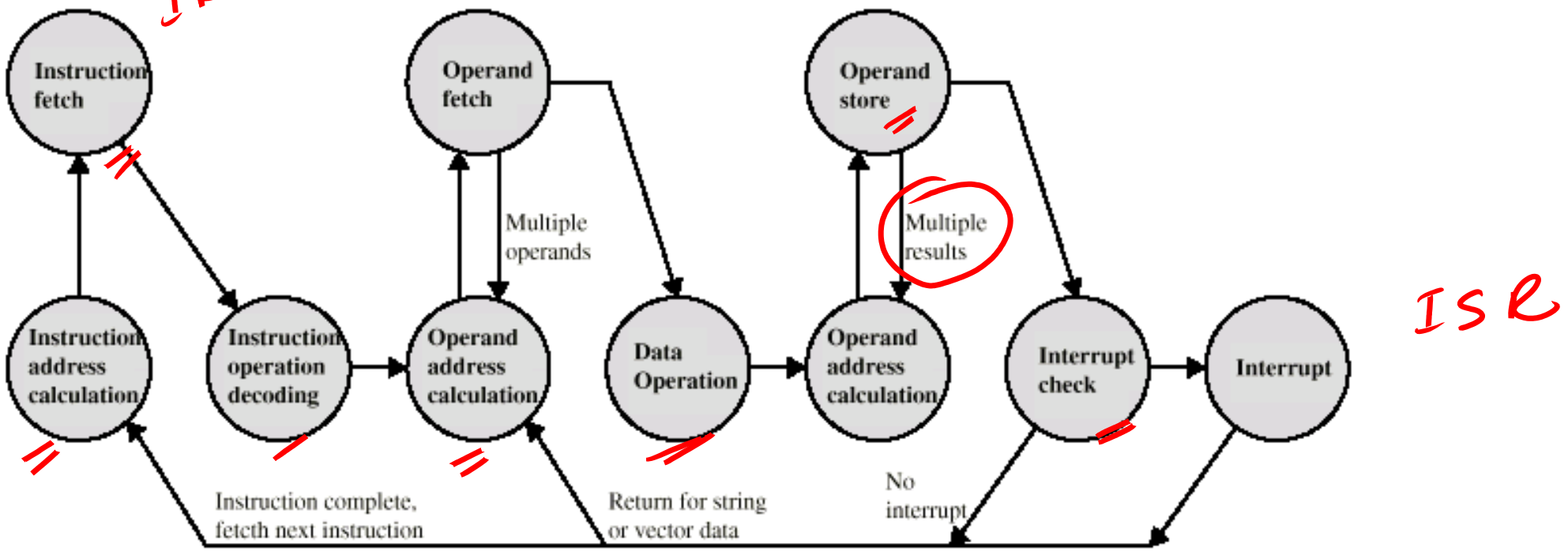
Interrupt Cycle

- Processor checks for interrupt
  - Indicated by an interrupt signal
- If no interrupt, fetch next instruction
- If interrupt pending:
  - Suspend execution of current program
  - Save context
  - Set PC to start address of interrupt handler routine
  - Process interrupt
  - Restore context and continue interrupted program

