Computer Organization and Software Systems

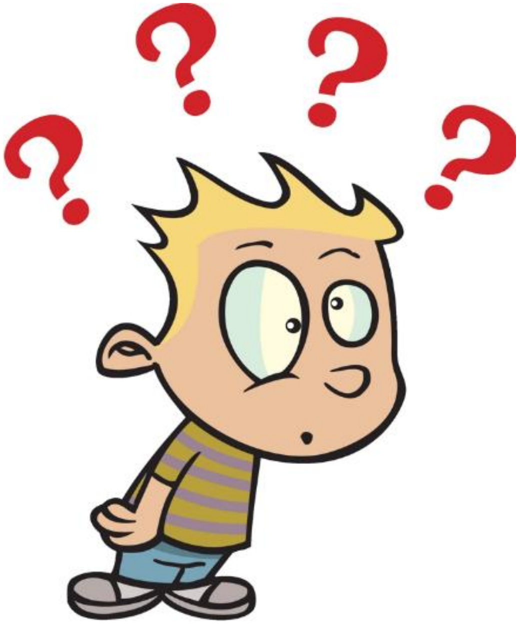
CONTACT SESSION 1

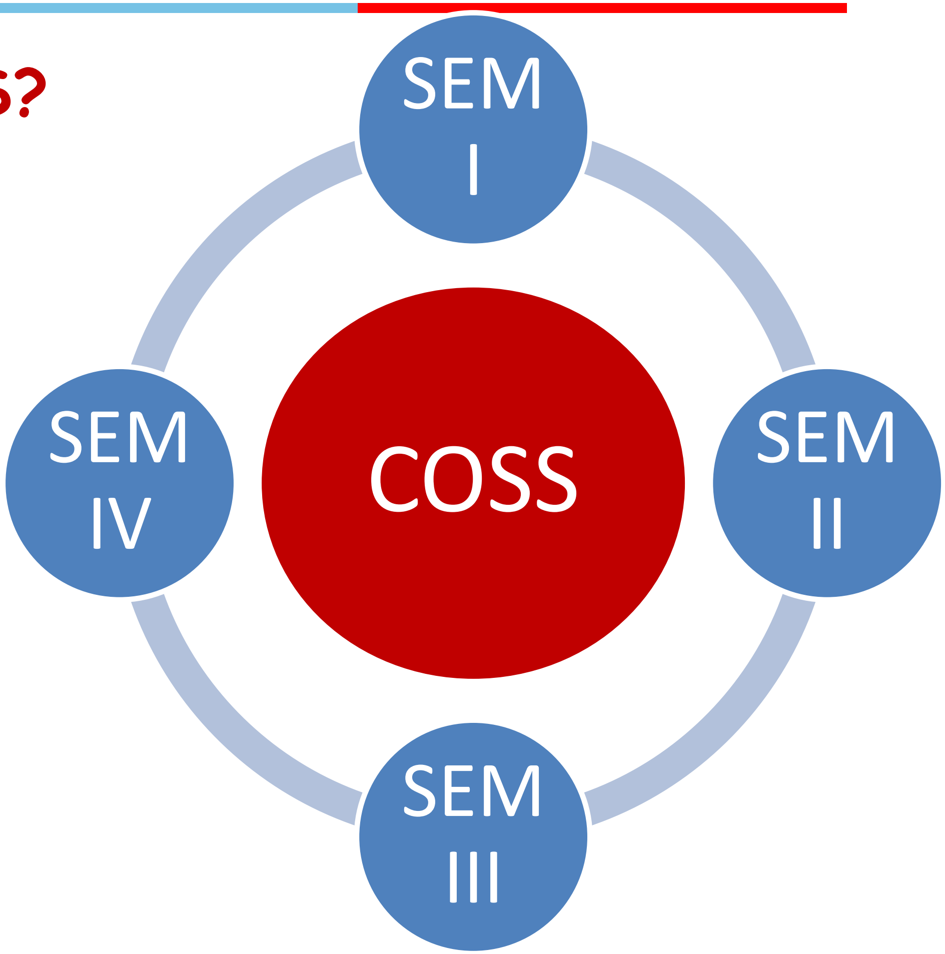
Prof. C R Sarma
WILP .BITS-PILANI

BITS Pilani
Pilani Campus

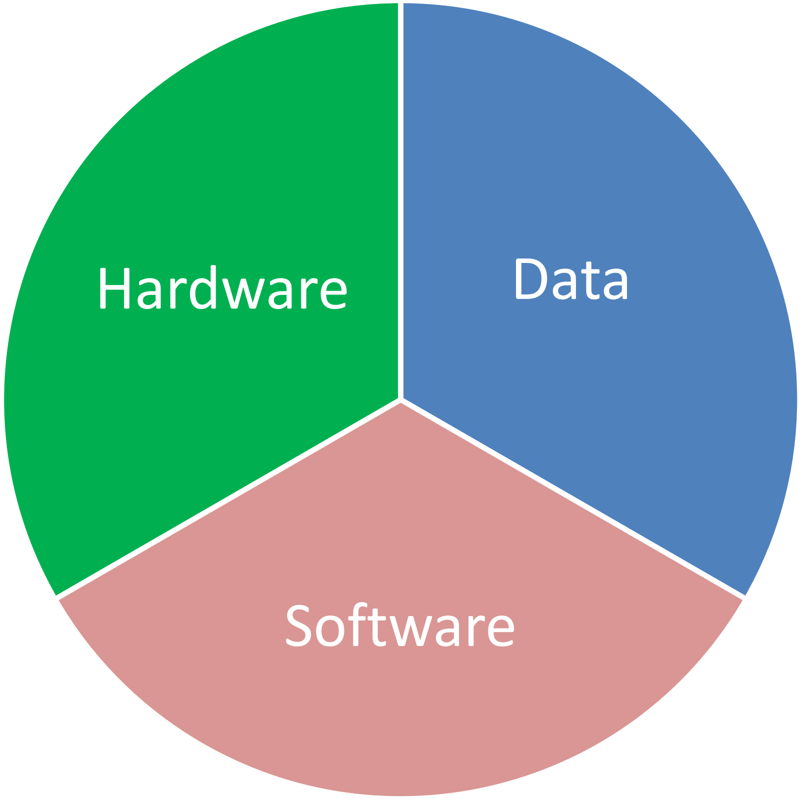
Introduction

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**.

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

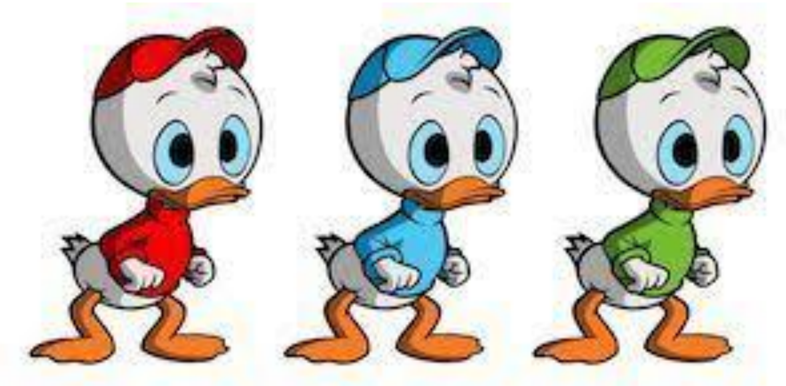
5 unit course.

Sl No.	Evaluation Component	Duration	Weightage %	Nature of Component
1	Mid Sem Exam	90 min	30%	CB
2	Comprehensive Examination	180 min	40%	OB
3	Quiz	----	5%	OB
4	Assignments	---	25%	OB

Assignments

- Two assignments:
  - One pre-midsem exam : 15%
  - One post-midsem : 10%
- Lab based
- Simulator to be used : CPU-OS simulator
  - Open source tool (http://www.teach-sim.com/)
  - Virtual lab (Platify)

Assignment should not be



Contact Details

Email : crsarma@wilp.pilani.bits-pilani.ac.in

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 if not done during.
4. Unanswered questions will be put up on the canvas forum-in discussions page of oursection(II)

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 • Running a Hello Program • Instruction Cycle State Diagram • Operating System role in Managing Hardware • Processes • Threads • Virtual Memory • Files.	T1

