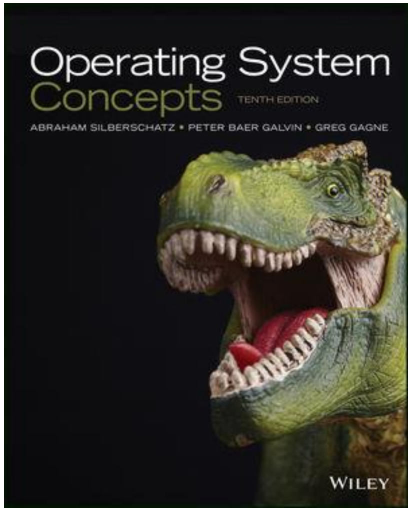
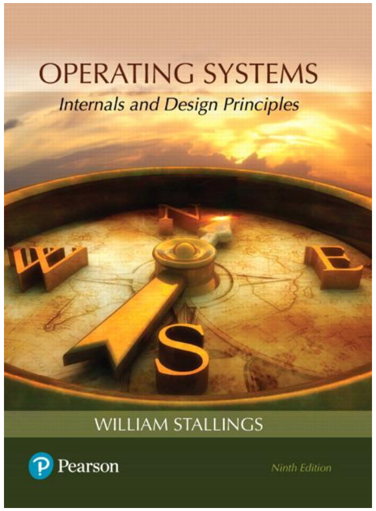
Sistemas Operativos

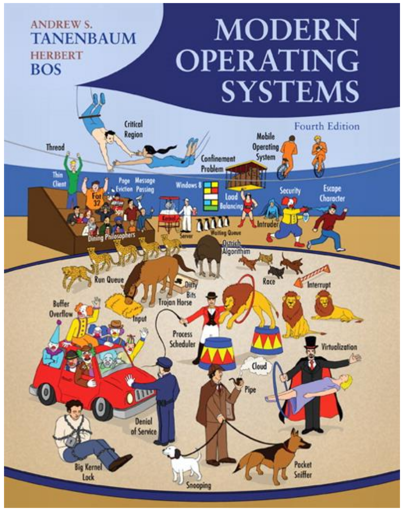
Operating System Operation

Disclaimer

- These slides and notes are based on the contents of the books:
  - Abraham Silberschatz, ” Operating System Concepts”, 10th Edition, Wiley, 2018;
  - William Stallings, “Operating Systems: Internals and Design Principles”, 9th Edition, Pearson, 2017;
  - Andrew S. Tanenbaum, “Modern Operating Systems”, 4th Edition, Pearson Education, 2014;
- The respective copyrights belong to their owners.





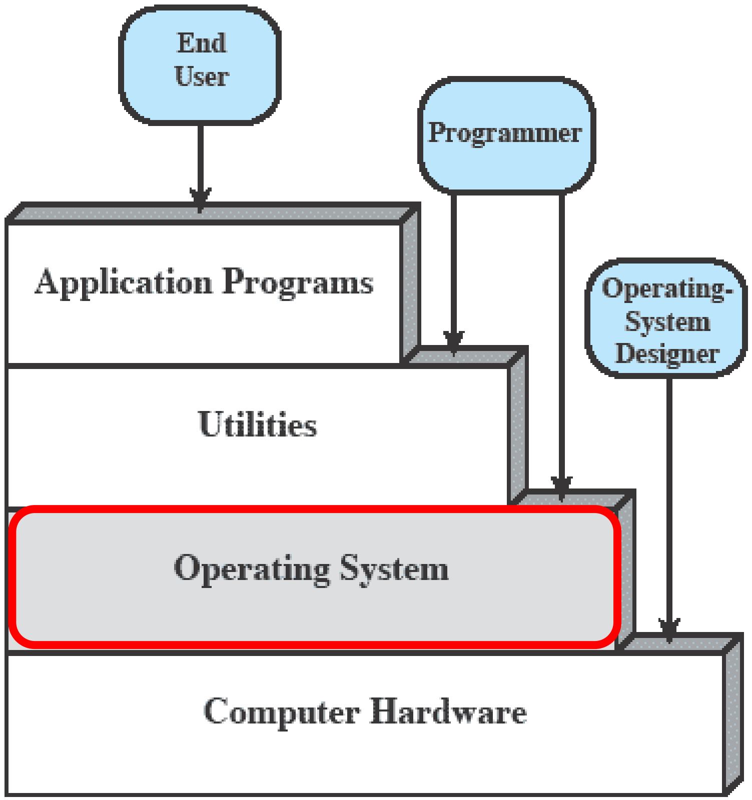


Outline

- Operating System Definition
- Computer System Components
- Operating System Operation
  - Fetch-Execute Cycle
  - Interrupts
  - I/O Operation
    - Programmed Driven I/O
    - Interrupted Driven I/O
  - Modes of Operation
    - user mode
    - kernel mode
  - Multiprogramming
  - Time sharing

Operating System Definition

- An **Operating System** (O.S.) is
  - a program that acts as an intermediary between a user of a computer and the computer hardware:
    - **manages the computer hardware**, deciding how to allocate the resources between users and programs
    - **controls** execution of programs to prevent errors and improper use of the computer.



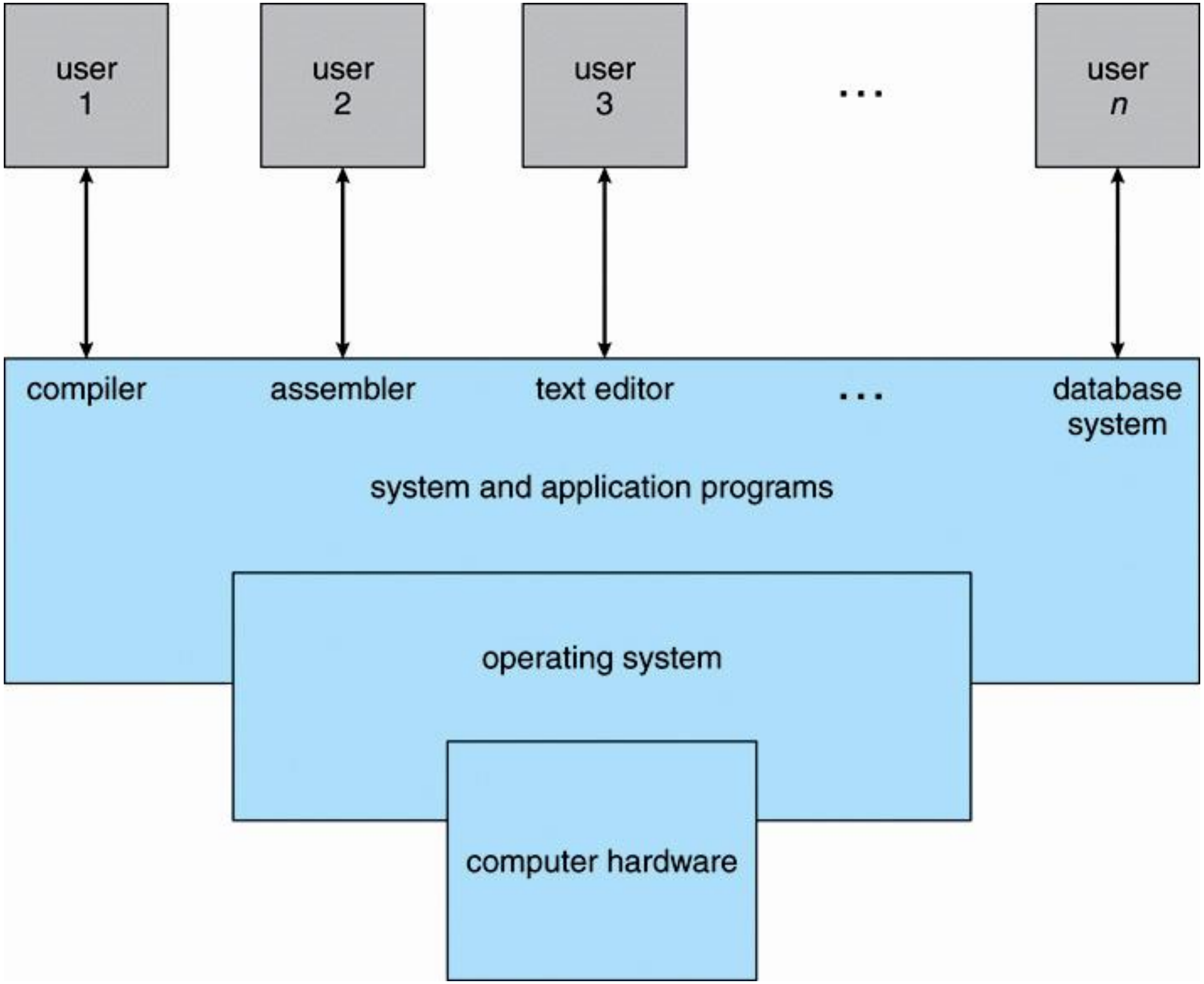
William Stallings, "Operating Systems: Internals and Design Principles"

Outline

- Operating System Definition
- Computer System Components
- Operating System Operation
  - Fetch-Execute Cycle
  - Interrupts
  - I/O Operation
    - Programmed Driven I/O
    - Interrupted Driven I/O
  - Modes of Operation
    - user mode
    - kernel mode
  - Multiprogramming
  - Time sharing

Computer System Components

- a computer system can be divided into four components:



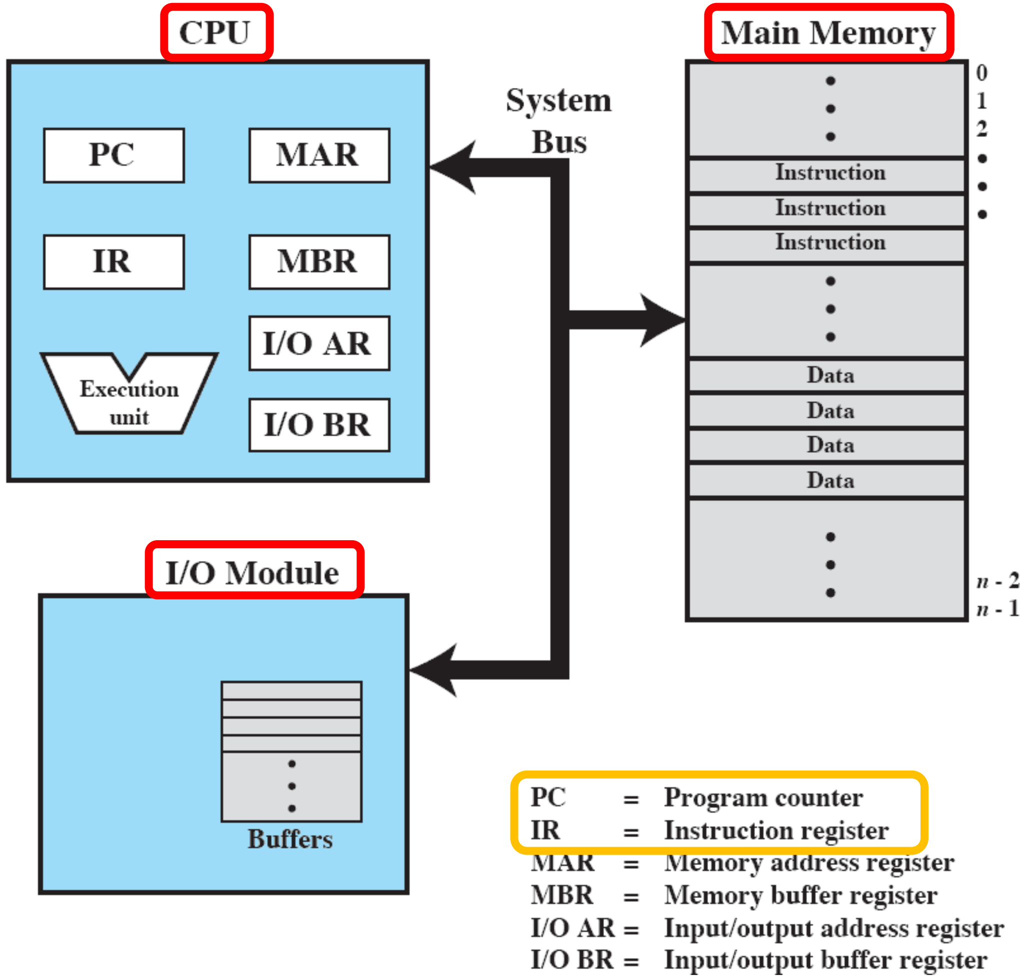
Abraham Silberschatz, " Operating System Concepts"