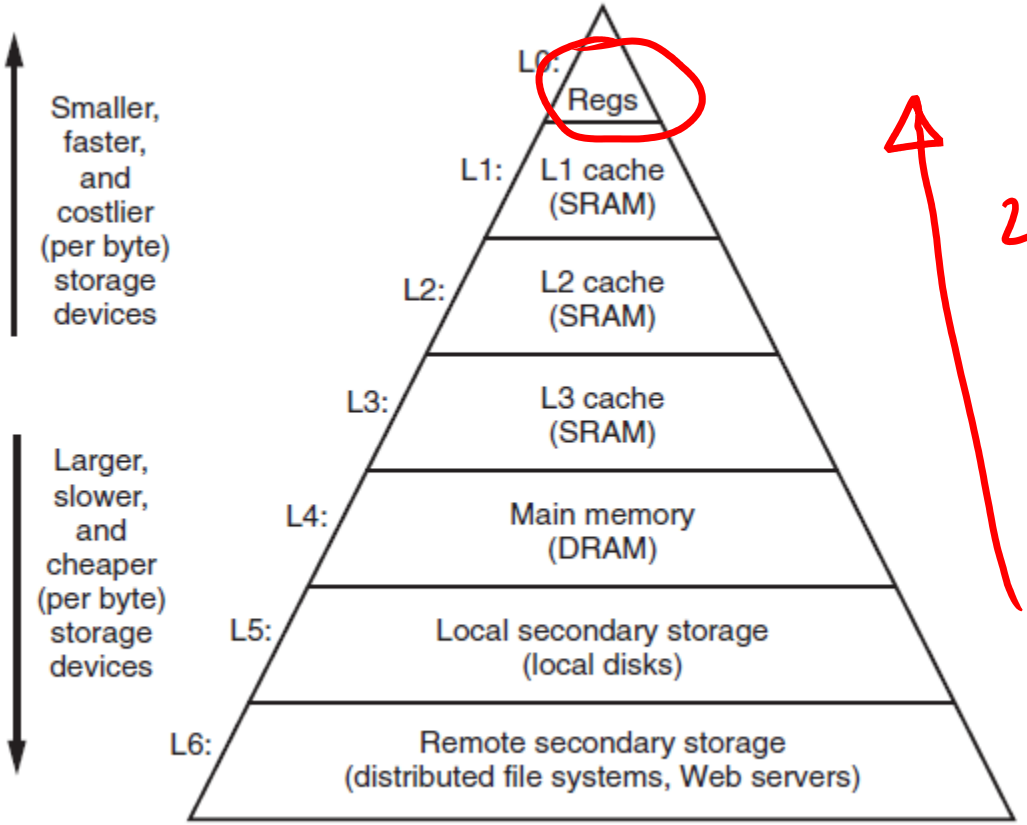
Transfer of Control via Interrupts



Instruction Cycle - State Diagram



Memory Hierarchy



An example of a memory hierarchy.

1000

PC → hold address of the instruction that is to be executed

MAR → memory address register address of the memory

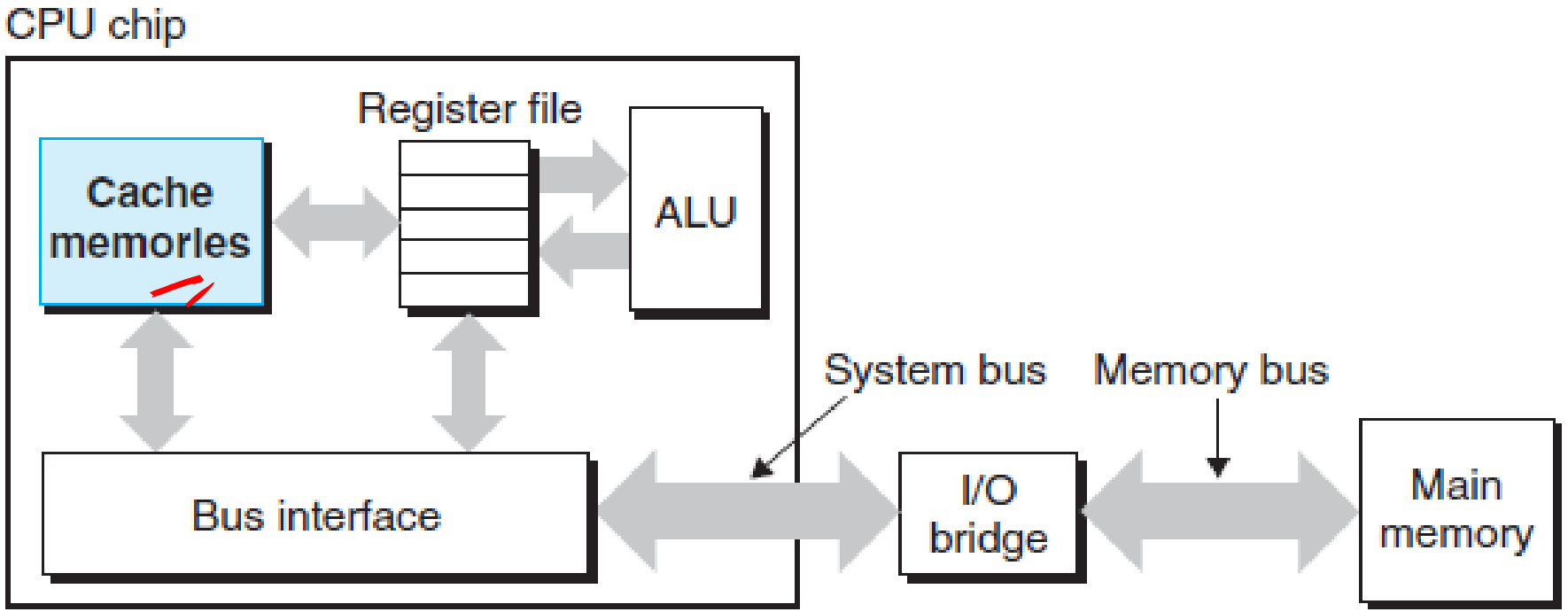
2000

- CISC vs RISC

CISC: fewer

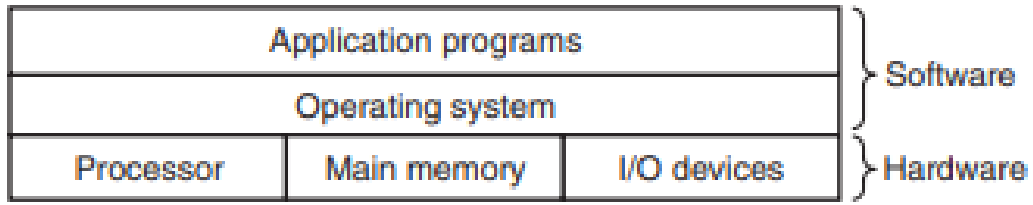
RISC: huge no registers ⇓ register file

Role of Cache Memory

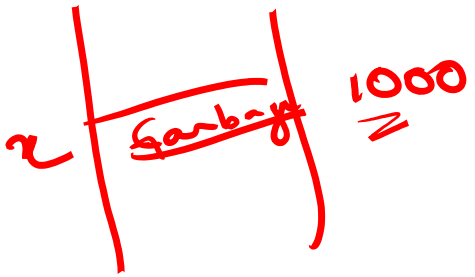


Operating System

- collection of software/ Program that acts as an intermediary between an user of a computer and the computer hardware.
- is a program that helps to run all the other programs
- Three main functions:
  - Resource management
  - Establish an user interface
  - Execute and provide services for application software