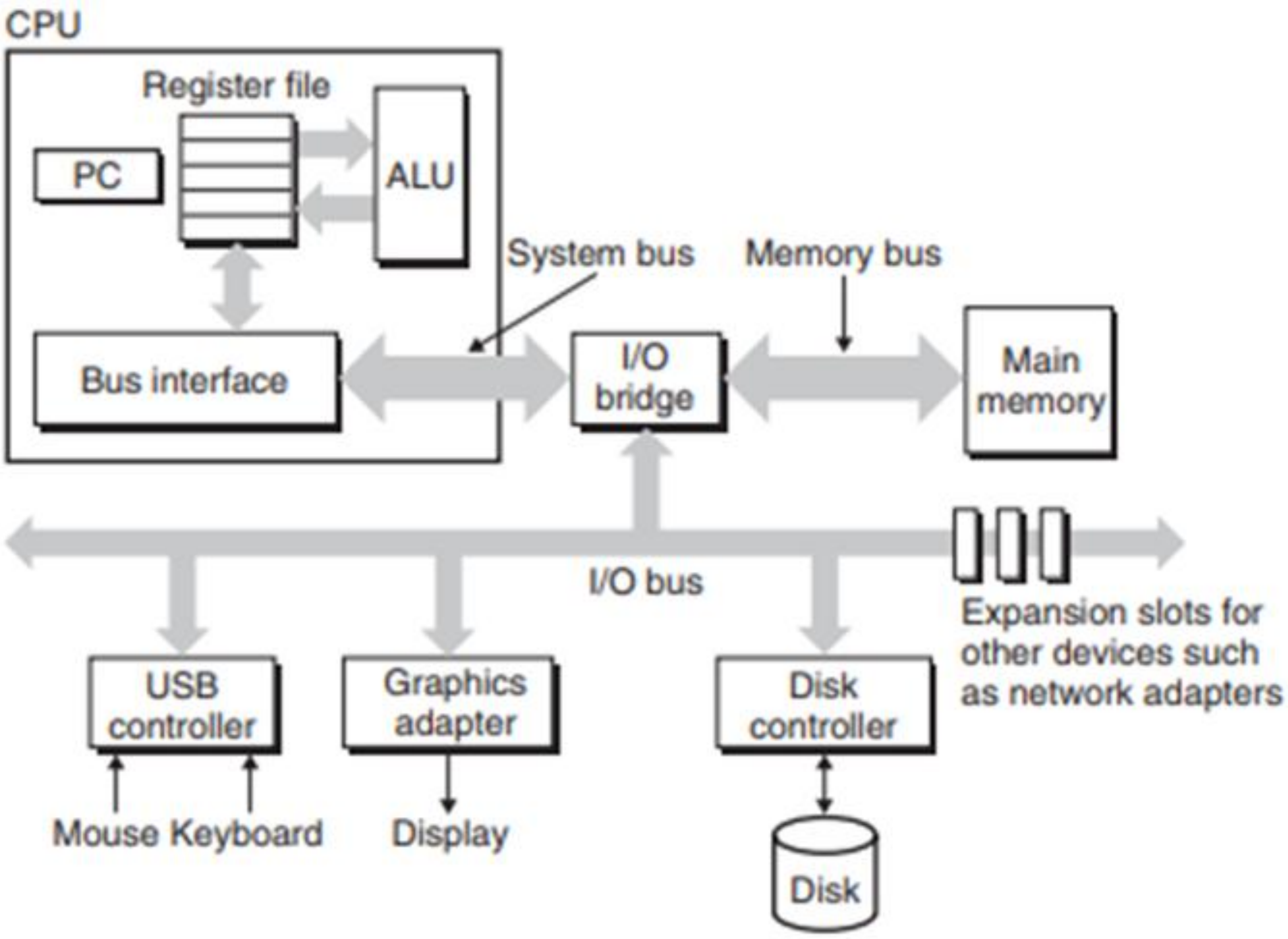
Definition of a Computer

- 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

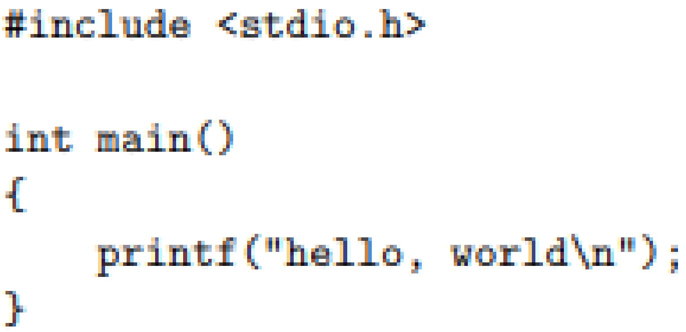
Computer System

- 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

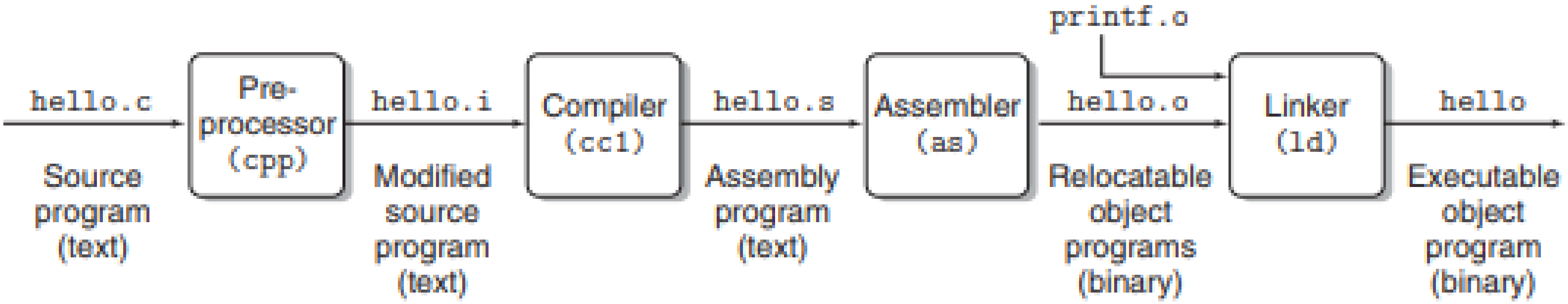


Running a Hello.c Program



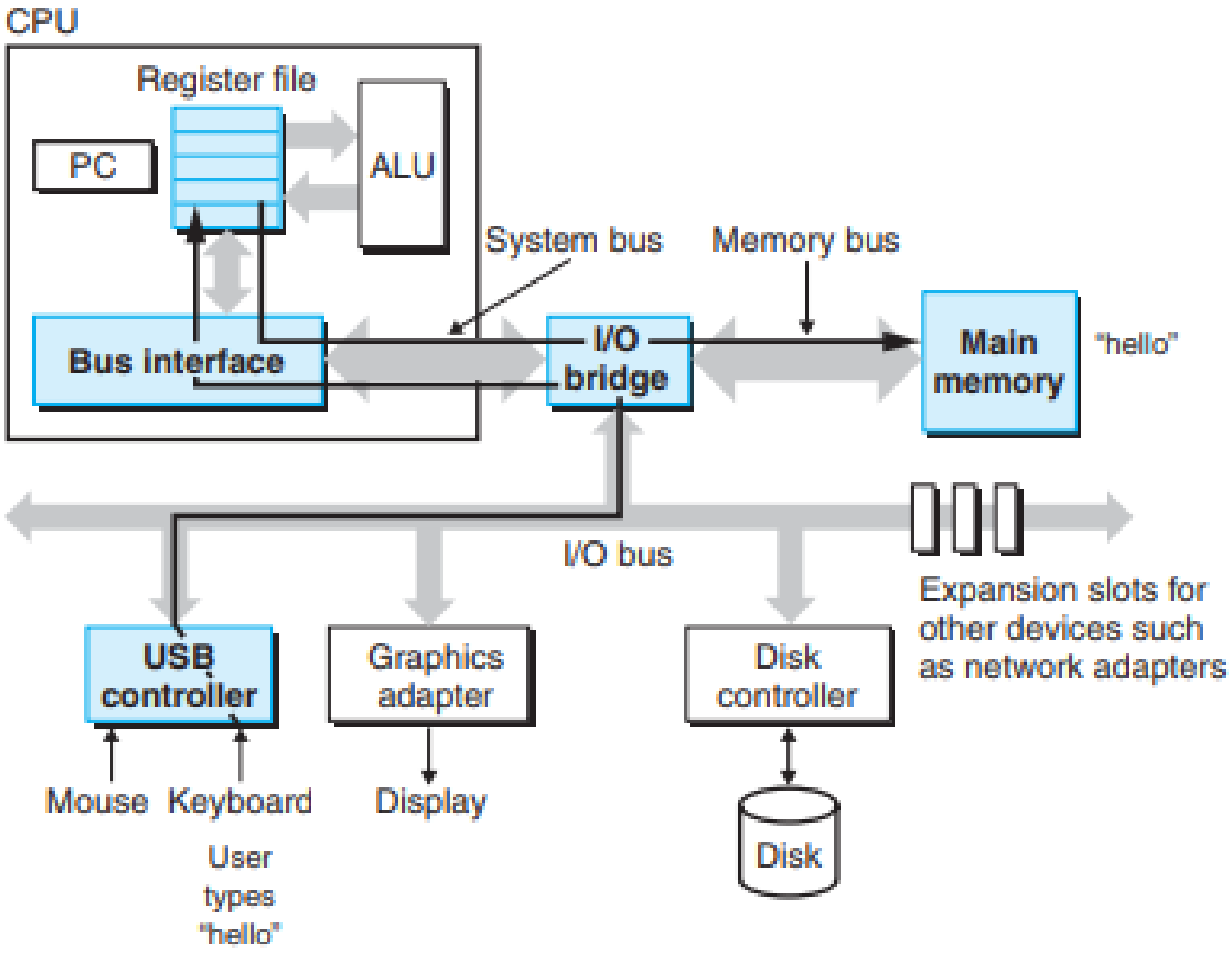
```
#include <stdio.h>

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

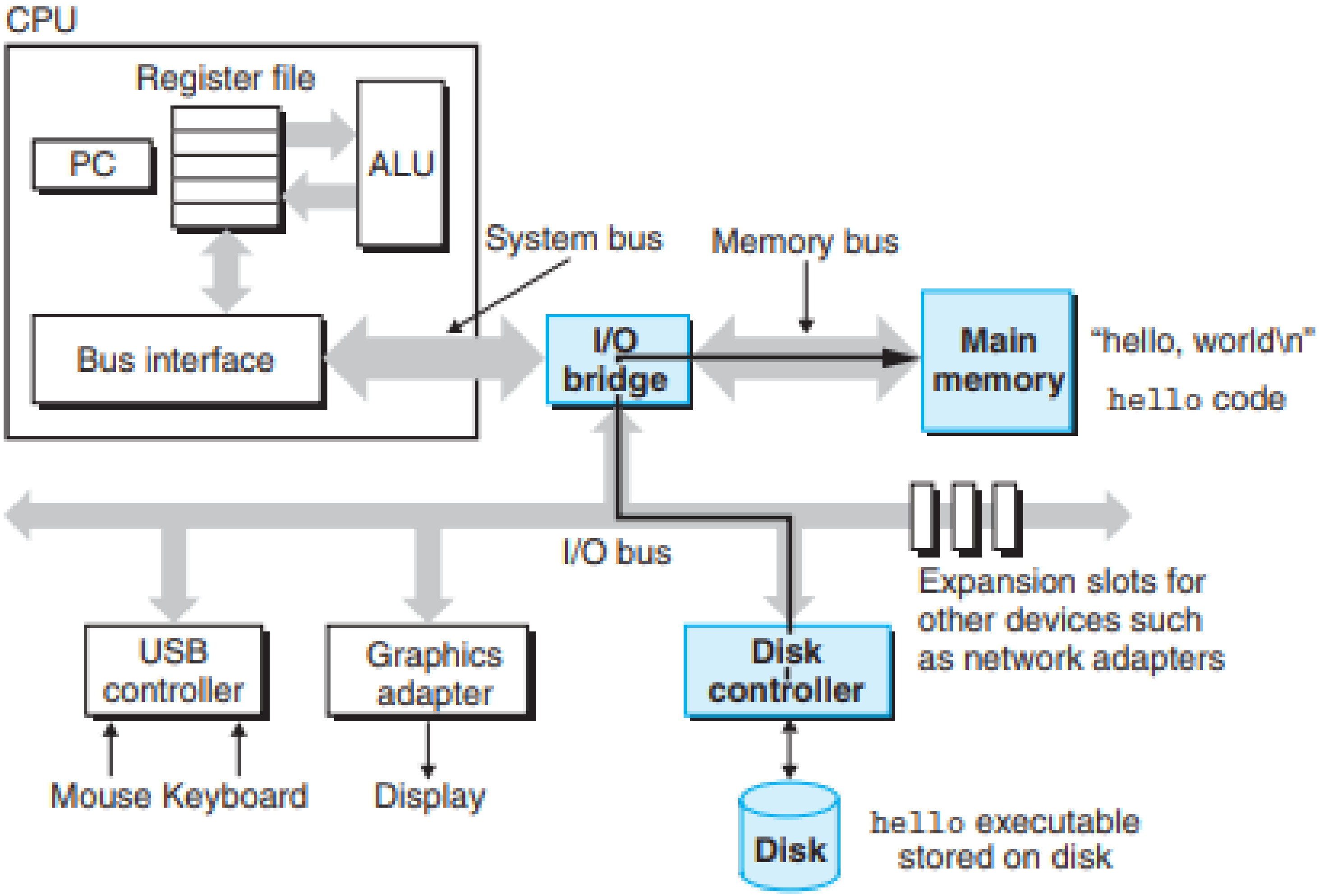


The compilation system.

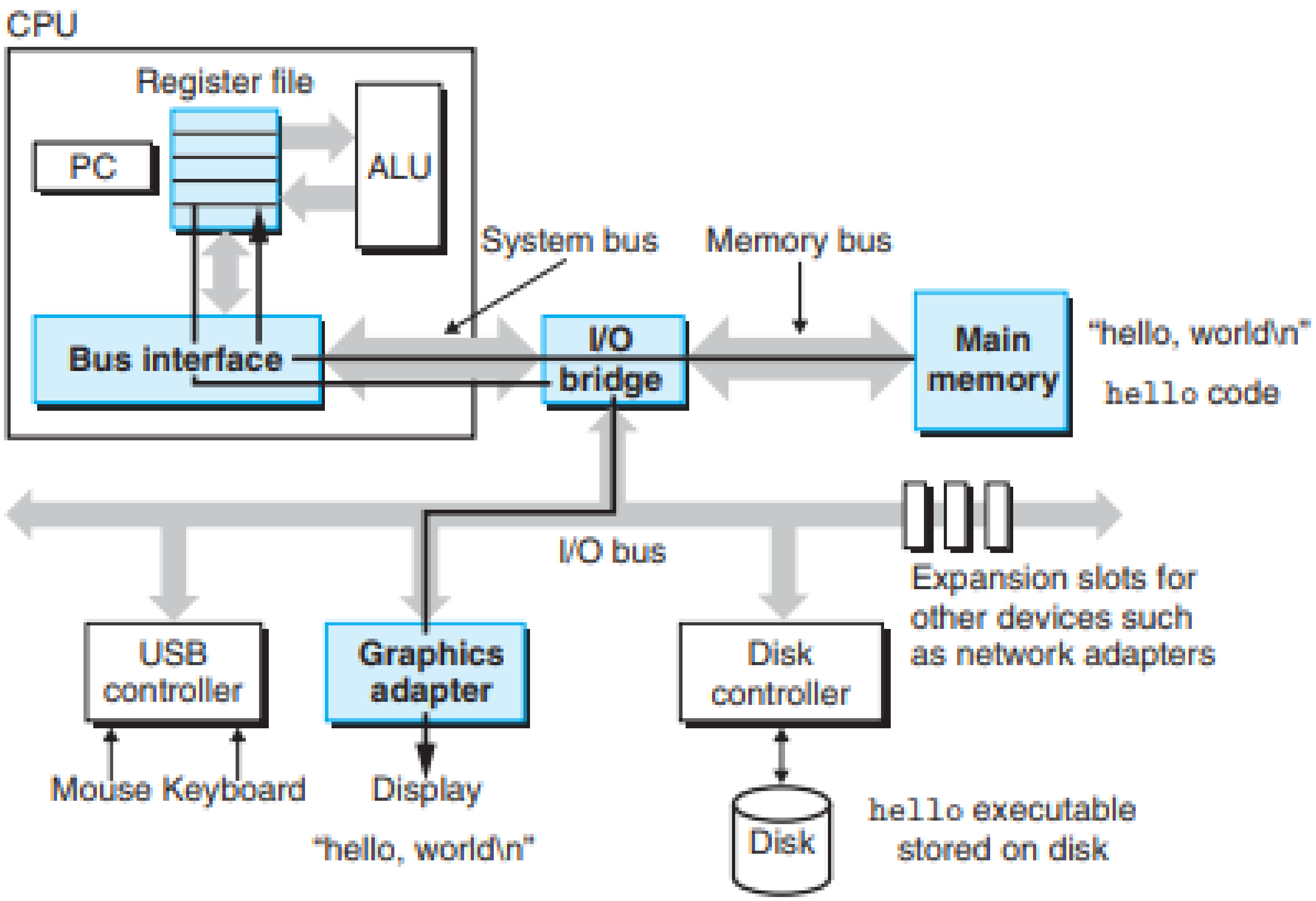
Reading ./hello command from Keyboard



Loading the executable from disk into main memory



Writing the output string from memory to the display

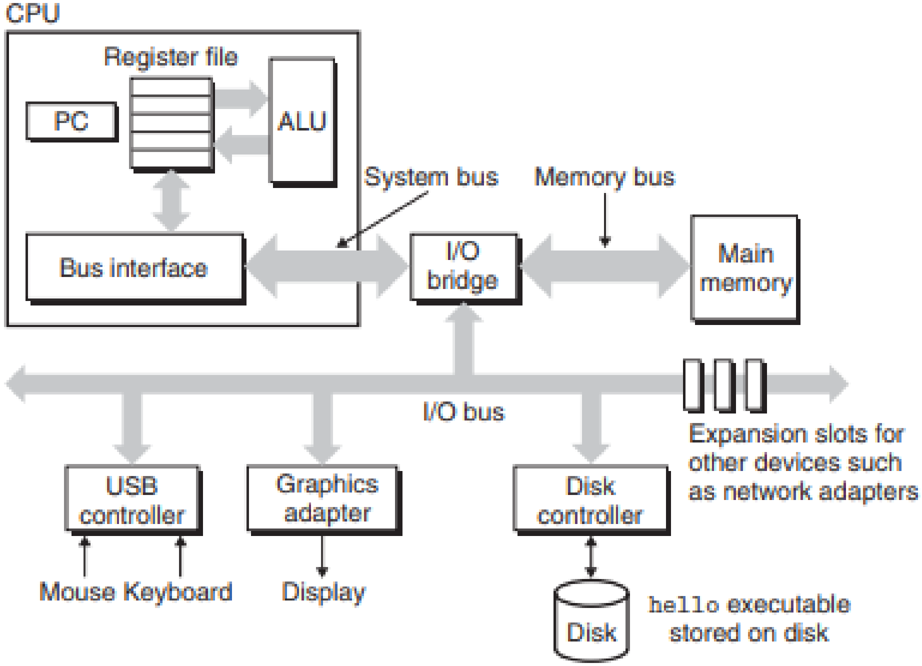


Why do we need to know how compilation works?

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

Von Neumann Architecture

- 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

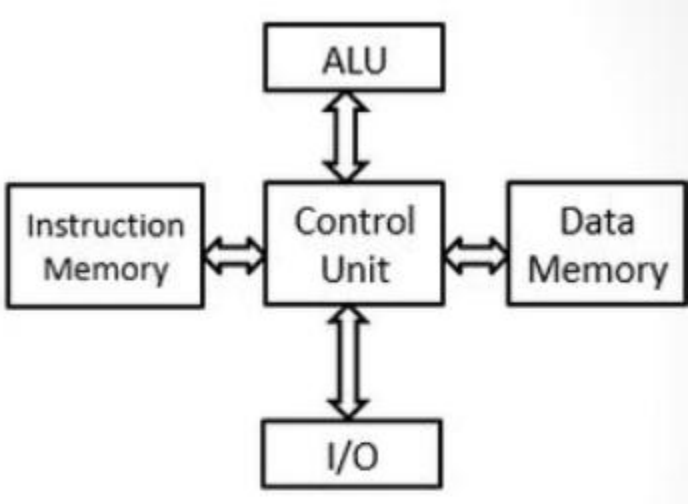


Von Neumann Architecture…

- 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)

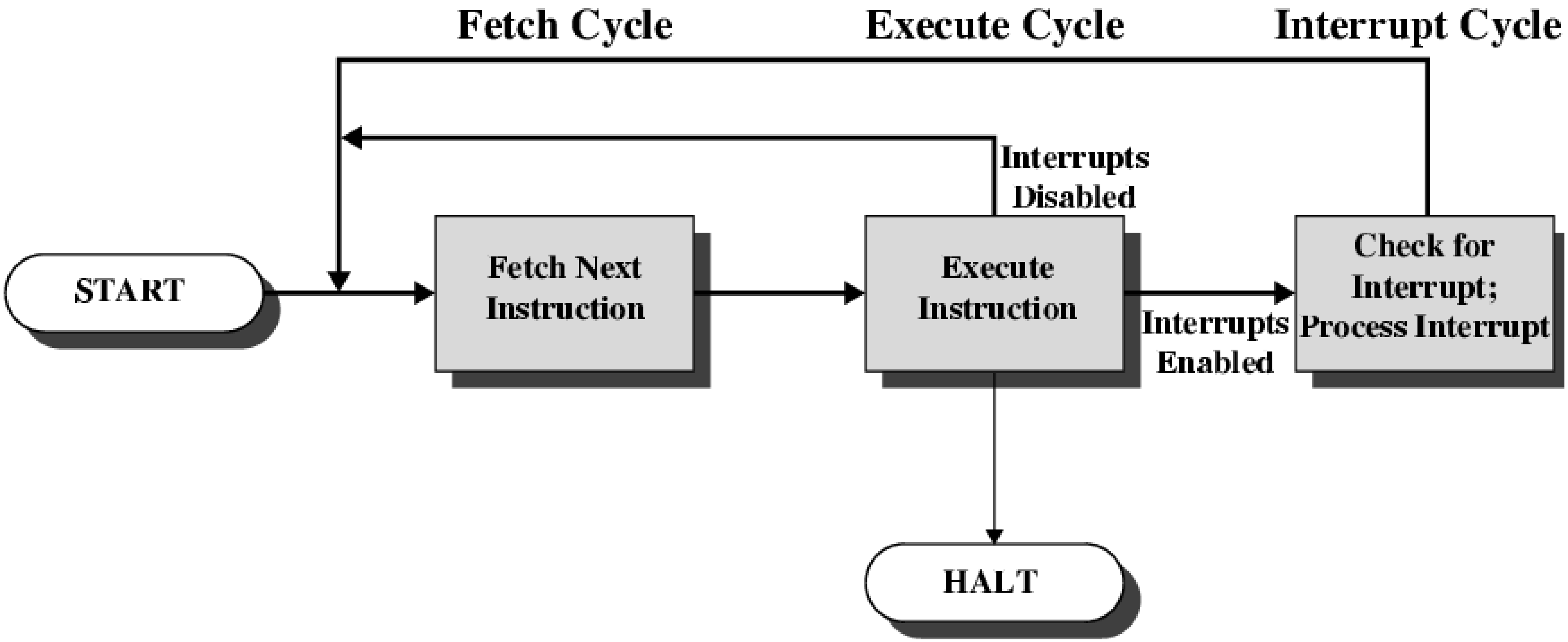