Computer System Components

- a computer system can be divided into four components:
  - **Hardware:**
    - **von Neumann architecture**: CPU, memory, I/O devices
    - provides basic computing resources
  - **Operating System:**
    - controls and coordinates the use of hardware among the various applications and users
  - **Application Programs:**
    - define the ways in which the system resources are used to solve the computing problems of the users
  - **Users**
    - people, machines, other computers

Computer System Components

- a computer system can be divided into four components:
  - **Hardware:**
    - **von Neumann architecture**: CPU, memory, I/O devices

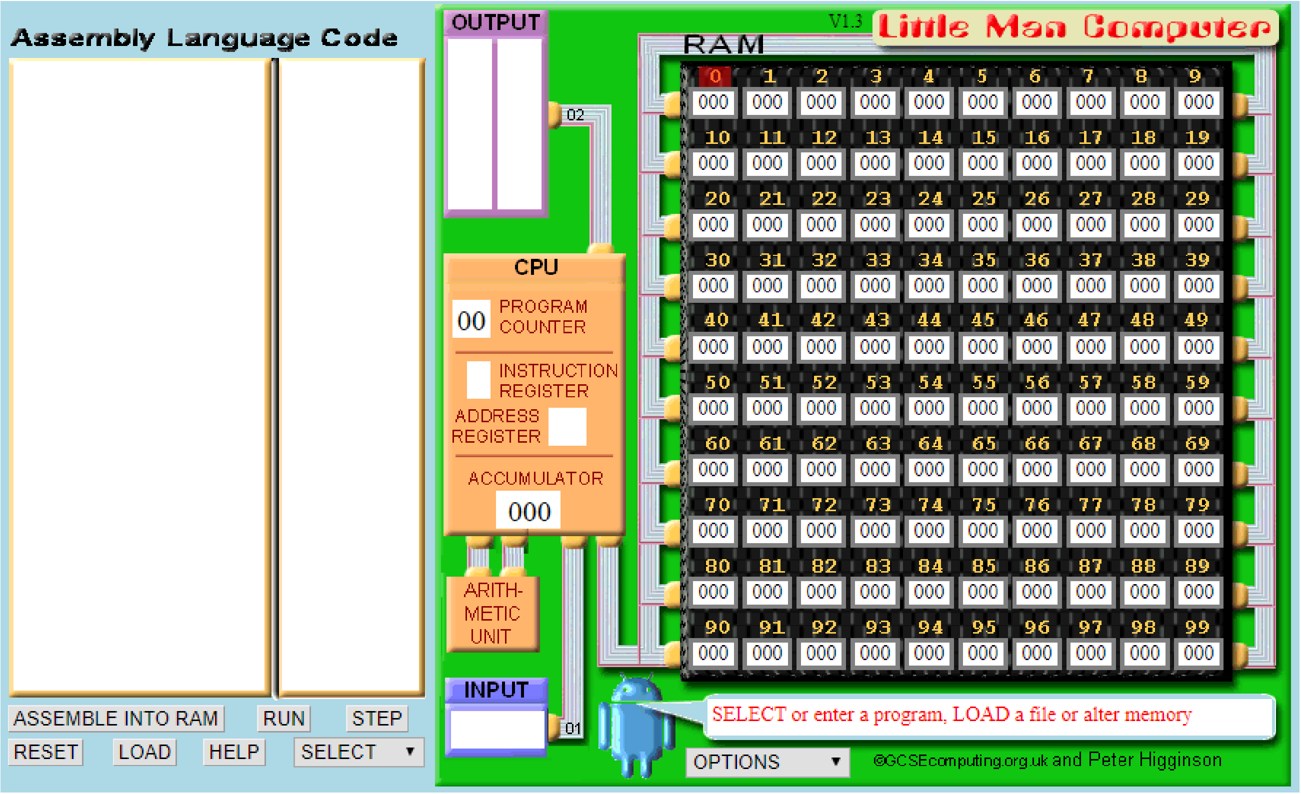


William Stallings, "Operating Systems: Internals and Design Principles"

Computer System Components

- a computer system can be divided into four components:
  - **Hardware:**
    - **von Neumann** architecture: CPU, memory, I/O devices

O **Little Man Computer (LMC)** consiste num modelo simplificado de um computador, que <u>modela a arquitetura von Neumann</u>, podendo ser programado em linguagem máquina (decimal em vez de binário), ou linguagem assembly (com somente 11 instruções). A sua implementação encontra-se disponível em diversas plataformas, nomeadamente em versão web.

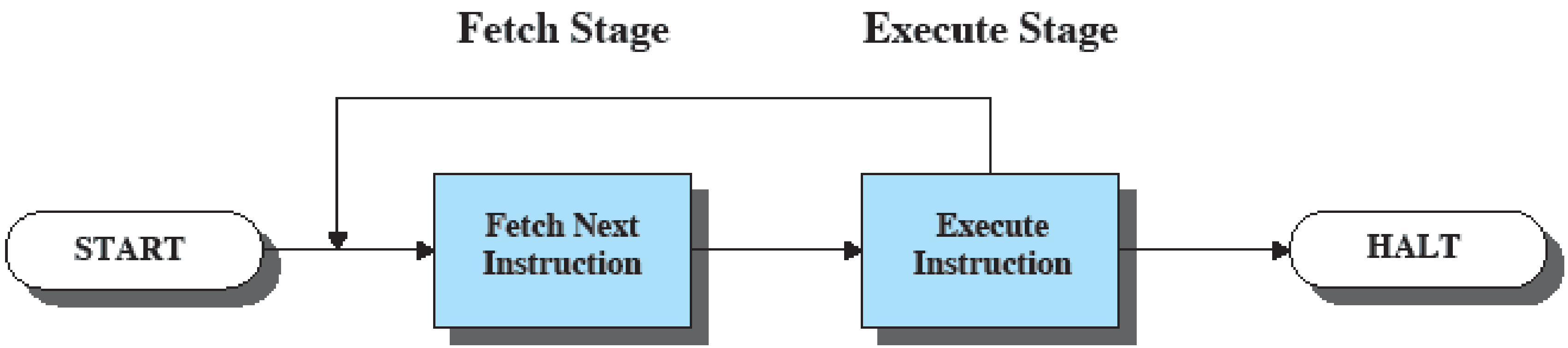


http://www.peterhigginson.co.uk/LMC/

Outline

- Operating System Definition
- Computer System Components
- Operating System Operation
  - Fetch-Execute Cycle
  - Interrupts
  - I/O Operation
    - Programmed Driven I/O
    - Interrupted Driven I/O
  - Modes of Operation
    - user mode
    - kernel mode
  - Multiprogramming
  - Time sharing

- **Fetch-Execute Cycle**



William Stallings, “Operating Systems: Internals and Design Principles”

1) **Processor reads (fetches) instructions from memory**
   - The processor fetches the instruction from memory. The register ***Program Counter (PC)*** holds the address of the instruction to be fetched next. PC is incremented after each fetch.

- **Fetch-Execute Cycle**

  2) **Processor executes each instruction**

     - The fetched instruction is loaded into the register called ***Instruction Register*** **(IR),** and it specifies the action required by the processor. Categories:

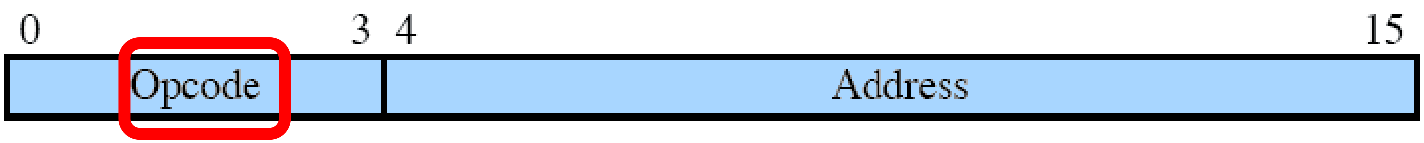
       \- Processor-memory: data transferred from memory to processor or vice versa.

       \- Processor-I/O: data exchanged between the external device through an I/O module.

       \- Data Processing: the processor may perform some arithmetic or logical action on the data.

       \- Control: an instruction may alter the execution by specifying a different location, which is normally done by loading PC register with that particular address.