Layered view of a computer system.




```
main()
{ int x;
  scanf("%d",&x);
  printf("x = %d",x);
}
```

Main objectives

- Convenience
- Efficiency
- Ability to evolve and offer new services
- Maximize System performance
- Protection and access control
- Footprint of OS should be small

Important Note

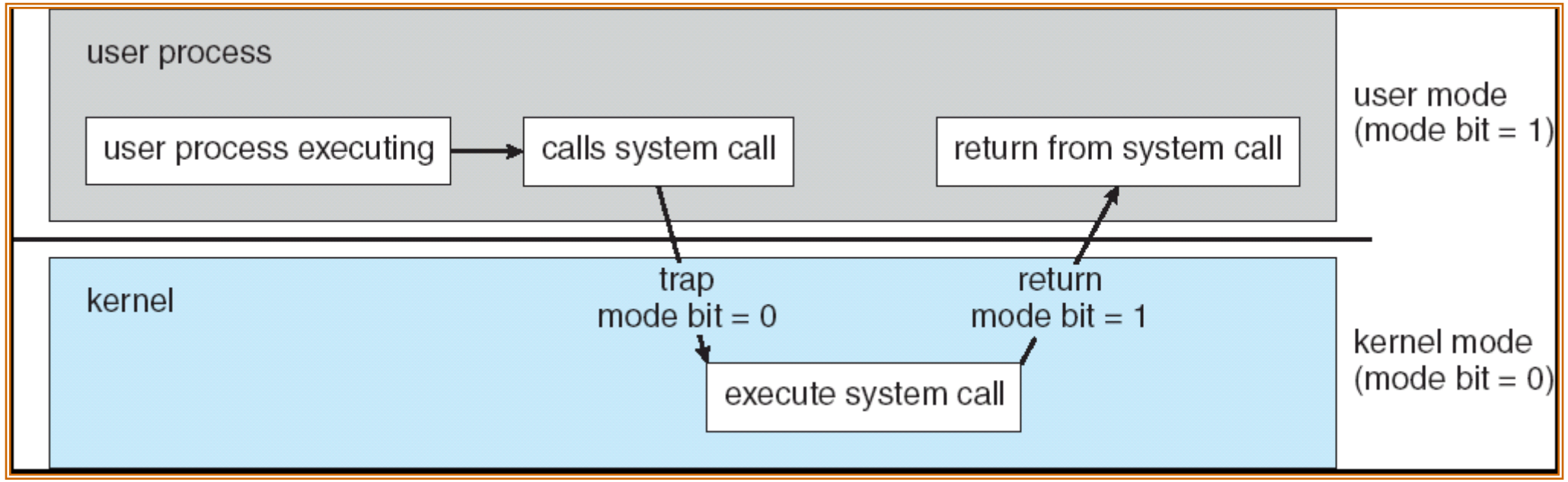
- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

- "The one program running at all times on the computer" is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program

Operating System Operations

- **Dual-mode** operation
  - User mode
  - Kernel mode (also known as System Mode / Supervisor mode/ privileged mode)
- User mode(1):
  - user program executes in user mode
  - certain areas of memory are protected from user access
  - certain privileged instructions may not be executed
- Kernel Mode (0)
  - privileged instructions may be executed
  - protected areas of memory may be accessed