Harvard Architecture

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



Instruction Cycle Diagram

- Instruction execution : Three steps:
  - Fetch
  - Execute
  - Interrupt



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
- Increment PC
  - Unless told otherwise

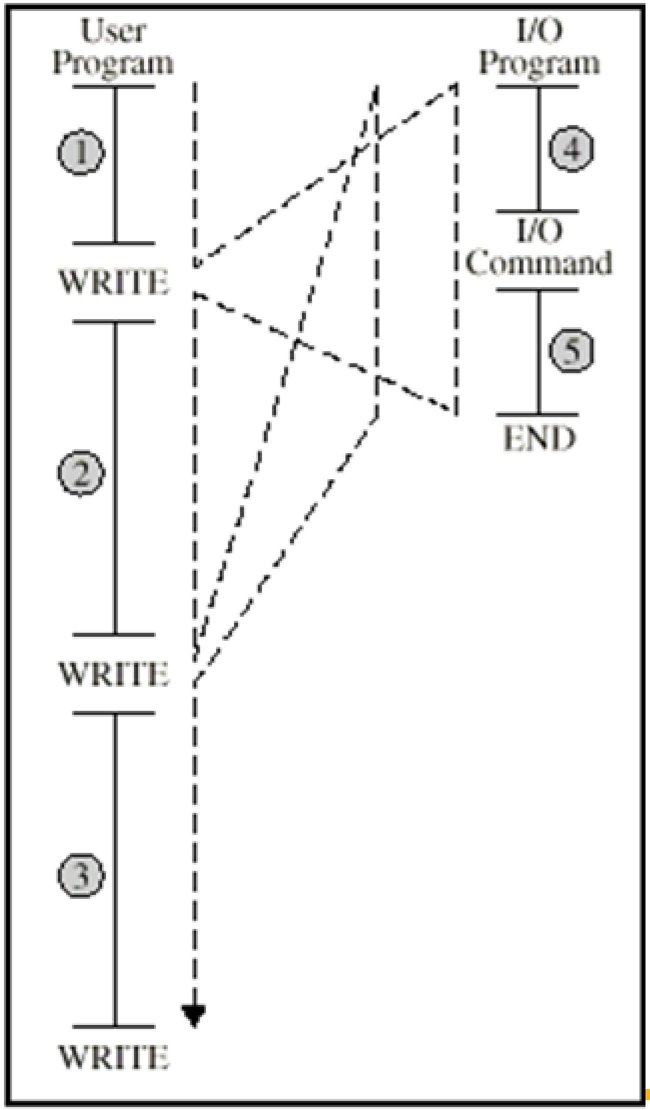
Execute Cycle

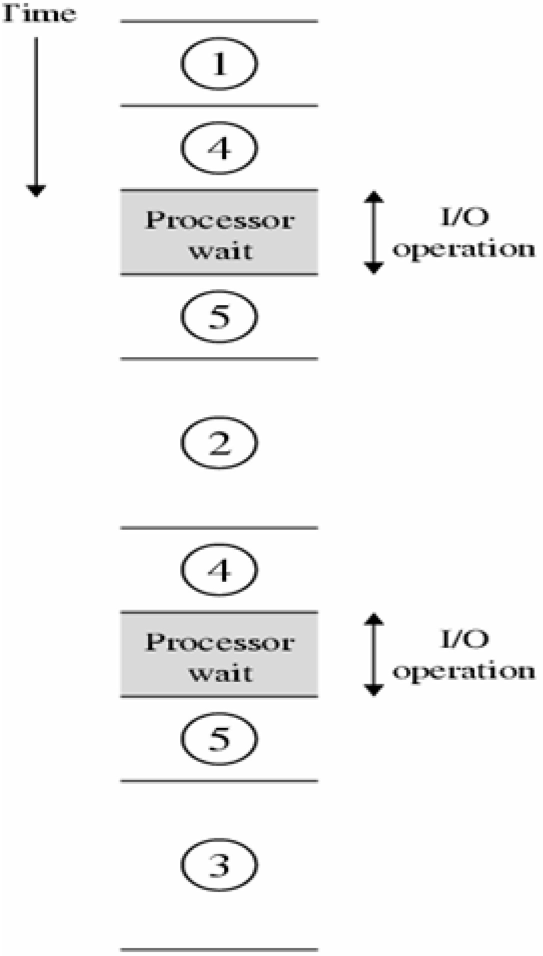
- Processor - memory
  - Data transfer between CPU and main memory
- Processor - I/O
  - Data transfer between CPU and I/O module
- Data processing
  - Some arithmetic or logical operation on data
- Control
  - Alteration of sequence of operations
  - e.g. jump
- Combination of above

Interrupt Cycle

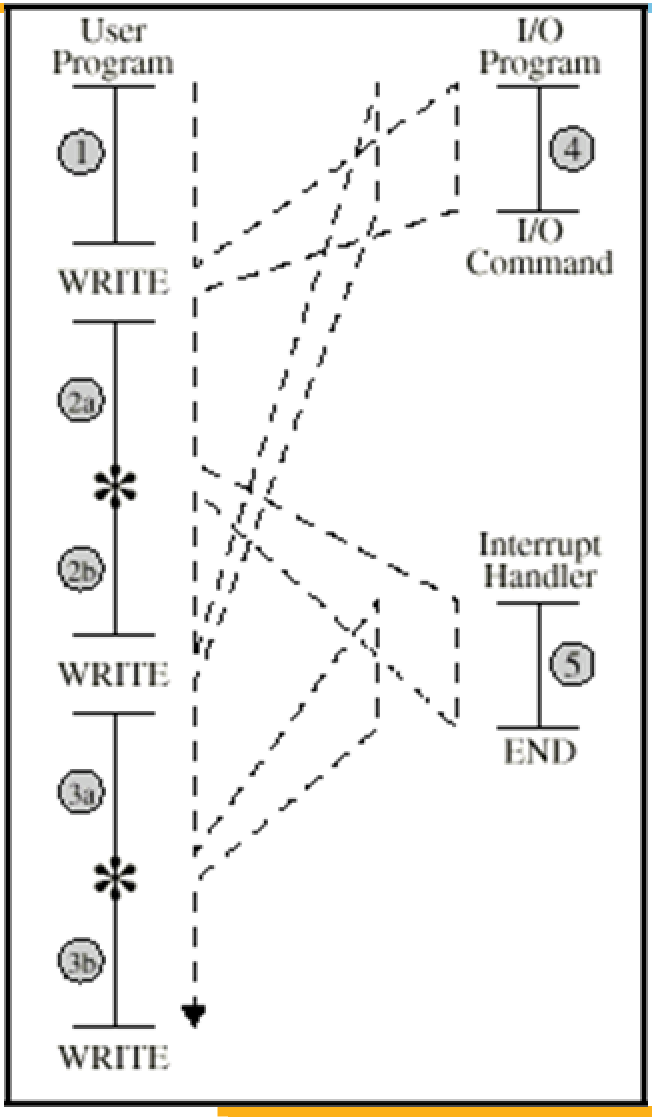
- Interrupts: Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing
- Interrupts enhances processing efficiency

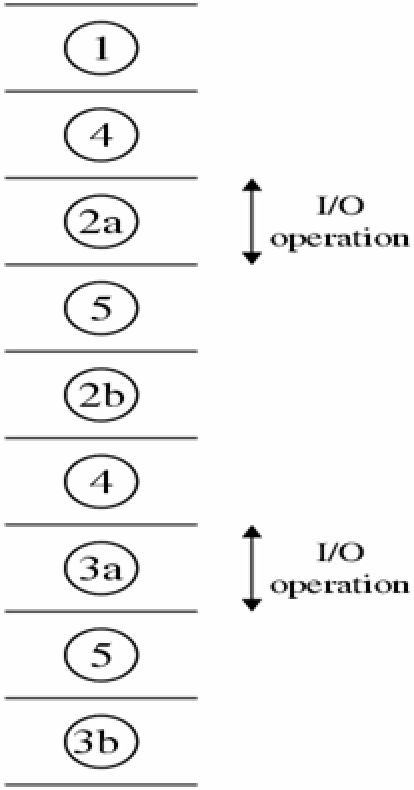
Program Flow Control (No Interrupts)





Program Flow Control





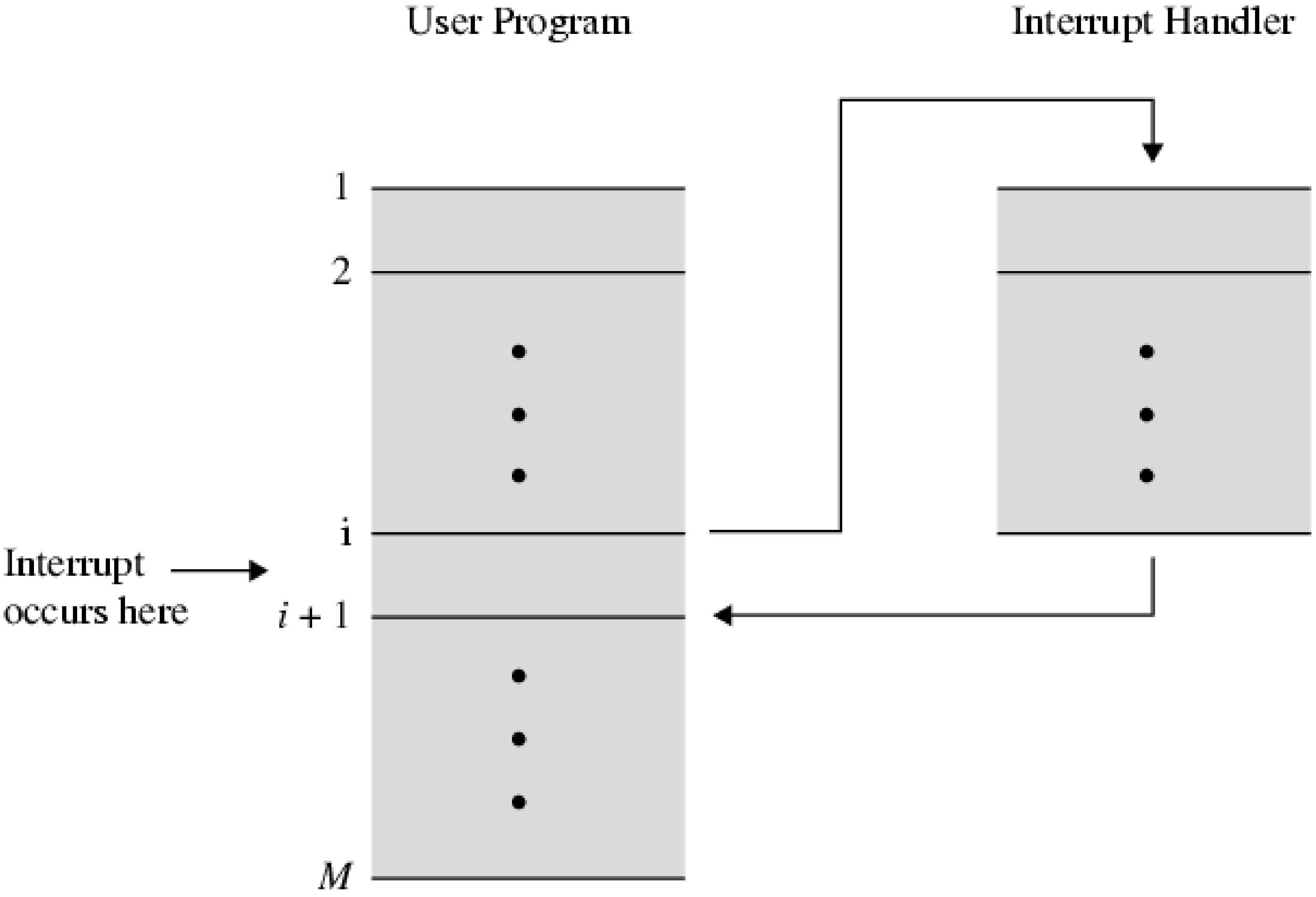
Contd…

- Classes 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

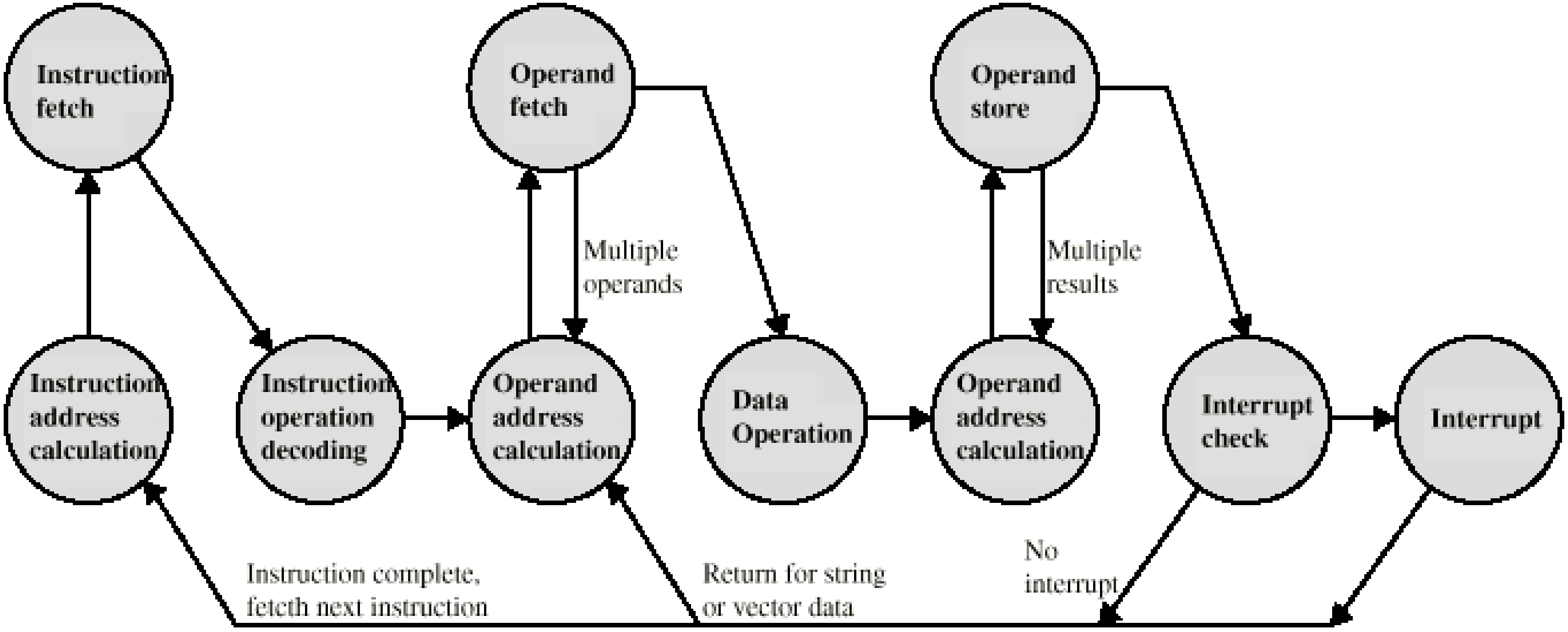