- **Fetch-Execute Cycle**
- Example:
  - add two numbers and store the result at some place in the memory



(a) Instruction format

(b) Integer format

Partial List of **Opcodes**

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

William Stallings, "Operating Systems: Internals and Design Principles"

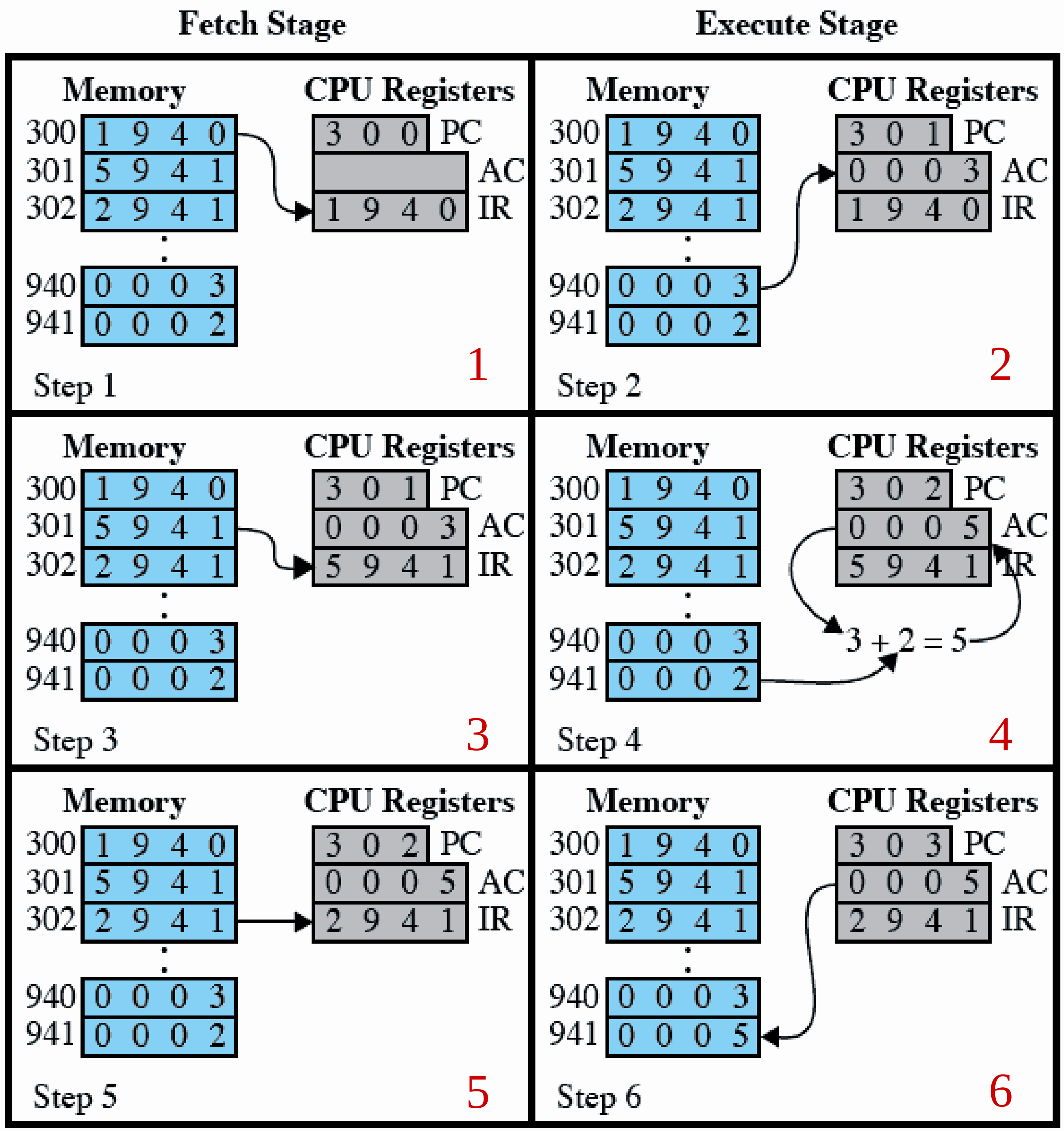
Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

Next instruction is at 300 address. The contents are loaded in to the IR.

Next instruction is fetched.

Next instruction is fetched.



PC is incremented. The contents at 300 location: first four bits are the Op-code (0001 = 1) and the remaining 12 bits indicate the address (9 4 0). The contents of the 940 location are loaded into the AC.

PC is incremented. Instruction according to the Op-code (0101 = 5) says to add the contents of AC (0 0 0 3) with the given address (9 4 1) contents (0 0 0 2).

PC is incremented. Next instructions Op-code (0010 = 2) says store the contents of AC to location (9 4 1).

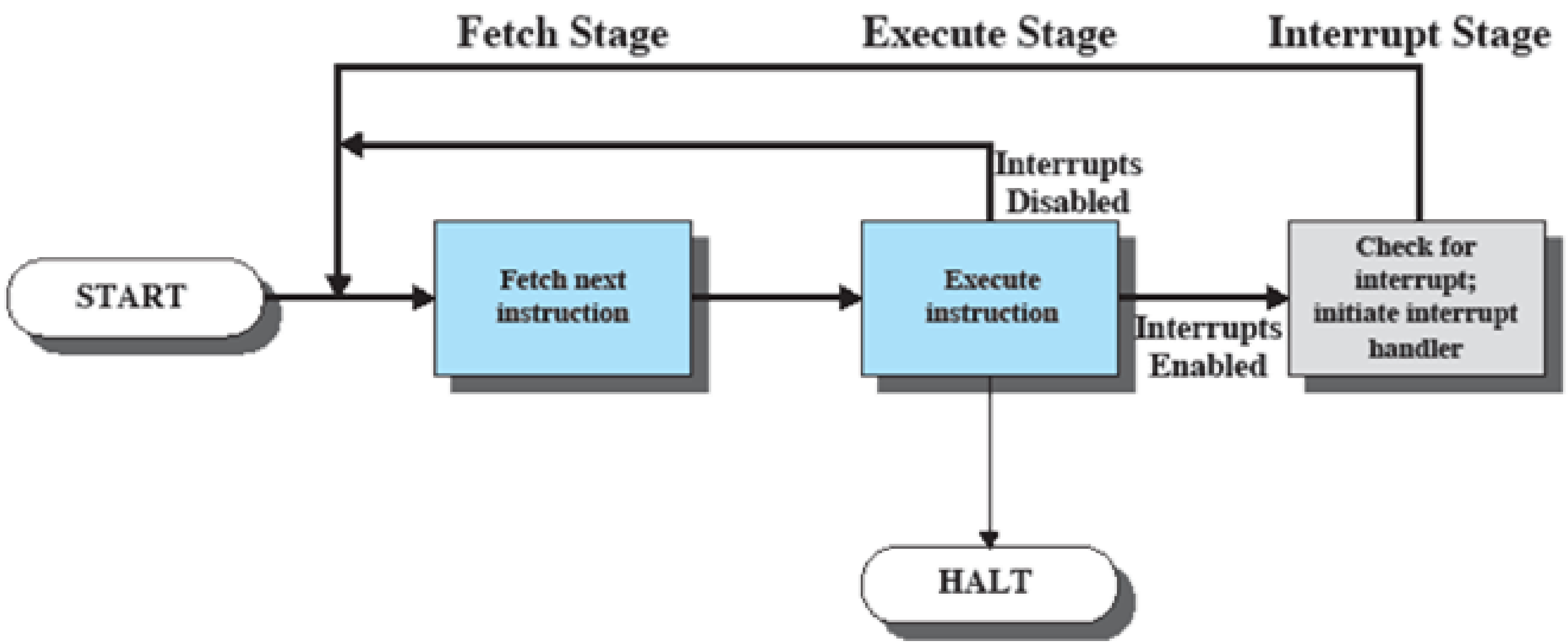
William Stallings, “Operating Systems: Internals and Design Principles”

Outline

- Operating System Definition
- Computer System Components
- Operating System Operation
  - Fetch-Execute Cycle
  - Interrupts
  - I/O Operation
    - Programmed Driven I/O
    - Interrupted Driven I/O
  - Modes of Operation
    - user mode
    - kernel mode
  - Multiprogramming
  - Time sharing

Operating System Operation

- **Interrupts**
  - Modern operating systems are **interrupt driven.**
  - For each type of interrupt, separate segments of code in the operating system determine what action should be taken.
  - An **interrupt service routine** is provided that is responsible for dealing with the interrupt.



William Stallings, “Operating Systems: Internals and Design Principles”

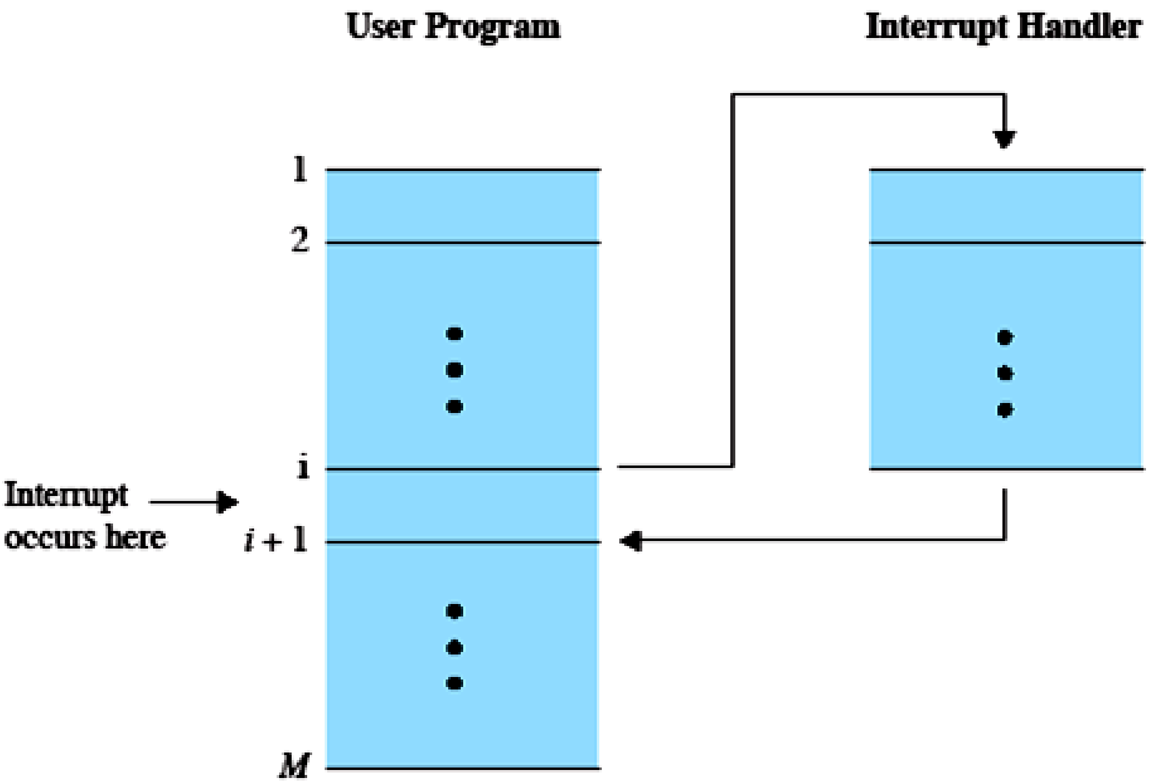
- **Interrupts**
  - **Classes of Interrupts**

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
Hardware failure	Generated by a failure, such as power failure or memory parity error.