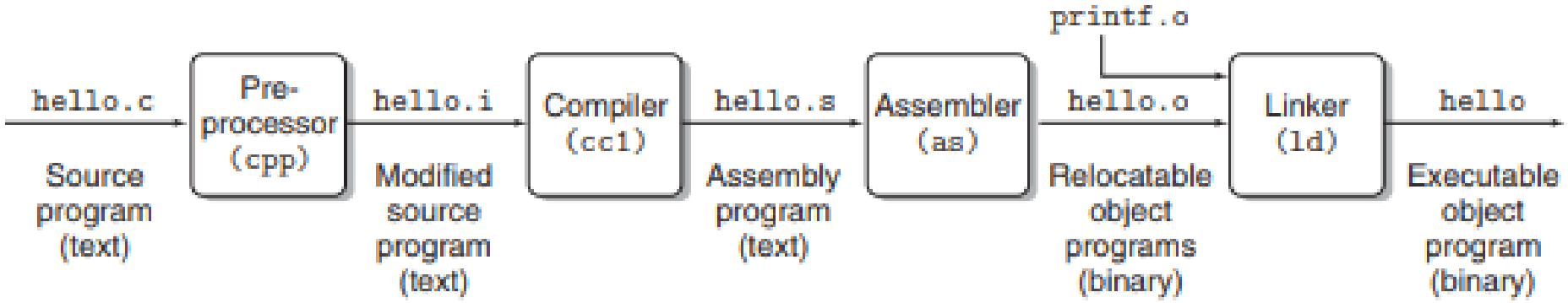
Transition from user to kernel mode



Running a Hello.c Program

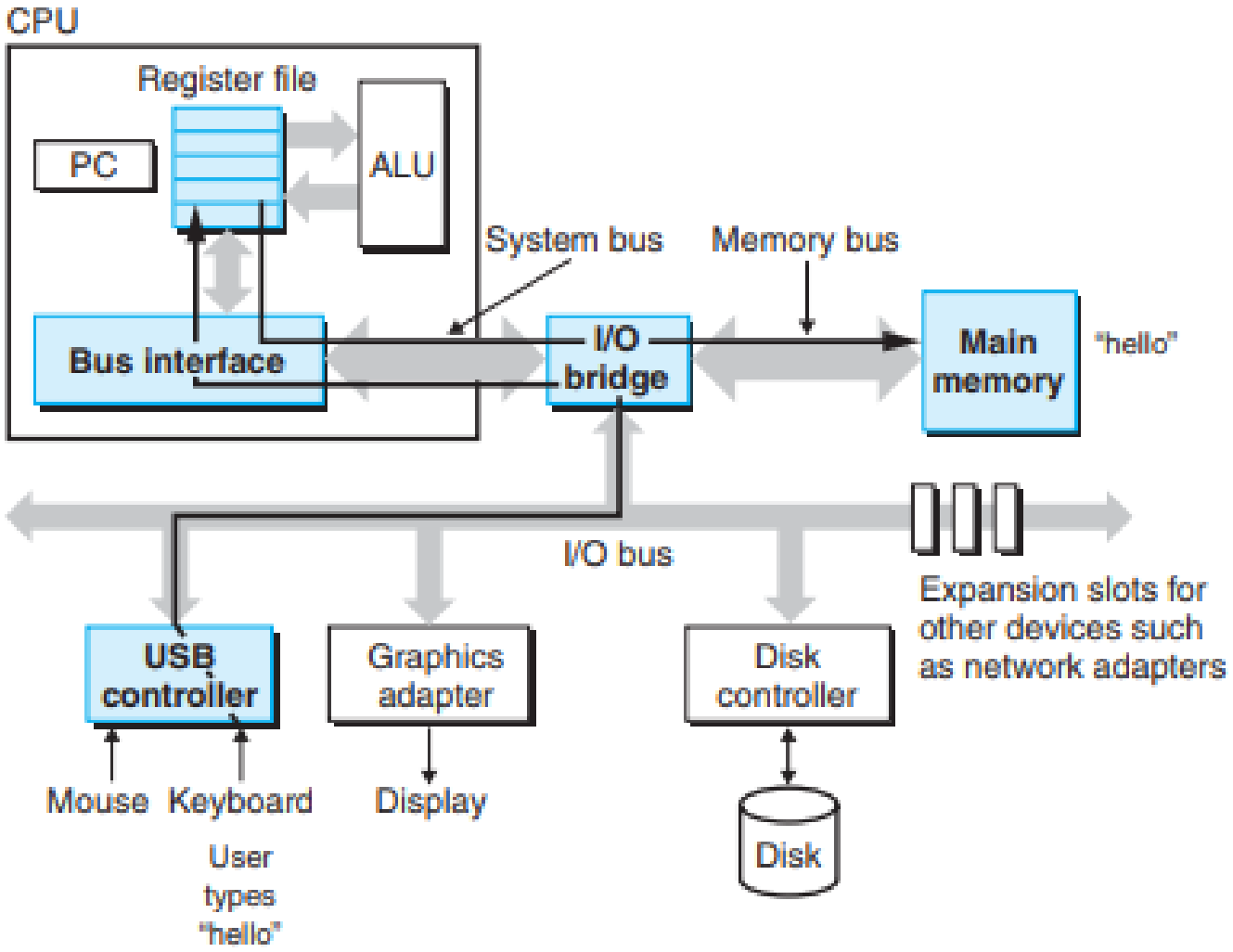
```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
}
```