Interrupt Cycle

- Added to instruction 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

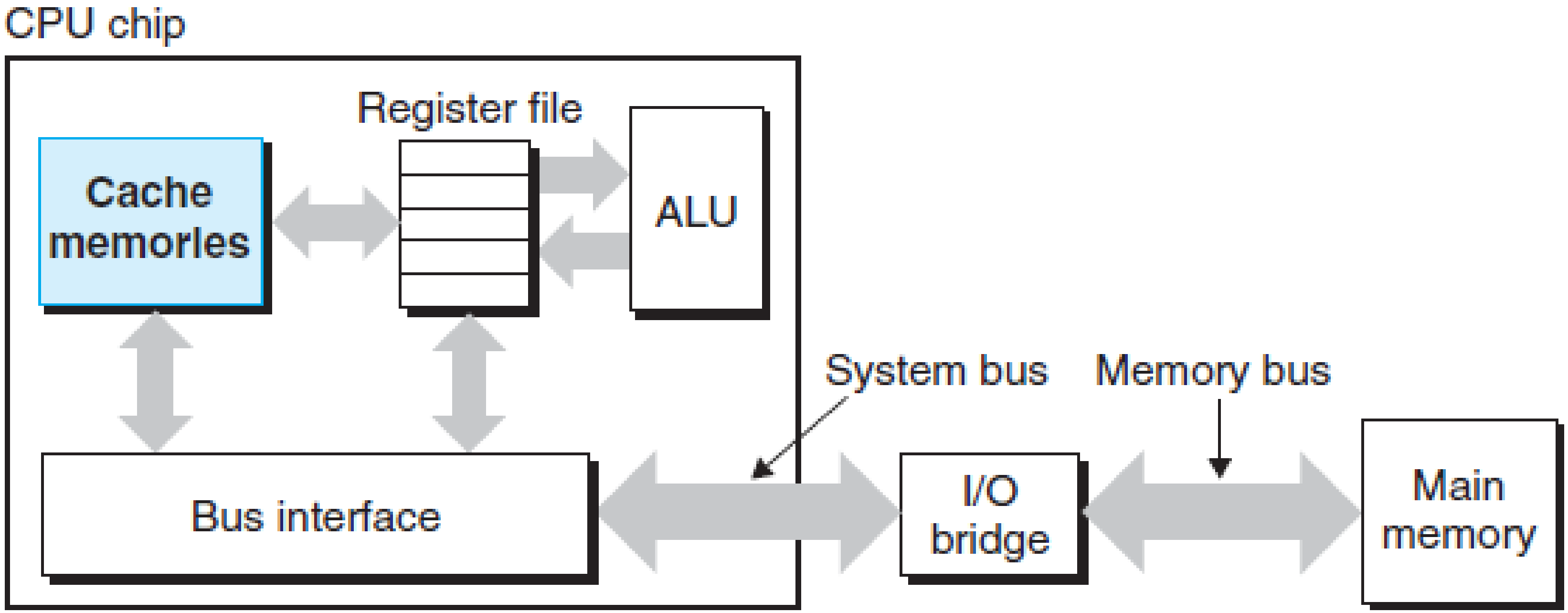
Transfer of Control via Interrupts



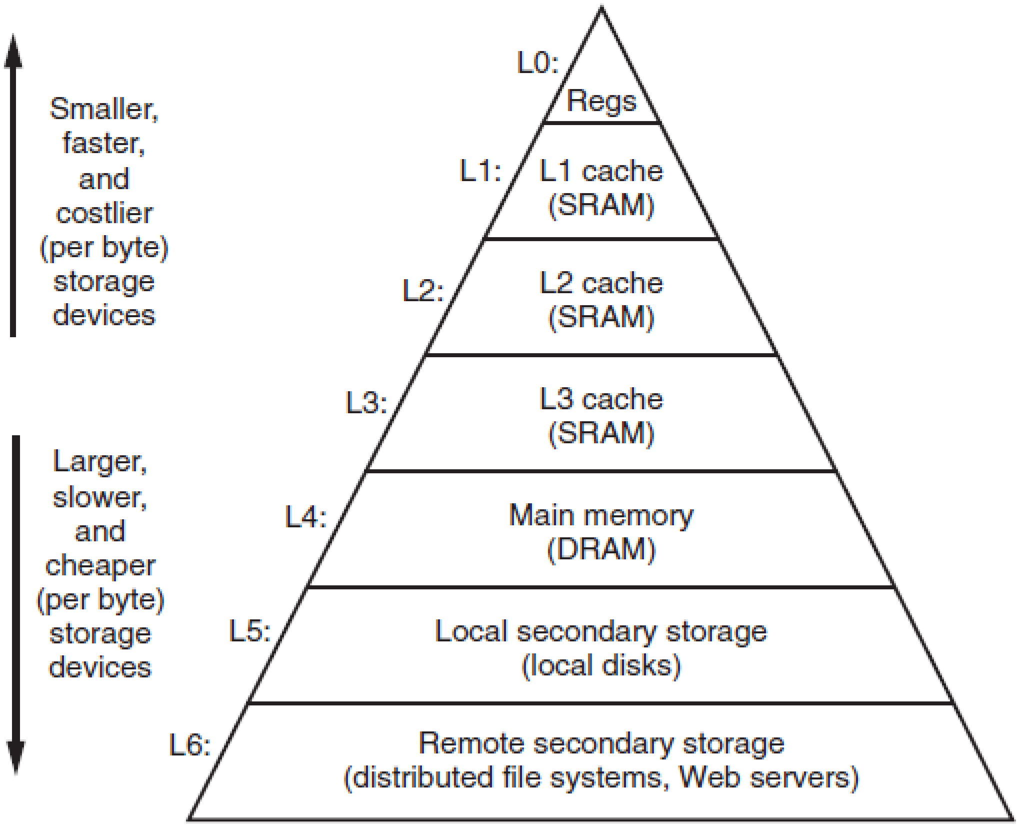
Instruction Cycle - State Diagram


Instruction fetch
Operand fetch
Operand store
Multiple operands
Multiple results
Instruction address calculation
Instruction operation decoding
Operand address calculation
Data Operation
Operand address calculation
Interrupt check
Interrupt
Instruction complete, fetcth next instruction
Return for string or vector data
No interrupt

Role of Cache Memory



Memory Hierarchy


Smaller, faster, and costlier (per byte) storage devices
Larger, slower, and cheaper (per byte) storage devices
L0:
Regs
L1:
L1 cache (SRAM)
L2:
L2 cache (SRAM)
L3:
L3 cache (SRAM)
L4:
Main memory (DRAM)
L5:
Local secondary storage (local disks)
L6:
Remote secondary storage (distributed file systems, Web servers)

An example of a memory hierarchy.

Operating System

- collection of software/ Program that acts as an intermediary between an user of a computer and the computer hardware.