William Stallings, “Operating Systems: Internals and Design Principles”

Operating System Operation

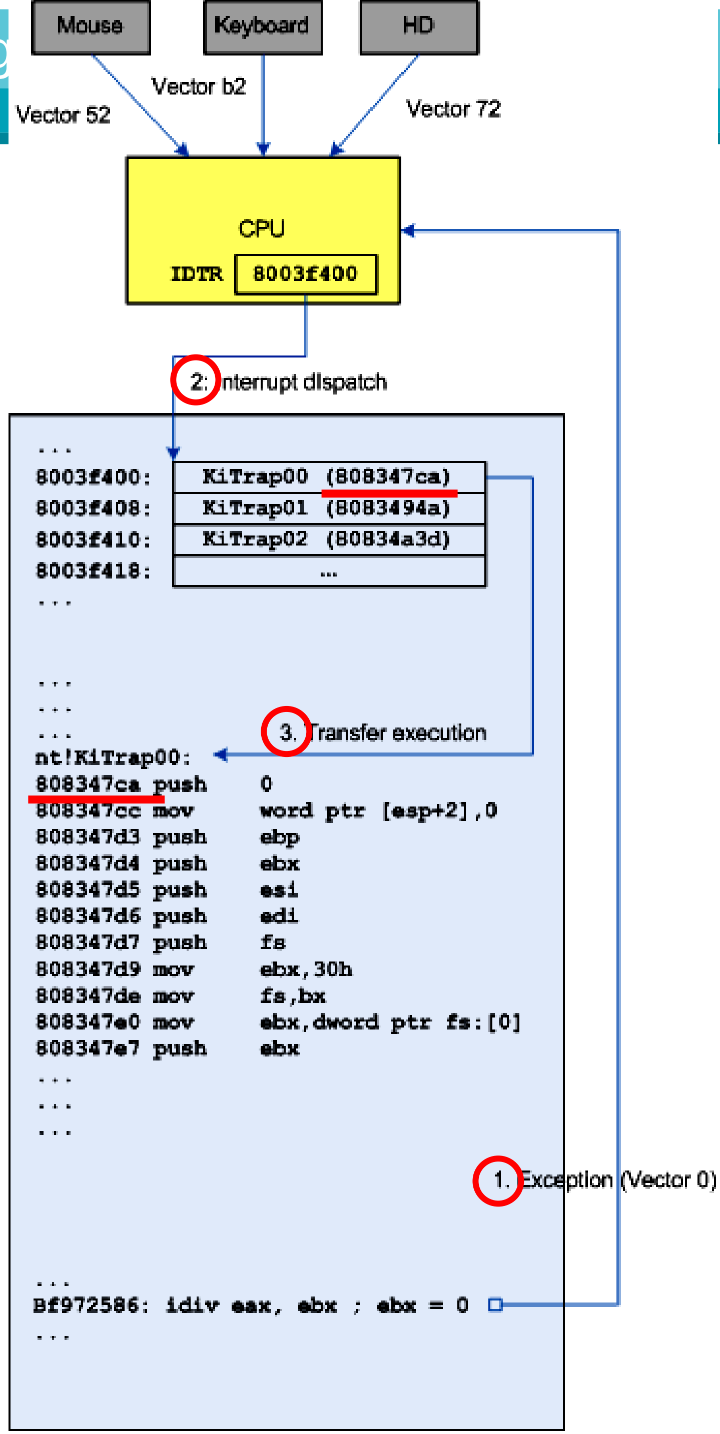
- **Interrupts**
  - When the CPU is interrupted, it stops what is doing and immediately transfers execution to a **fixed location**.
  - The fixed location usually contains the starting address where the **service routine for the interrupt** is located.
  - on completion of the interrupt service routine the CPU resumes the interrupted computation.



William Stallings, "Operating Systems: Internals and Design Principles"

- **Interrupts**
  - **Interrupt Vector Table**
  - The type of interrupt is specified by a number called **interrupt number**.
  - CPU has to transfer execution to some procedure in memory to handle that interrupt. CPU has to find that procedure:
    - a) having one procedure for all interrupts and specifying interrupt vector number as a parameter
    - b) having a table of pointers (**interrupt vector table**) to various procedures (**interrupt routines**) that correspond to different interrupt vectors.
      - Intel x86 and x64 CPUs use the latter approach
  - the interrupt routine needs explicitly save the current state and then restore that state before returning

- **Interrupts**



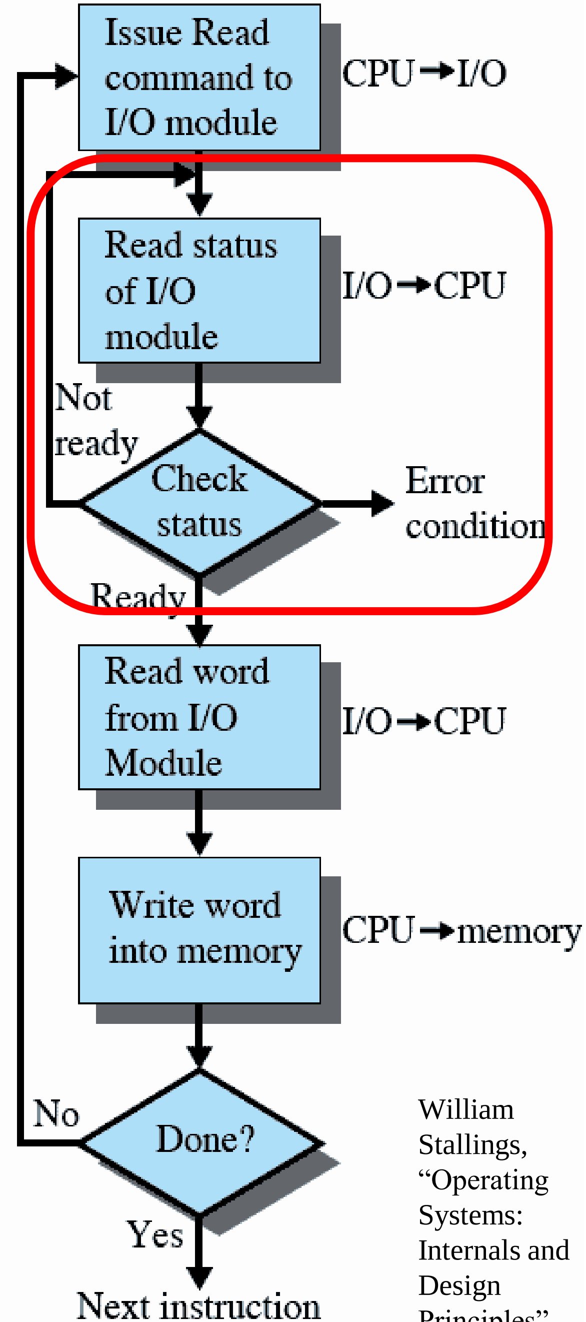
The **Interrupt Descriptor Table** (IDT) is a data structure used by the x86 architecture to implement an interrupt vector table.

Outline

- Operating System Definition
- Computer System Components
- Operating System Operation
  - Fetch-Execute Cycle
  - Interrupts
  - I/O Operation
    - Programmed Driven I/O
    - Interrupted Driven I/O
  - Modes of Operation
    - user mode
    - kernel mode
  - Multiprogramming
  - Time sharing

Operating System Operation

- **Programmed I/O**
  - the processor executes an instruction relating an I/O module by issuing command to the appropriate I/O module.
  - The I/O module will interpret the action asked by the processor and set the required bits in the status register.
    - That is the only action performed by the I/O module in the programmed I/O type of communication mode.
  - It does not perform any action further, therefore, <u>it is the duty of the processor to check the status of the I/O module periodically unless the operation is complete</u>.

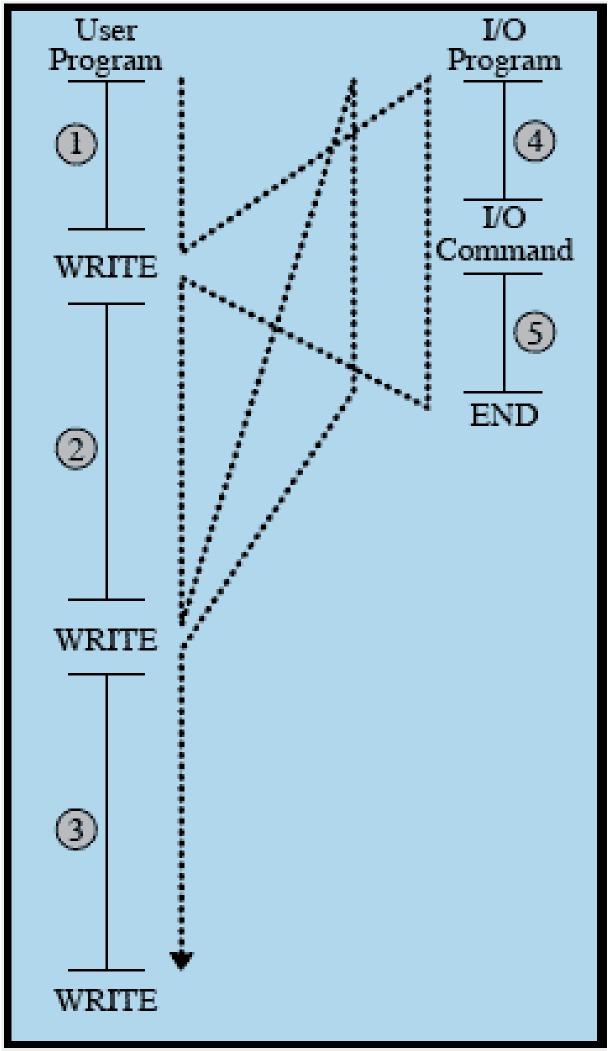


(a) Programmed I/O

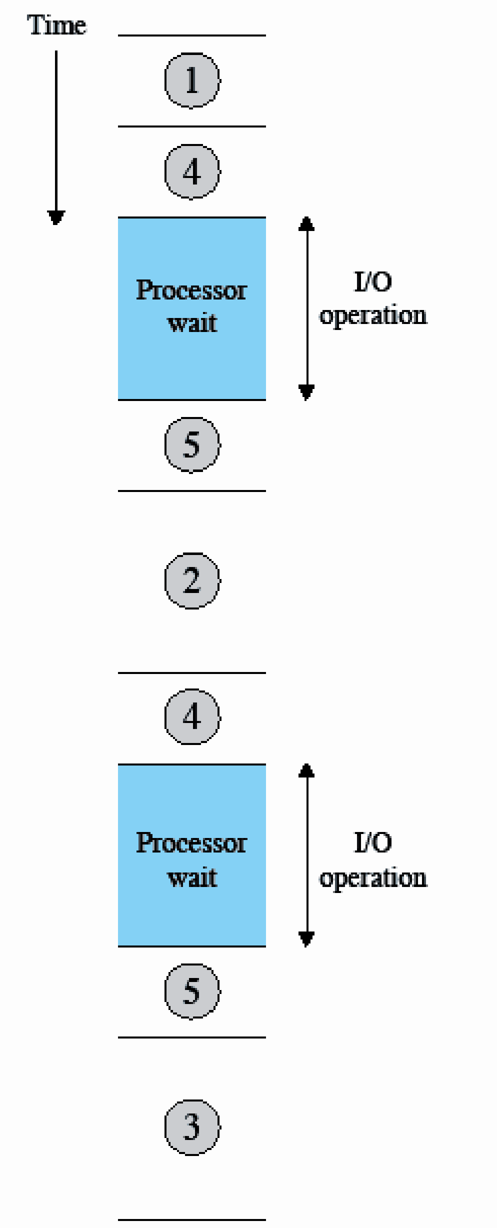
William Stallings, "Operating Systems: Internals and Design Principles"