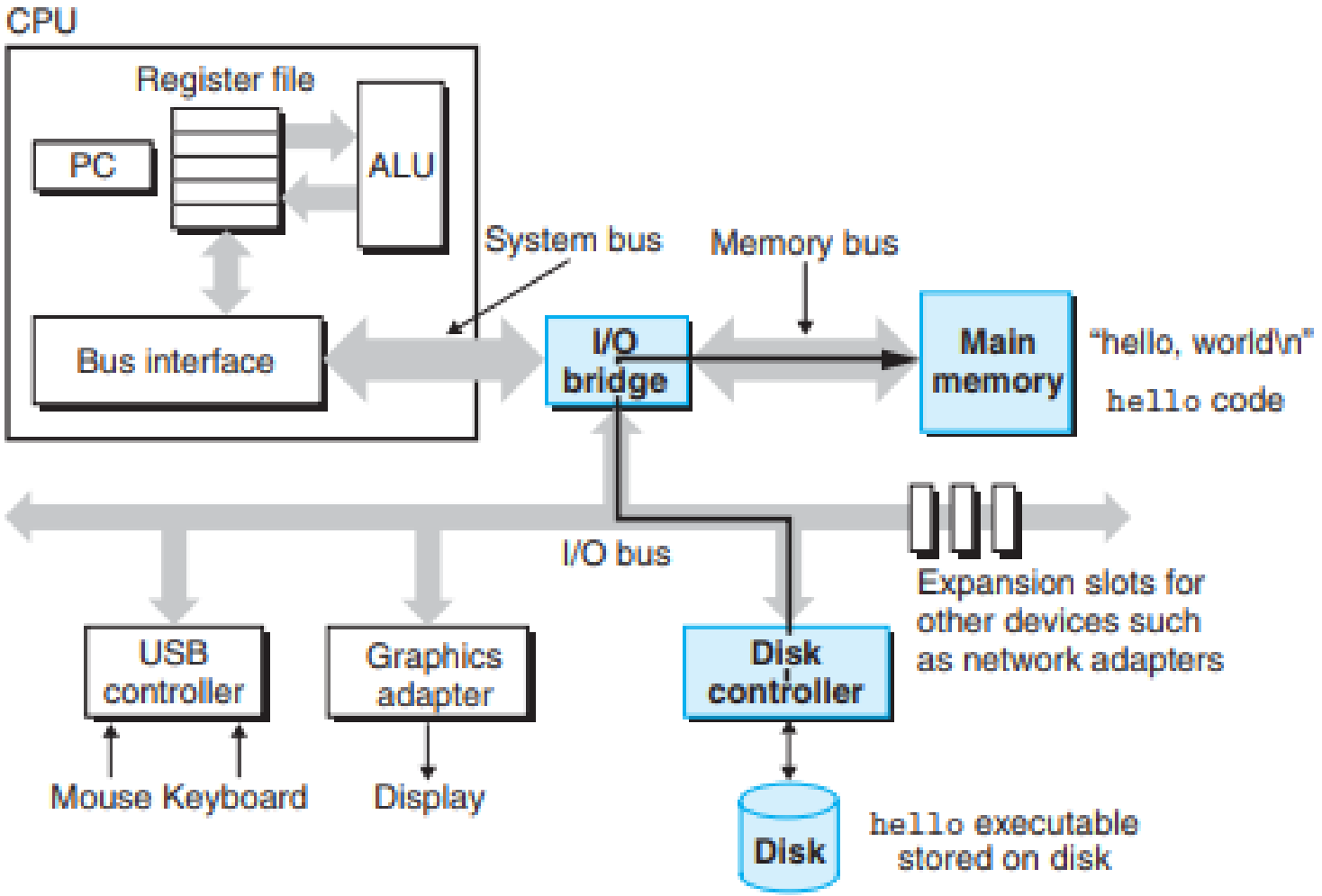


The compilation system.