- Three main functions:
  - Resource management
  - Establish a user interface
  - Execute and provide services for application software
- is a program that helps to run all the other programs
- OS is a **resource allocator**
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

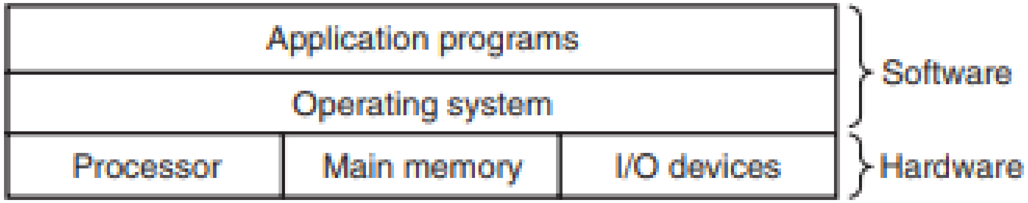
What if no Operating System?

- We need a mechanism to
  - ✓Load the program into main memory
  - ✓Run the program in processor
  - ✓Store the result in persistent storage and
  - ✓Unload the program to release memory [for the next program to use]

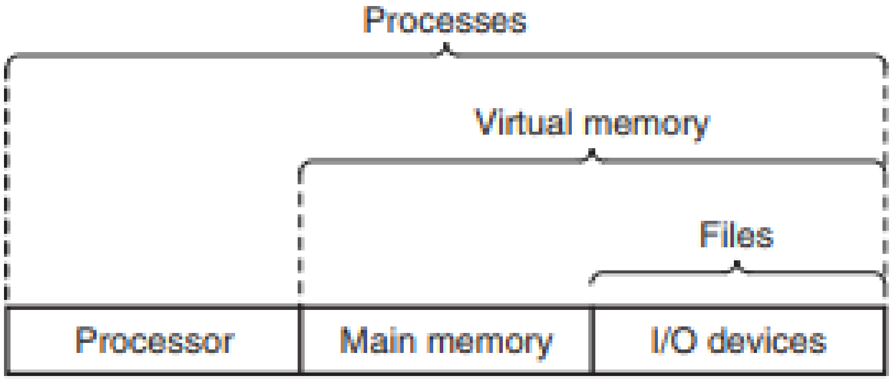
Main objectives

- **Convenience**
  - An OS makes a computer more convenient to use.
  - Provides Ease of operation
- **Efficiency**
  - Provides efficient resource management
- ***Ability to evolve and offer new services***
  - An OS should be constructed in such a way as to permit the effective development, testing, and introduction of new system functions without interfering with service.
- Maximize ***System performance***