Operating System Operation

- **Programmed I/O**

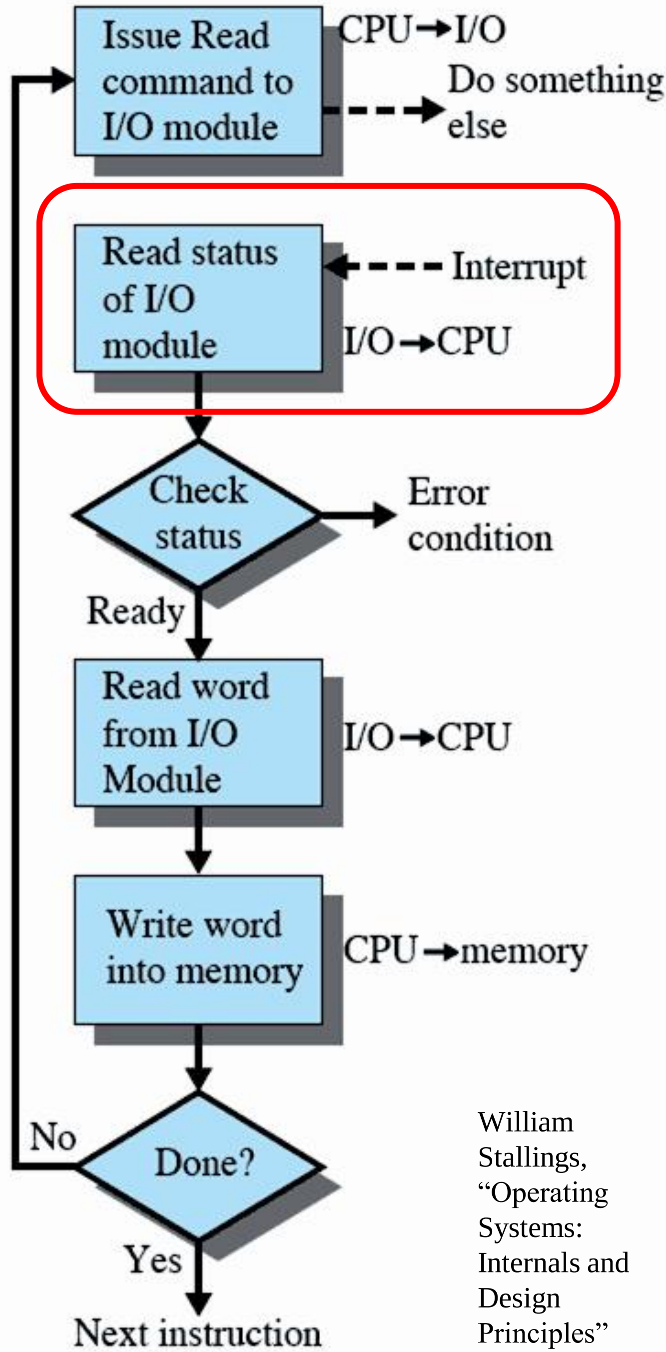


(a) No interrupts



Operating System Operation

- **Interrupted Driven I/O**
  - programmed I/O mode is not efficient since while data is being transferred between the processor (memory) and the I/O module, the CPU is taking no action but to waiting and also repeatedly checking the status of the device for further transmission between the two.
  - The efficiency can be improved if, during data transmit ion between memory and the I/O module, processor may go some where else and perform certain task and <u>the I/O module interrupt the processor when it needs its services</u>.

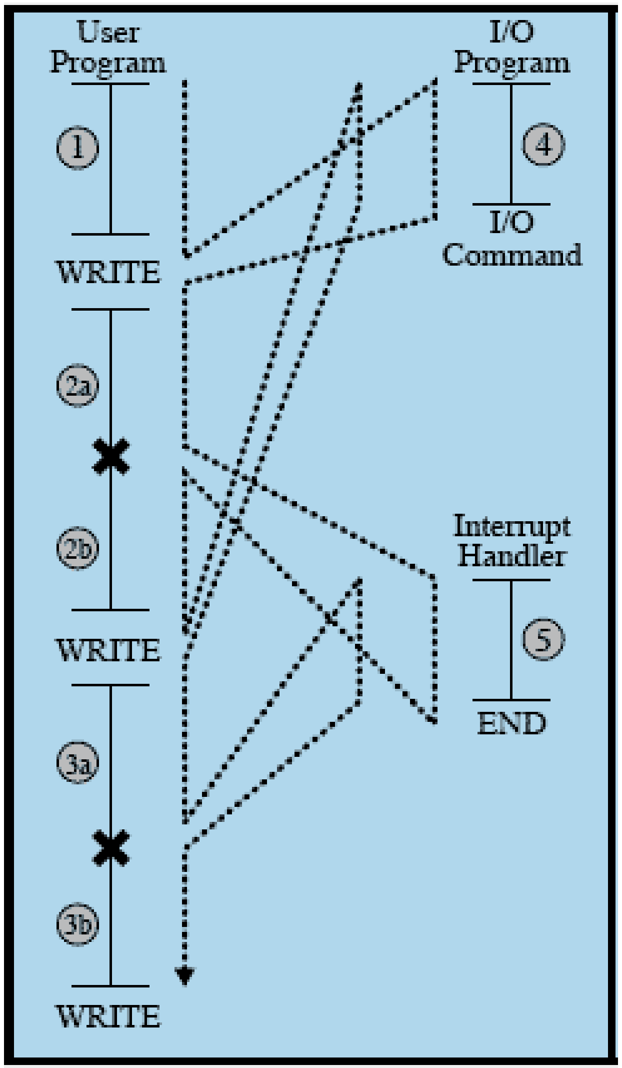


(b) Interrupt-driven I/O

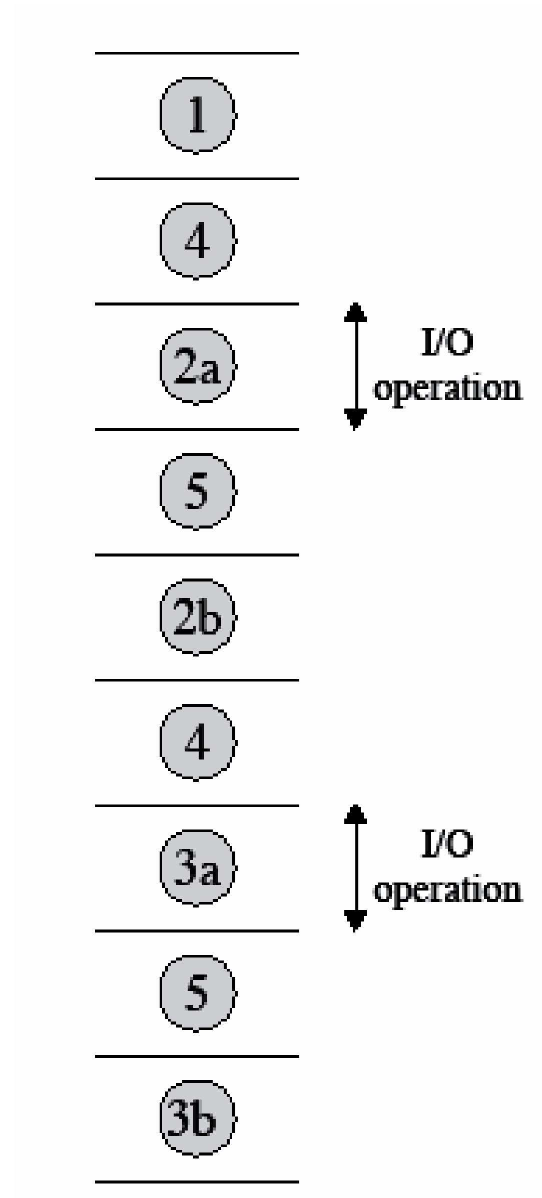
William Stallings, "Operating Systems: Internals and Design Principles"

Operating System Operation

- **Interrupted Driven I/O**



(b) Interrupts; short I/O wait



Outline

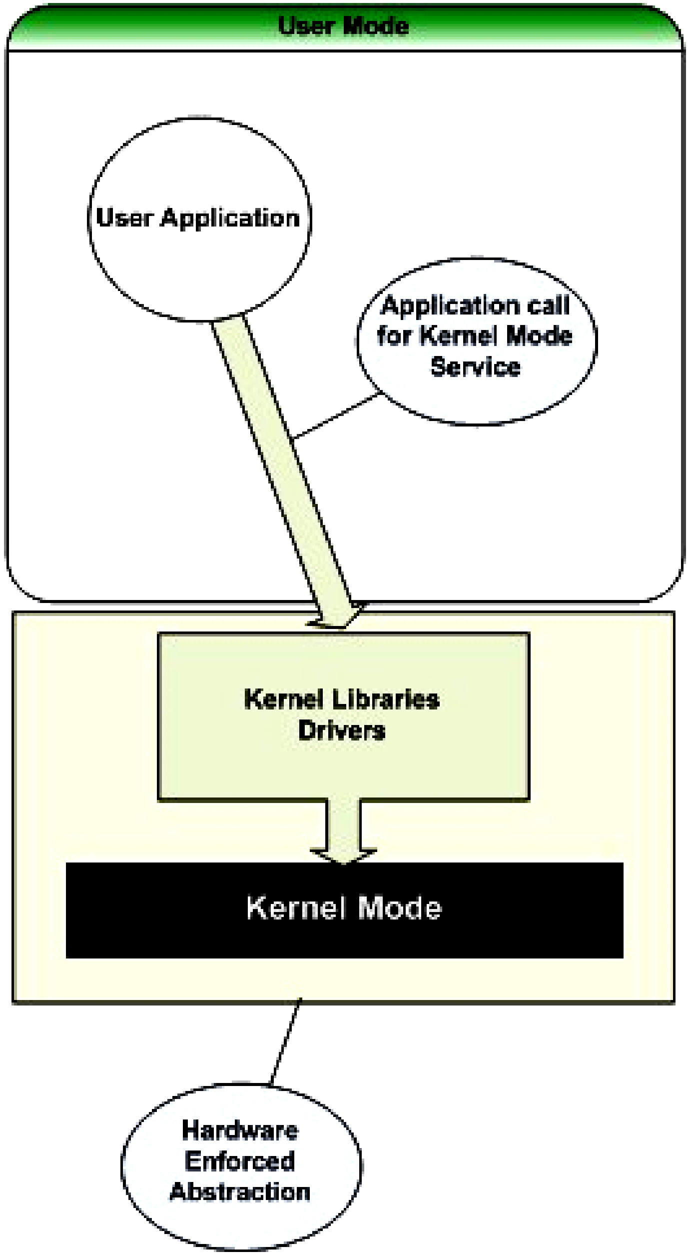
- Operating System Definition
- Computer System Components
- Operating System Operation
  - Fetch-Execute Cycle
  - Interrupts
  - I/O Operation
    - Programmed Driven I/O
    - Interrupted Driven I/O
  - Modes of Operation
    - user mode
    - kernel mode
  - Multiprogramming
  - Time sharing