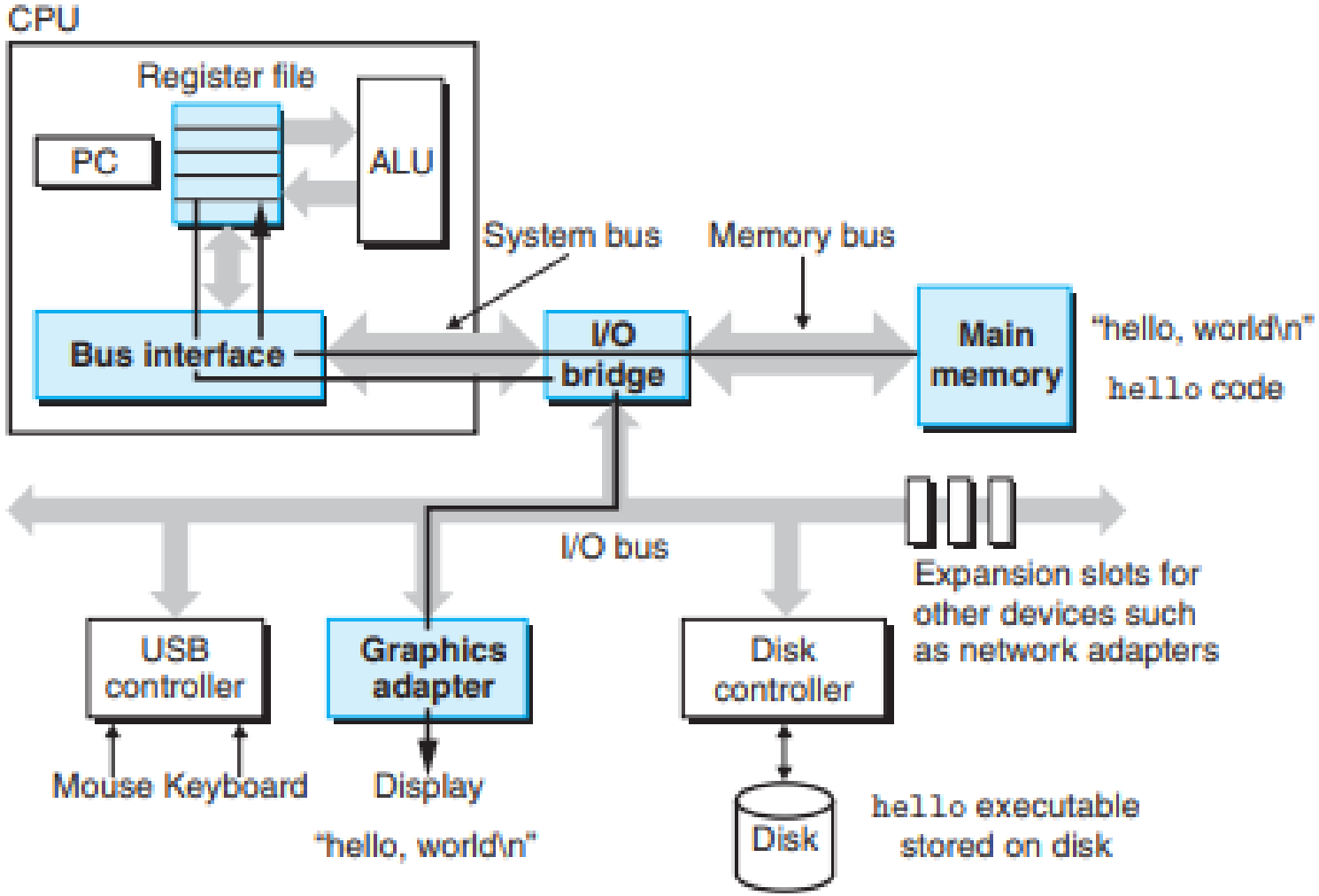
Reading ./hello command from Keyboard



Loading the executable from disk into main memory



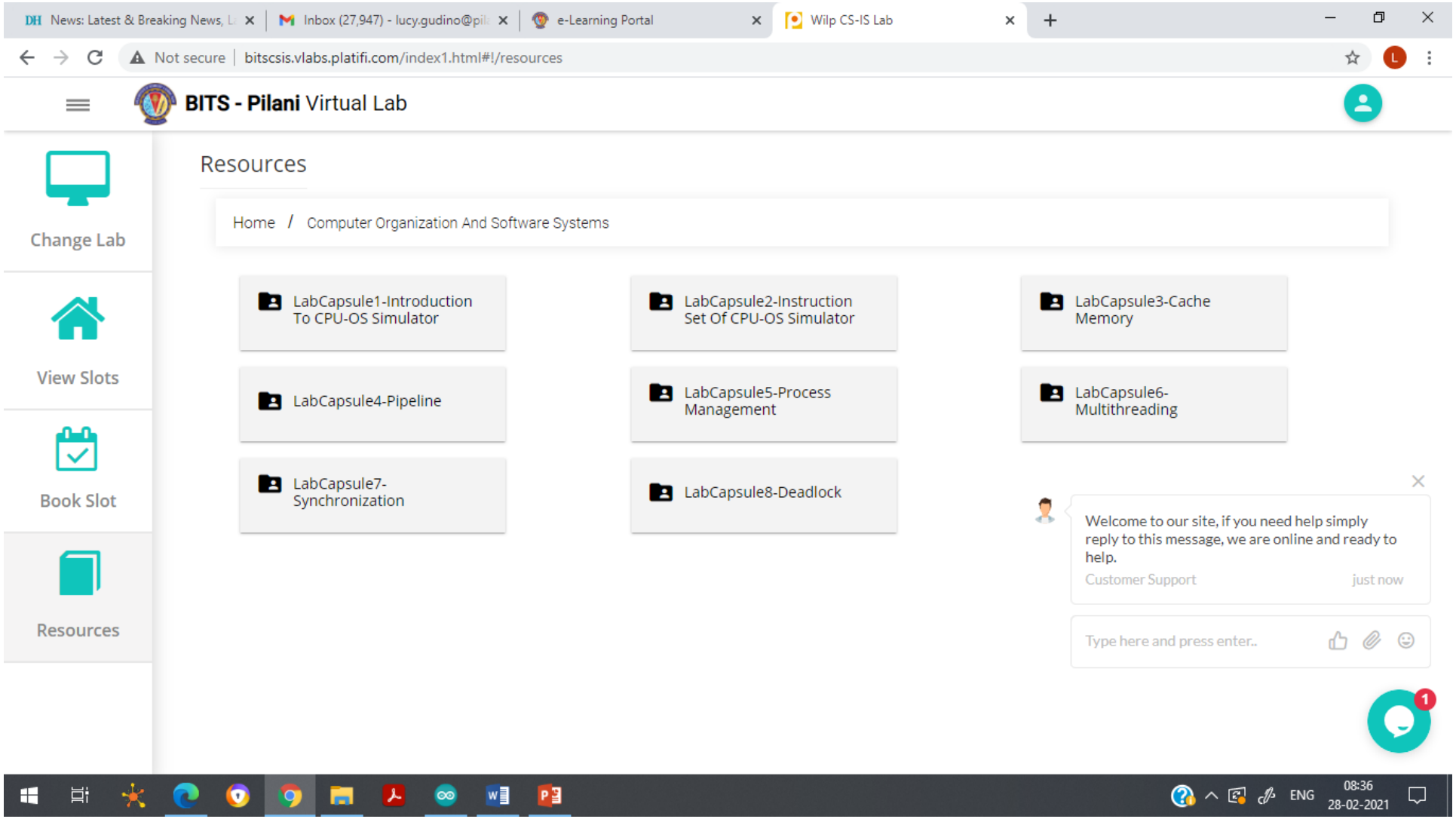
Writing the output string from memory to the display