- Protection and access control
- Footprint of OS should be small

Operating System role in Managing Computer Hardware



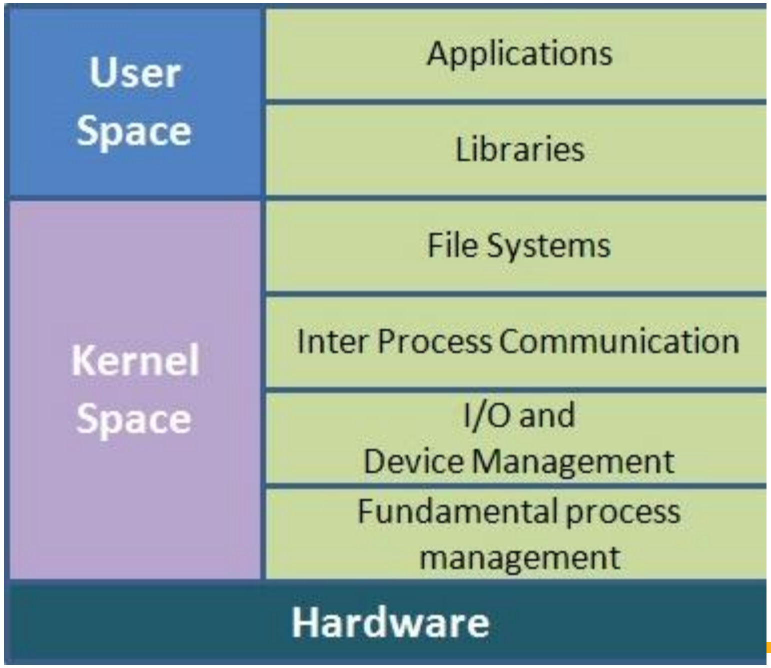
Layered view of a computer system.



Abstractions provided by an operating system.

Important Note

- "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
- Monolithic Vs. Microkernel



User Space

Applications

Libraries

Kernel Space

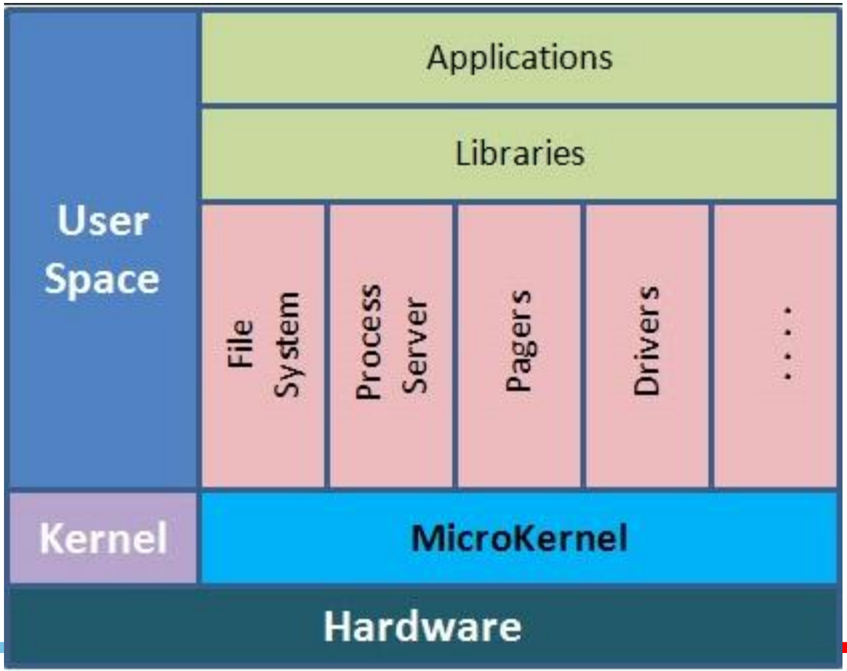
File Systems

Inter Process Communication

I/O and Device Management

Fundamental process management

Hardware



User Space

Applications

Libraries

File System

Process Server

Pagers

Drivers

. . .

Kernel

MicroKernel

Hardware

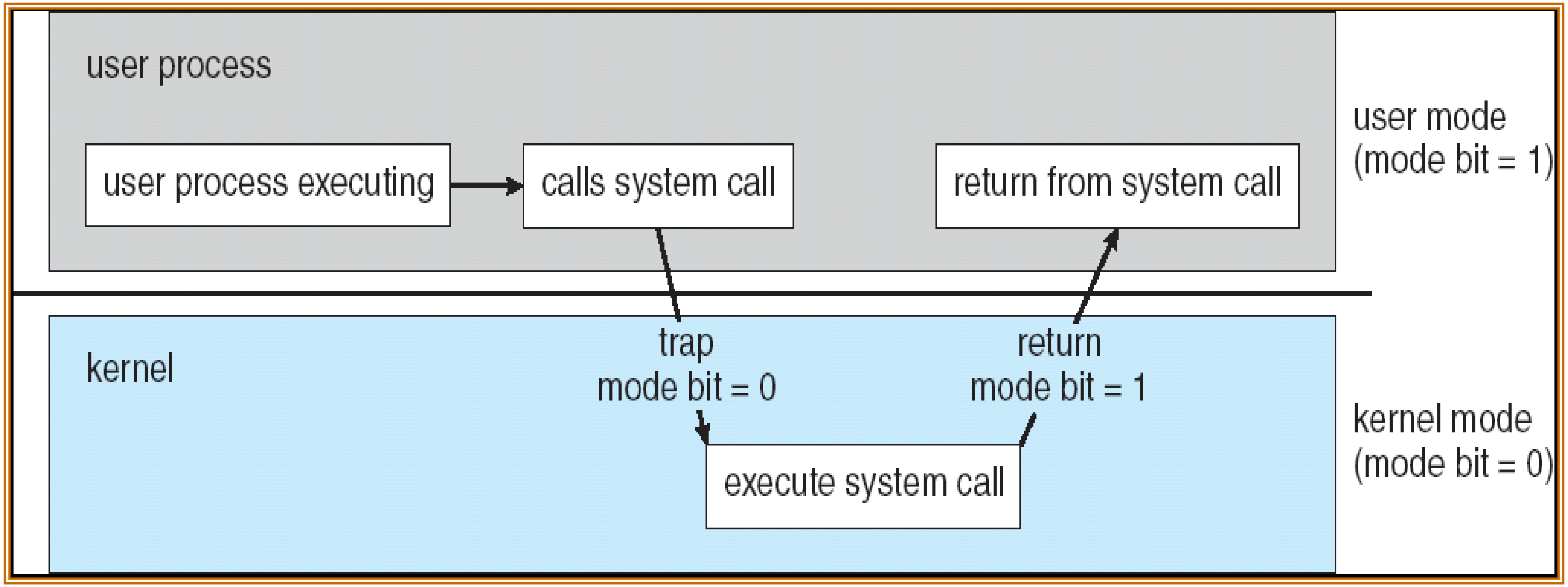
Important Note

- **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
  - System processes or System Daemons
    - syslogd
  - init

Operating System Operations

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

Transition from user to kernel mode



Contd…

- **Mode bit** provided by hardware
  - Provides ability to distinguish when system is running user code or kernel code