Operating System Operation

- **Modes of Operation**
- most computer systems provide hardware support that allows to differentiate among modes of execution:
  - **user mode**:
    - the executing code has no ability to directly access hardware or reference memory. It must delegate to system APIs to access hardware or memory
    - crashes are always recoverable.
    - Most of the code running on a PC will execute in user mode
  - **kernel mode** (aka supervisor, system, or privileged mode):
    - the executing code has complete and unrestricted access to the underlying hardware. It can execute any CPU instruction and reference any memory address.
    - generally reserved for the lowest-level, most trusted functions of the operating system.
    - crashes are catastrophic; they will halt the entire PC

Operating System Operation

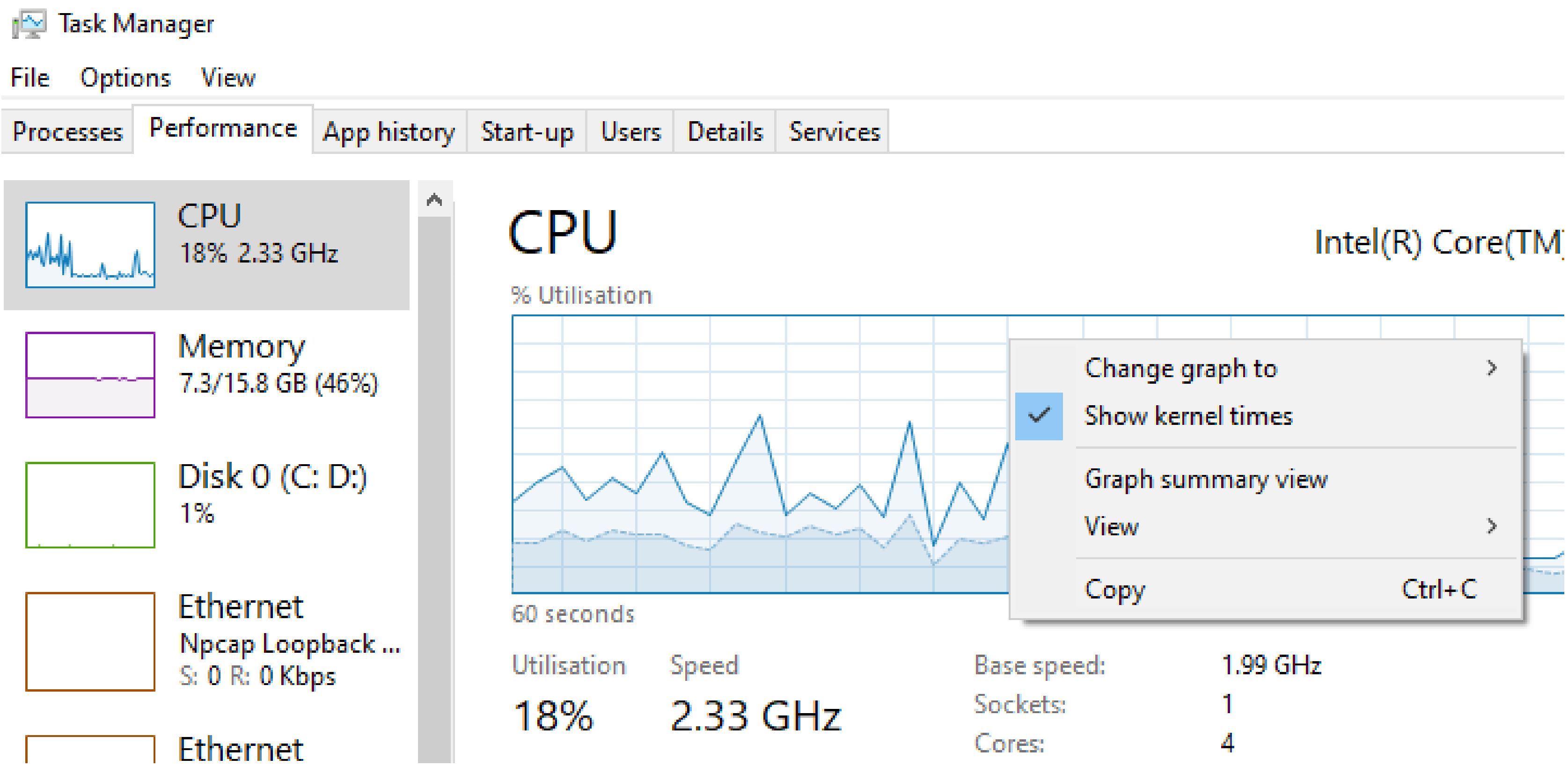
- **Modes of Operation**



Operating System Operation

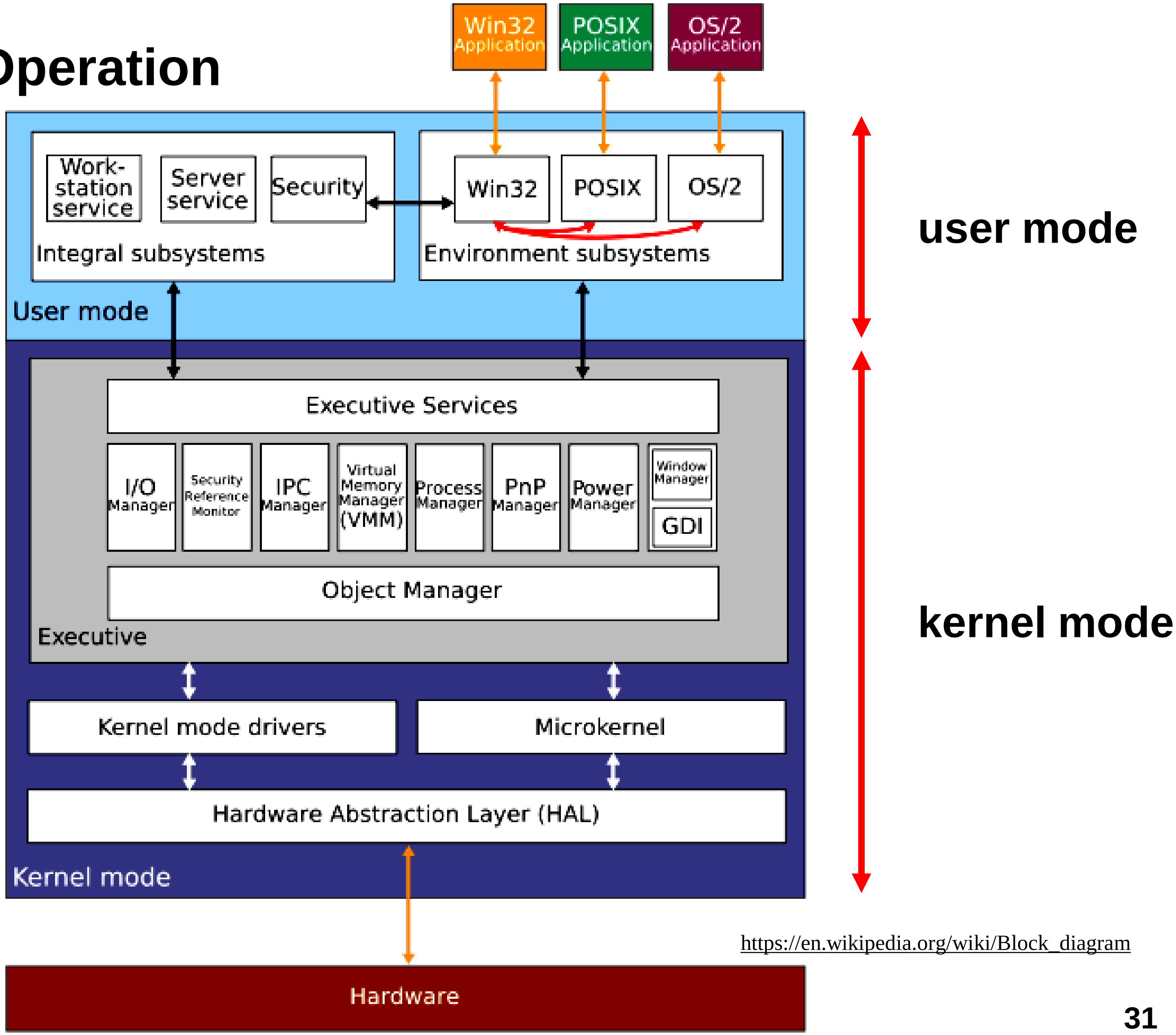
- **Modes of Operation**
- A bit, called the **mode bit** is added to the hardware of the computer to indicate the current mode: **kernel (0) or user (1).**
  - At system boot time, the hardware starts in kernel mode.
  - The operating system is then loaded and starts user applications in user mode:
    - **whenever a trap or interrupt occurs, the hardware switches from user mode to kernel mode (i.e. changes the state of the mode bit to 0).**
    - the system always switches to user mode (by setting the mode bit to 1) before passing control to a user program.
- The CPU's strict segregation of code between User and Kernel mode is completely transparent to most of us.