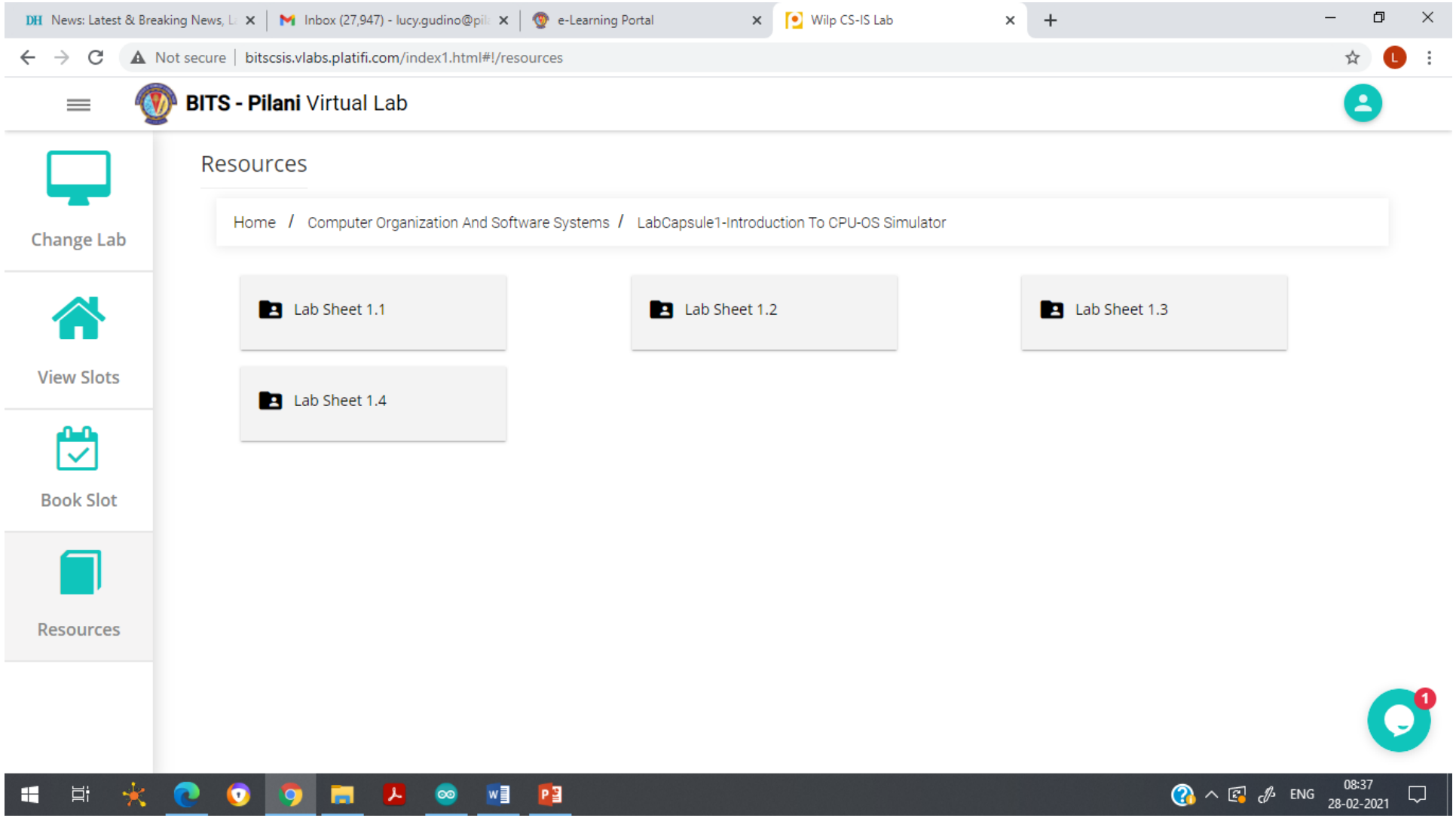
Why do we need to know how compilation works?

- Optimizing program performance.
- Understanding link-time errors
- Avoiding security holes.