  - Some instructions designated as **privileged**, only executable in kernel mode
  - System call changes mode to kernel, return from call resets it to user
- Software error or request creates **exception** or **trap**
  - Division by zero, request for operating system service
- Other process problems include infinite loop, processes modifying each other or the operating system

Services provided by the OS

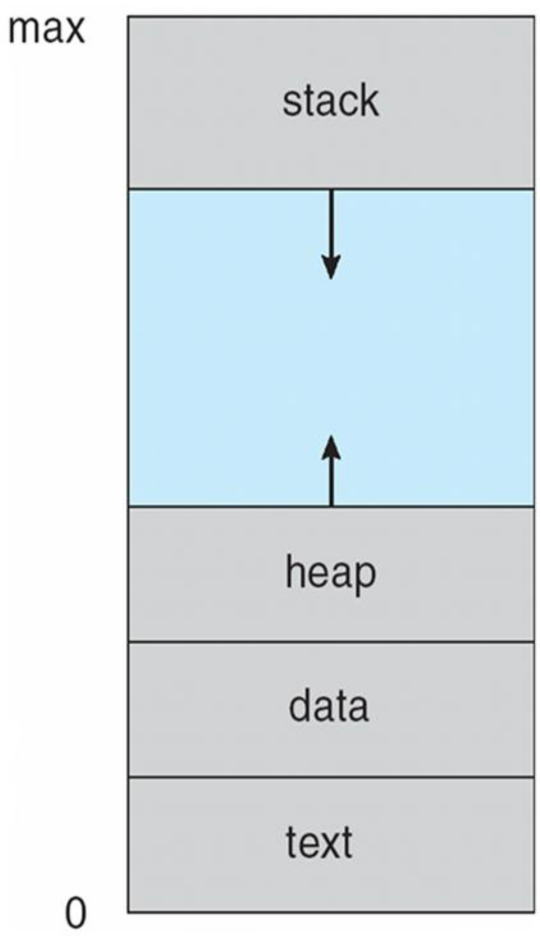
- Process Management
- Memory Management
- Storage Management
- Protection and security

Process

- ✓ A program in execution
- ✓ Needs resources : CPU, memory, files, I/O devices
- ✓ A program becomes process when executable file loaded into memory
- ✓ Process execution must progress in sequential fashion
- ✓ word processor, a Web browser and an e-mail package are different processes.
- ✓ Types: System processes and user processes

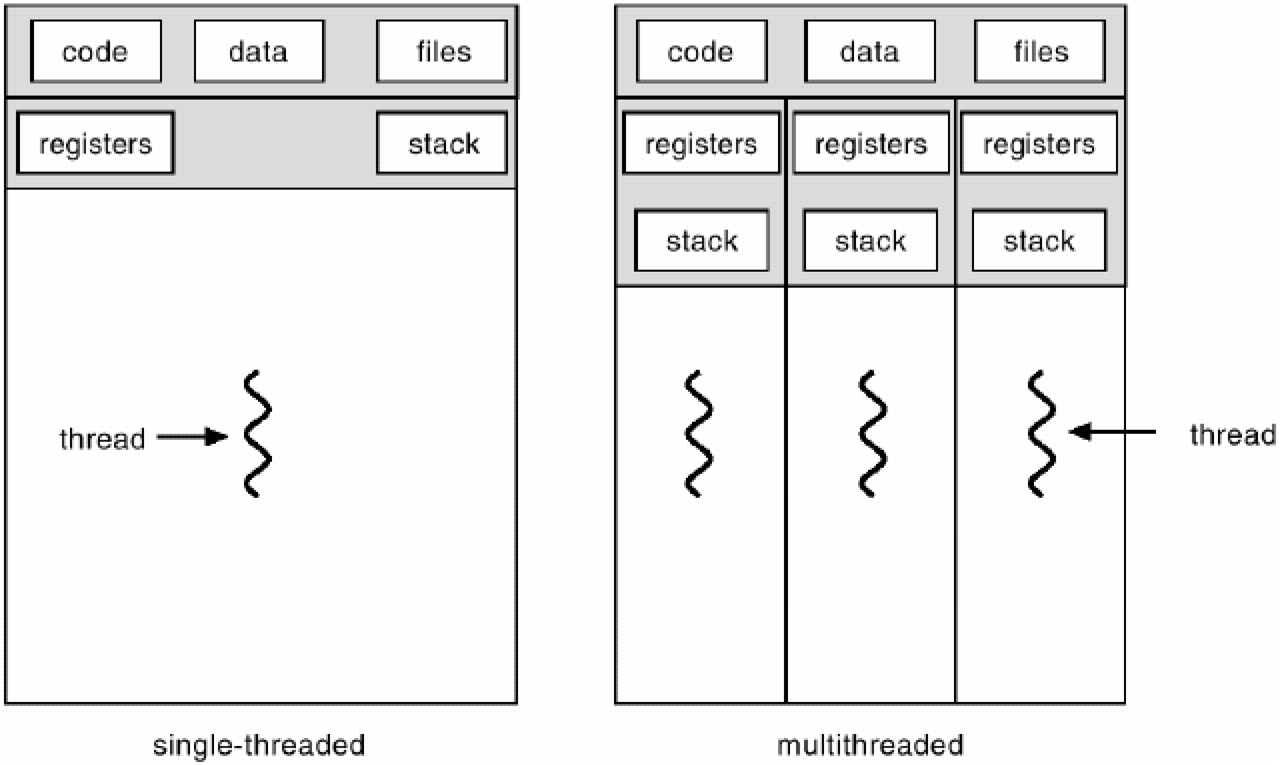
Process in Memory

- The text section: comprises the compiled program code.
- The data section: stores global and static variables, allocated and initialized prior to executing main.
- The heap: is used for dynamic memory allocation.
- The stack: is used for local variables. Space on the stack is reserved for local variables when they are declared and the space is freed up when the variables go out of scope stack is also used for function return values.



Threads

- A thread is a single sequence stream within in a process



Virtual Memory

- Virtual memory is a technique that allows the execution of processes that are not completely in memory.
- Motivation:
  - Programs often have code to handle unusual error conditions which is almost never executed.
  - Arrays, lists, and tables are often allocated more memory than they actually need.
  - Certain options and features of a program may be used rarely
  - Principle of locality
  - trashing is a condition where system spends more time in swapping than executing instructions

Files

- Is a sequence of bytes
- I/O device such as disks, keyboards, displays, and even networks, is modeled as a file
- Reading and writing requires set of system calls