- **Modes of Operation**
- Example: Windows 10



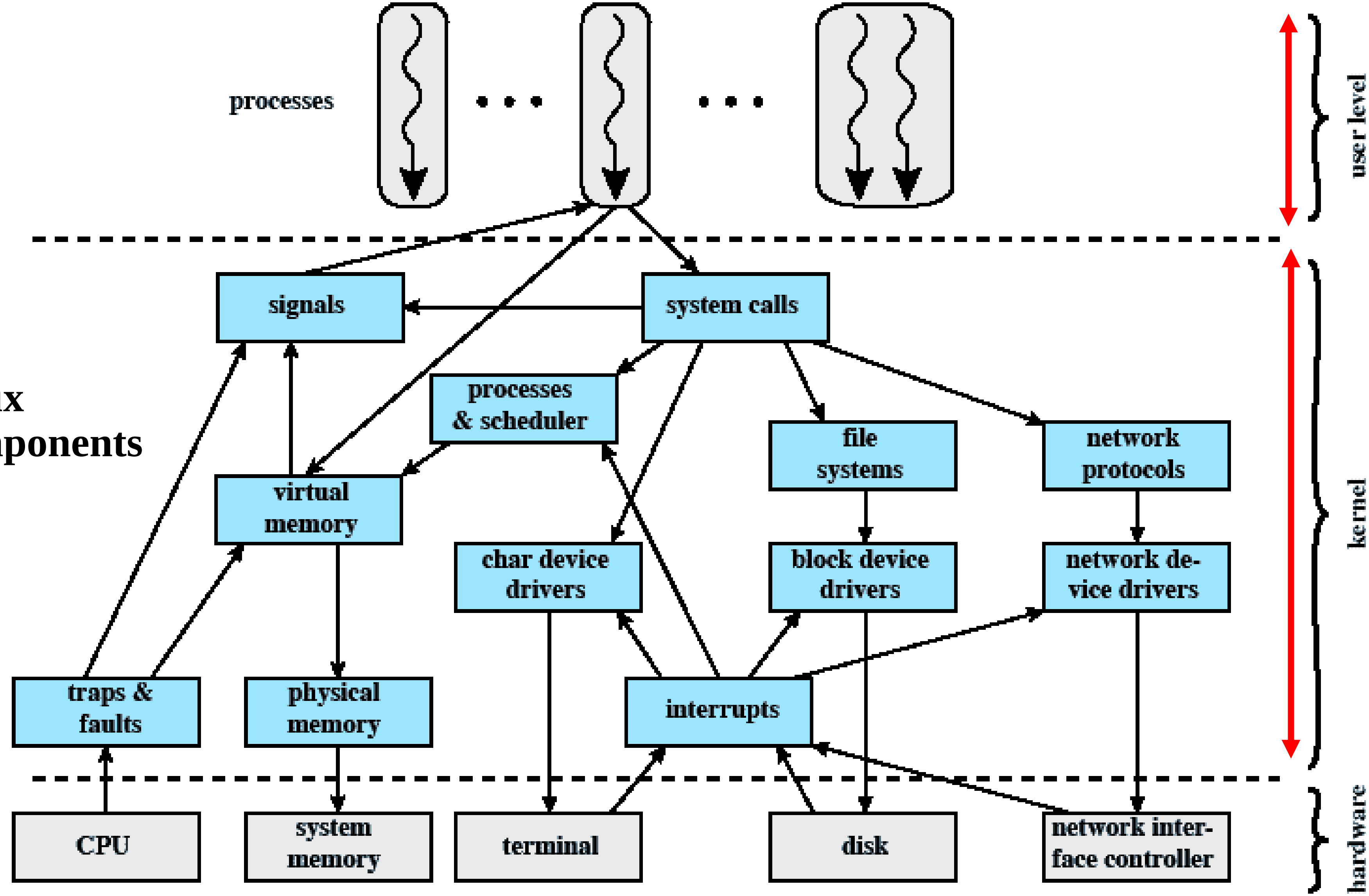
Operating System Operation

- **Modes of Operation**



https://en.wikipedia.org/wiki/Block_diagram

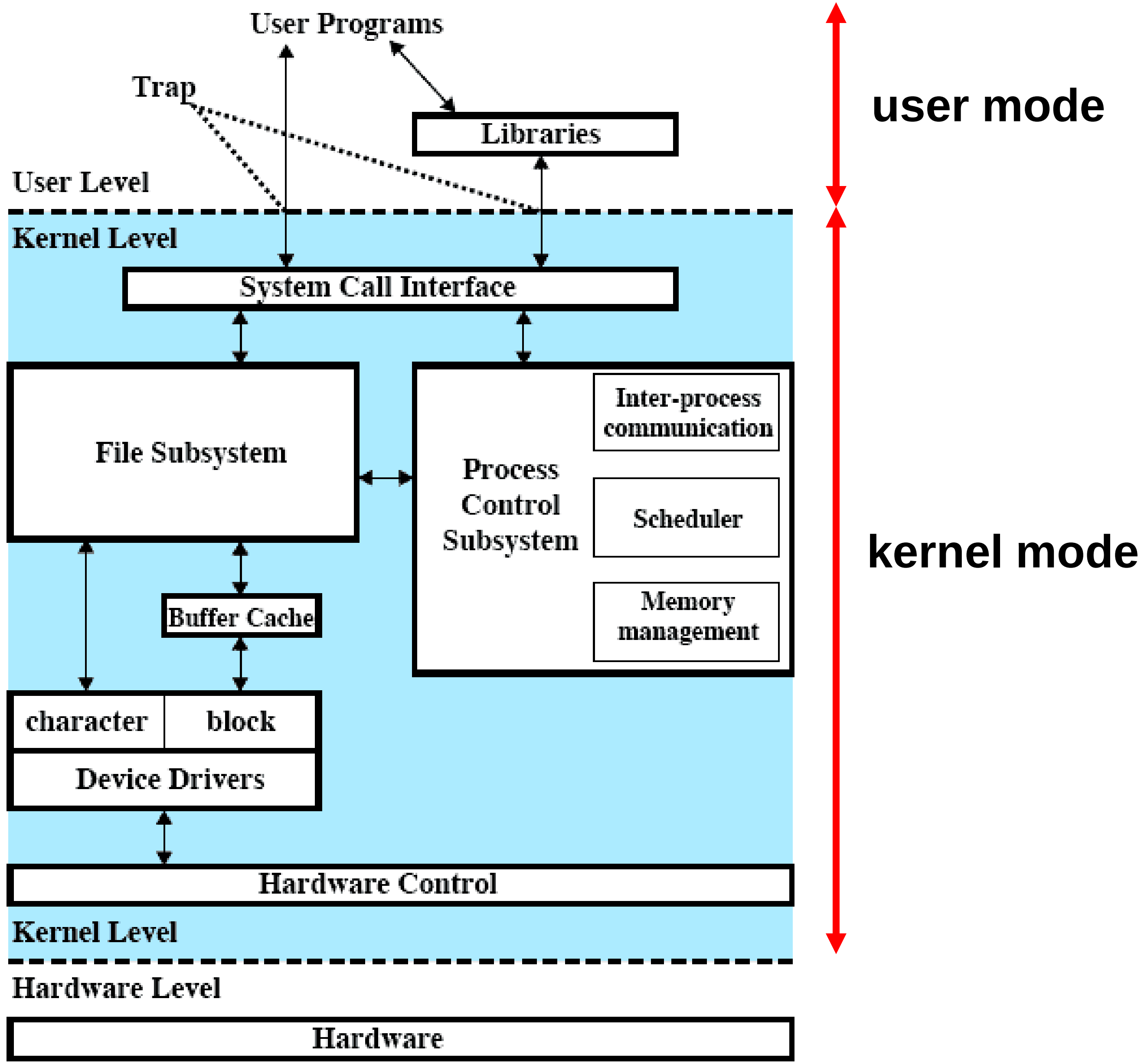
Linux Components



William Stallings, “Operating Systems: Internals and Design Principles”

Operating System Operation

Unix Components



William Stallings, "Operating Systems: Internals and Design Principles"

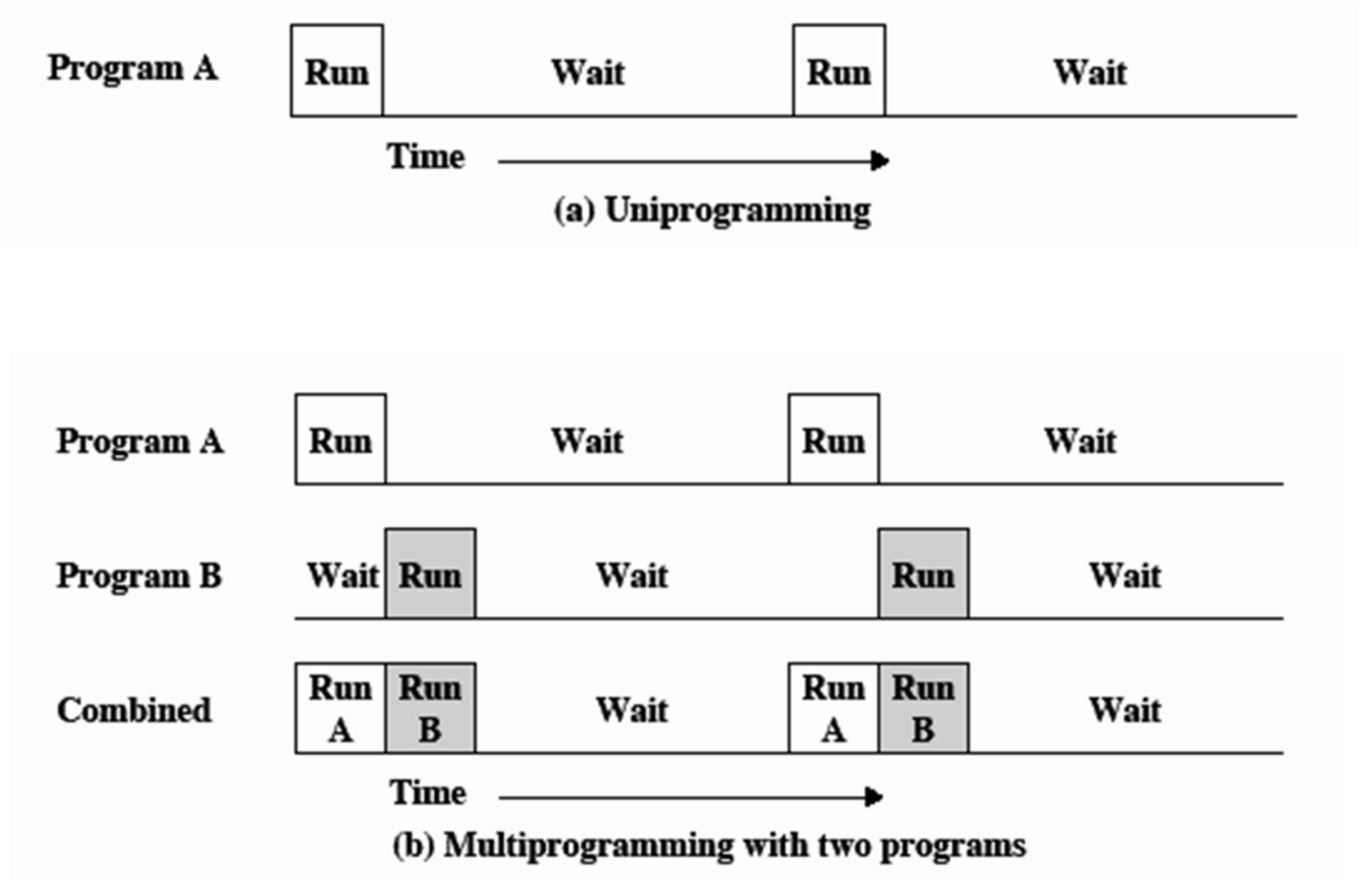
ESTGOH
Escola Superior de Tecnologia e Gestão de Oliveira do Hospital
Instituto Politécnico de Coimbra

Outline

- Operating System Definition
- Computer System Components
- Operating System Operation
  - Fetch-Execute Cycle
  - Interrupts
  - I/O Operation
    - Programmed Driven I/O
    - Interrupted Driven I/O
  - Modes of Operation
    - user mode
    - kernel mode
  - Multiprogramming
  - Time sharing