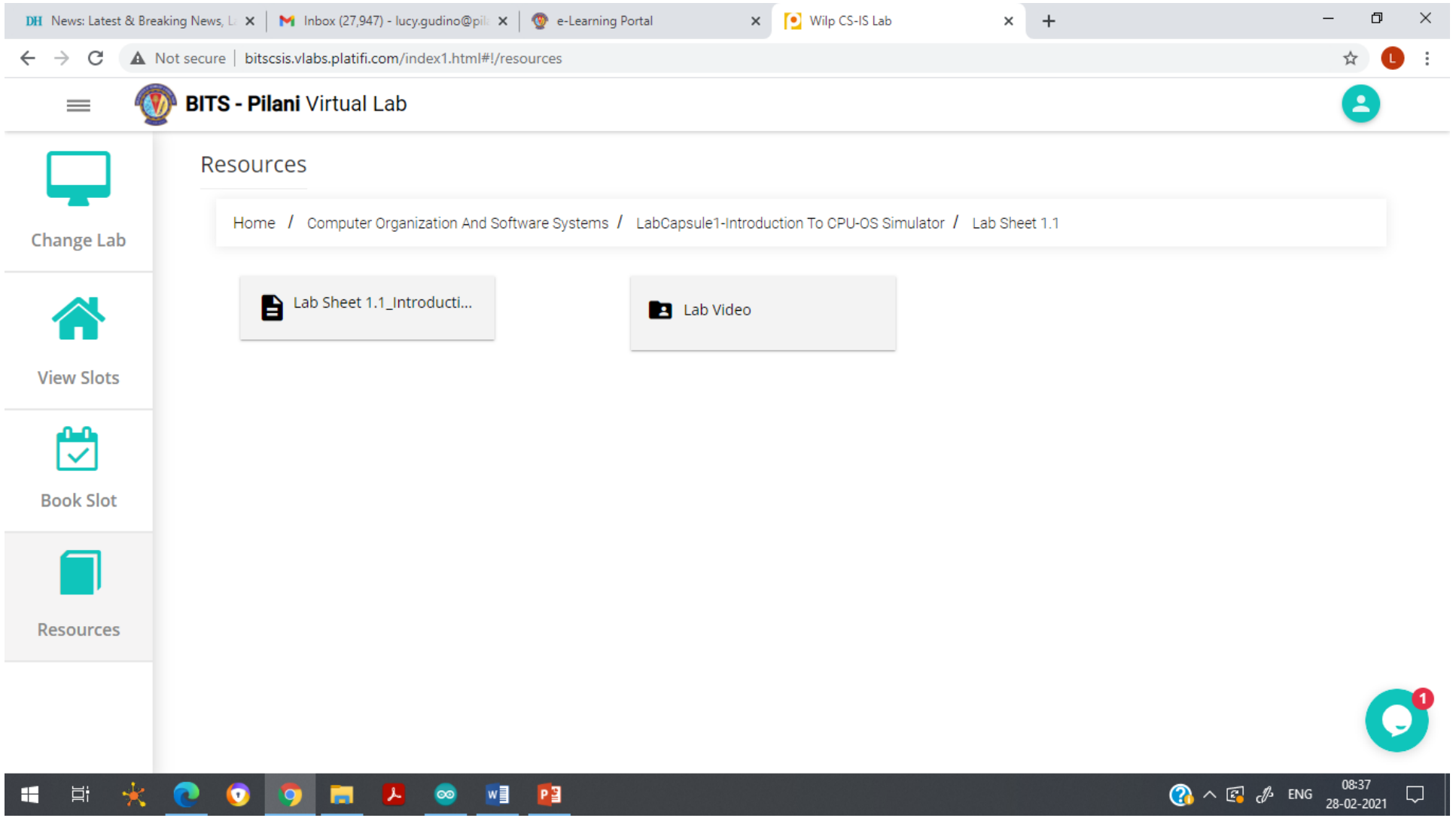
Lab Activity



Contd…


News: Latest & Breaking News,
Inbox (27,947) - lucy.gudino@pil
e-Learning Portal
Wilp CS-IS Lab
Not secure
bitscsis.vlabs.platifi.com/index1.html#!/resources
BITS - Pilani Virtual Lab
Change Lab
View Slots
Book Slot
Resources
Resources
Home / Computer Organization And Software Systems / LabCapsule1-Introduction To CPU-OS Simulator
Lab Sheet 1.1
Lab Sheet 1.2
Lab Sheet 1.3
Lab Sheet 1.4
ENG
08:37
28-02-2021

Contd…


News: Latest & Breaking News,
Inbox (27,947) - lucy.gudino@pil
e-Learning Portal
Wilp CS-IS Lab
Not secure
bitscsis.vlabs.platifi.com/index1.html#!/resources
BITS - Pilani Virtual Lab
Change Lab
View Slots
Book Slot
Resources
Resources
Home / Computer Organization And Software Systems / LabCapsule1-Introduction To CPU-OS Simulator / Lab Sheet 1.1
Lab Sheet 1.1_Introducti...
Lab Video
ENG
08:37
28-02-2021

Thank you!