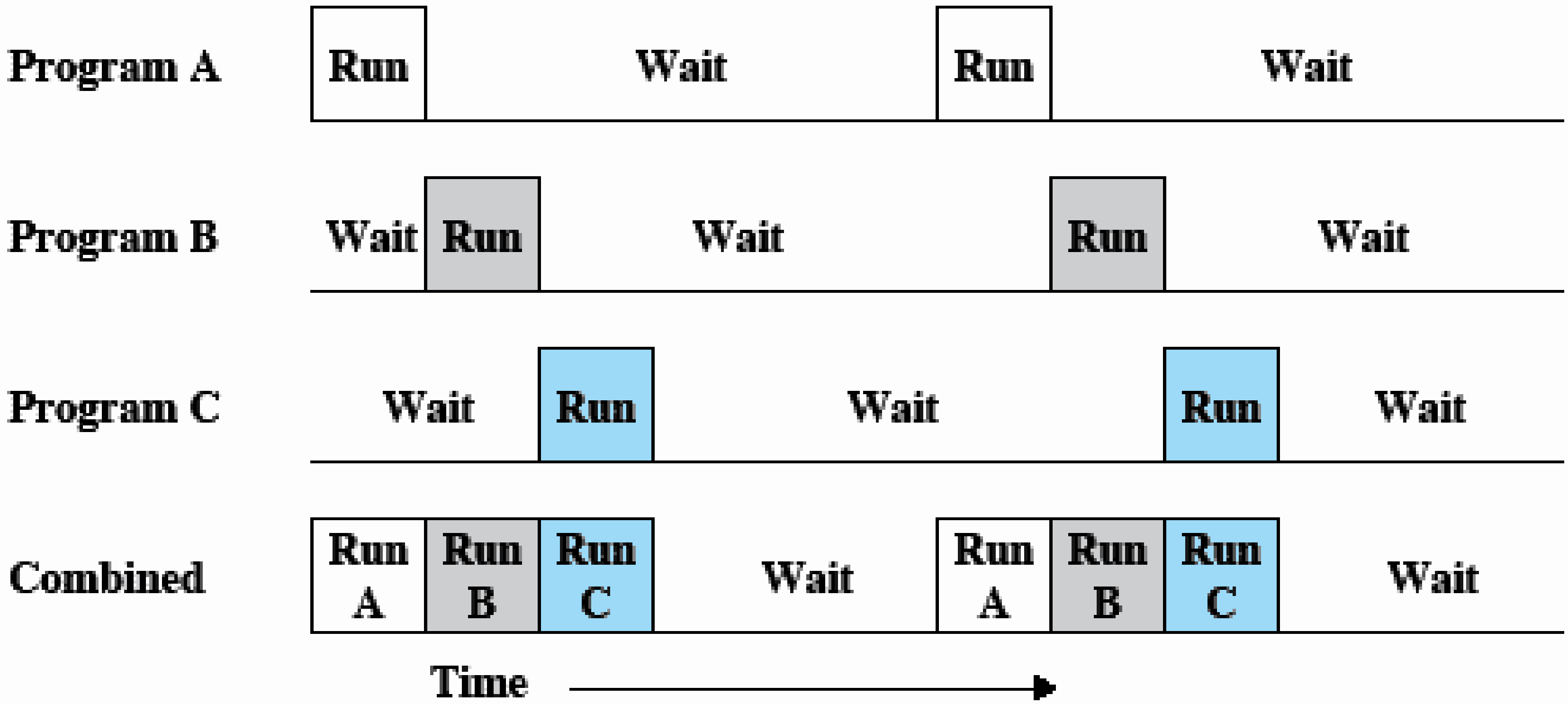
Operating System Operation

- **Multiprogramming**
  - increases CPU utilization by organizing processes in a way that the CPU always has one to execute:



William Stallings, "Operating Systems: Internals and Design Principles"

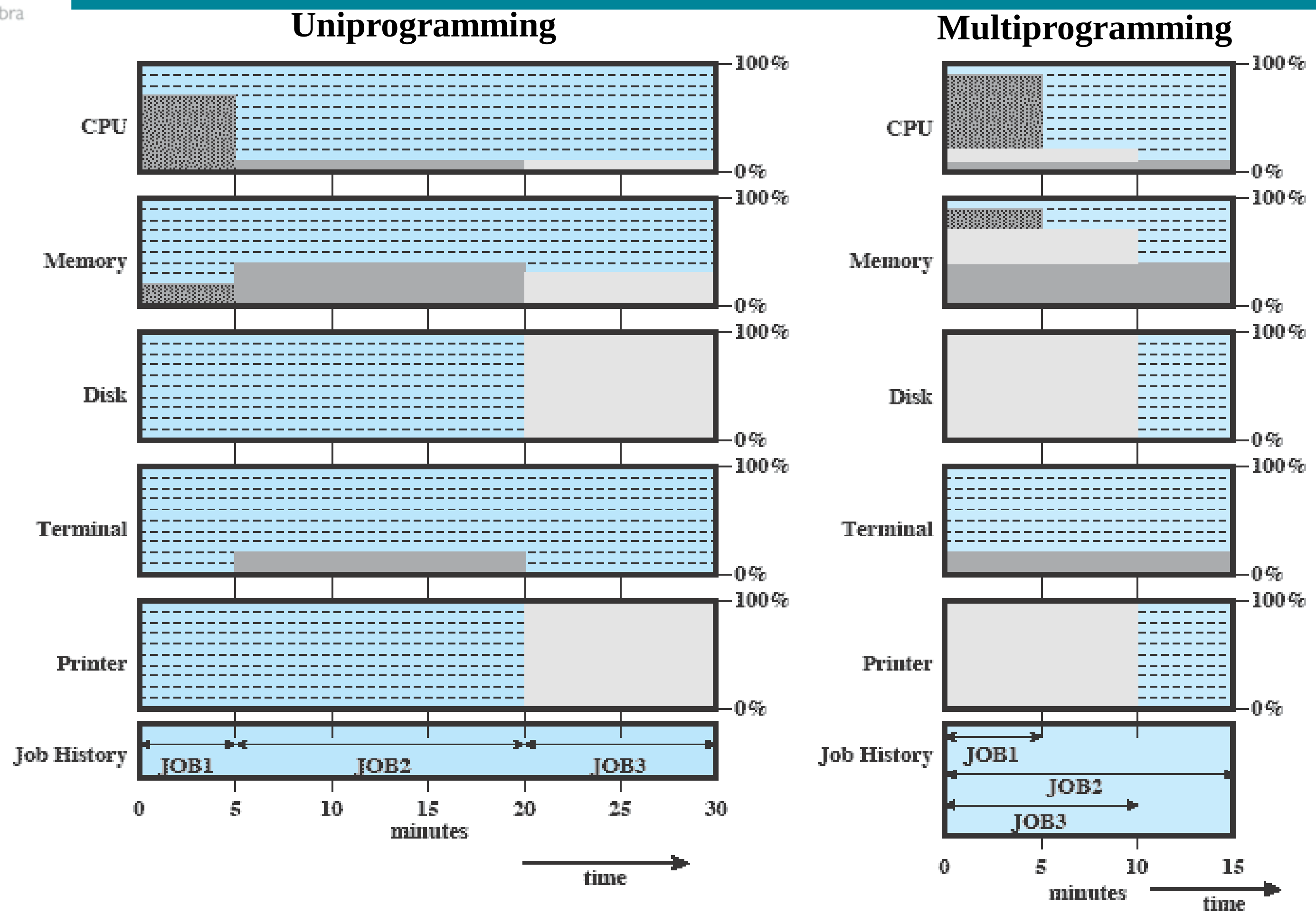
- **Multiprogramming**



(c) Multiprogramming with three programs

William Stallings, "Operating Systems: Internals and Design Principles"

Operating System Operation



	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min	15 min	10 min
Memory required	50 M	100 M	75 M
Need disk?	No	No	Yes
Need terminal?	No	Yes	No
Need printer?	No	No	Yes

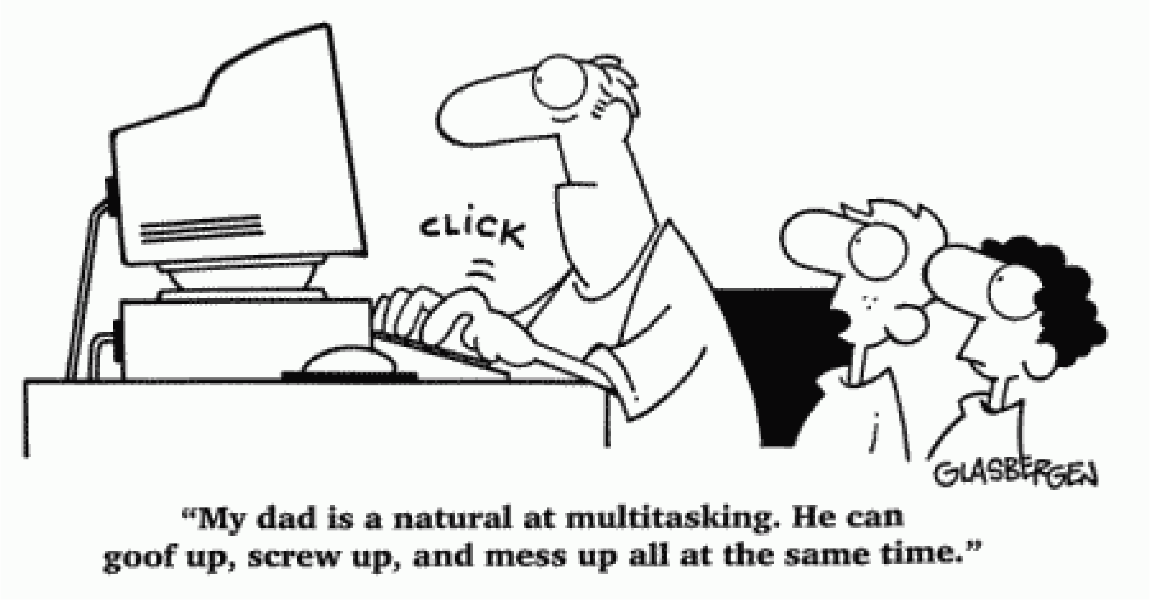
William Stallings, “Operating Systems: Internals and Design Principles”

Outline

- Operating System Definition
- Computer System Components
- Operating System Operation
    - Fetch-Execute Cycle
    - Interrupts
    - I/O Operation
        - Programmed Driven I/O
        - Interrupted Driven I/O
    - Modes of Operation
        - user mode
        - kernel mode
    - Multiprogramming
    - Time sharing

Operating System Operation

- **Time sharing (or multitasking)**
  - the CPU executes multiple processes from different users by switching among them so rapidly that users can interact with each program while it is running.
  - allows many users to share the computer simultaneously.
  - each user is given the impression that the entire computer system is dedicated to his use, even though it is being shared among many users.



https://www.glasbergen.com/gallery-search/?tag=multitask

References

- Abraham Silberschatz, ” Operating System Concepts”, 10th Edition, Wiley, 2018;
- William Stallings, “Operating Systems: Internals and Design Principles”, 9th Edition, Pearson, 2017;
- Andrew S. Tanenbaum, “Modern Operating Systems”, 4th Edition, Pearson Education, 2014;