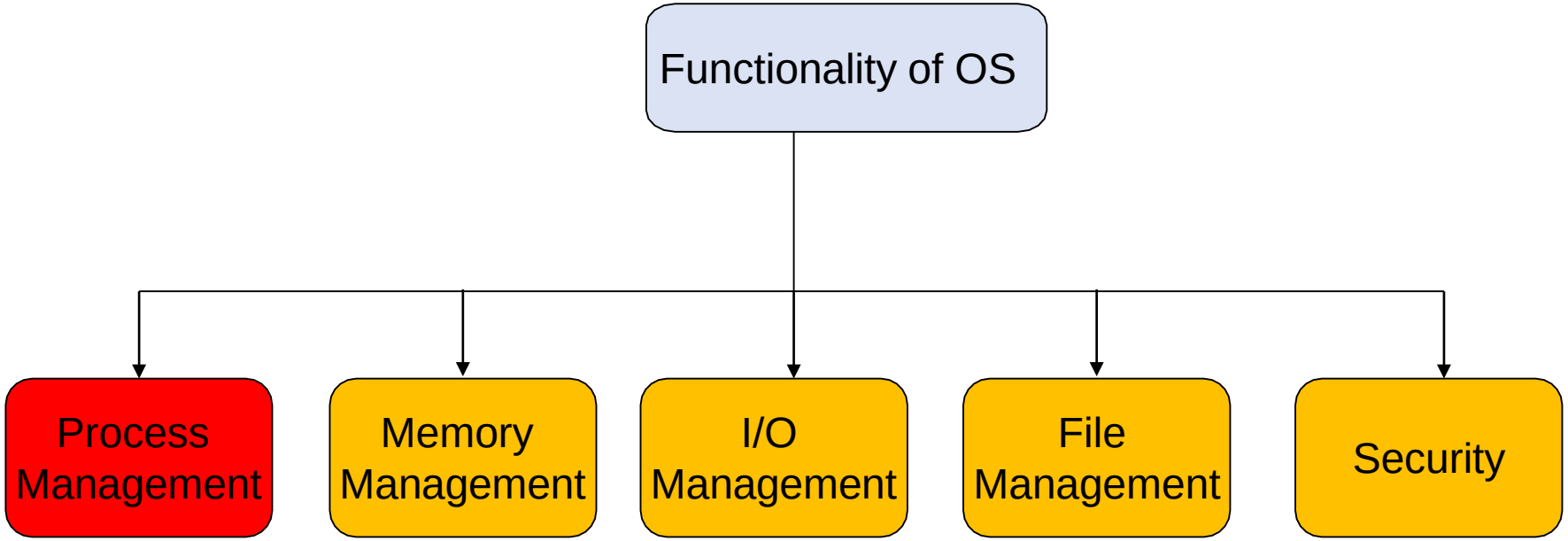
OPERATING SYSTEM: CSET209

FUNCTIONALITY OF OS



CONCEPT OF PROCESS MANAGEMENT

Contents to be Discussed:

1. Process management
   - Process and its state diagram
   - Process Control Block
2. Context switching
3. Schedular and dispatcher
4. System calls and interrupts

1. PROCESS MANAGEMENT

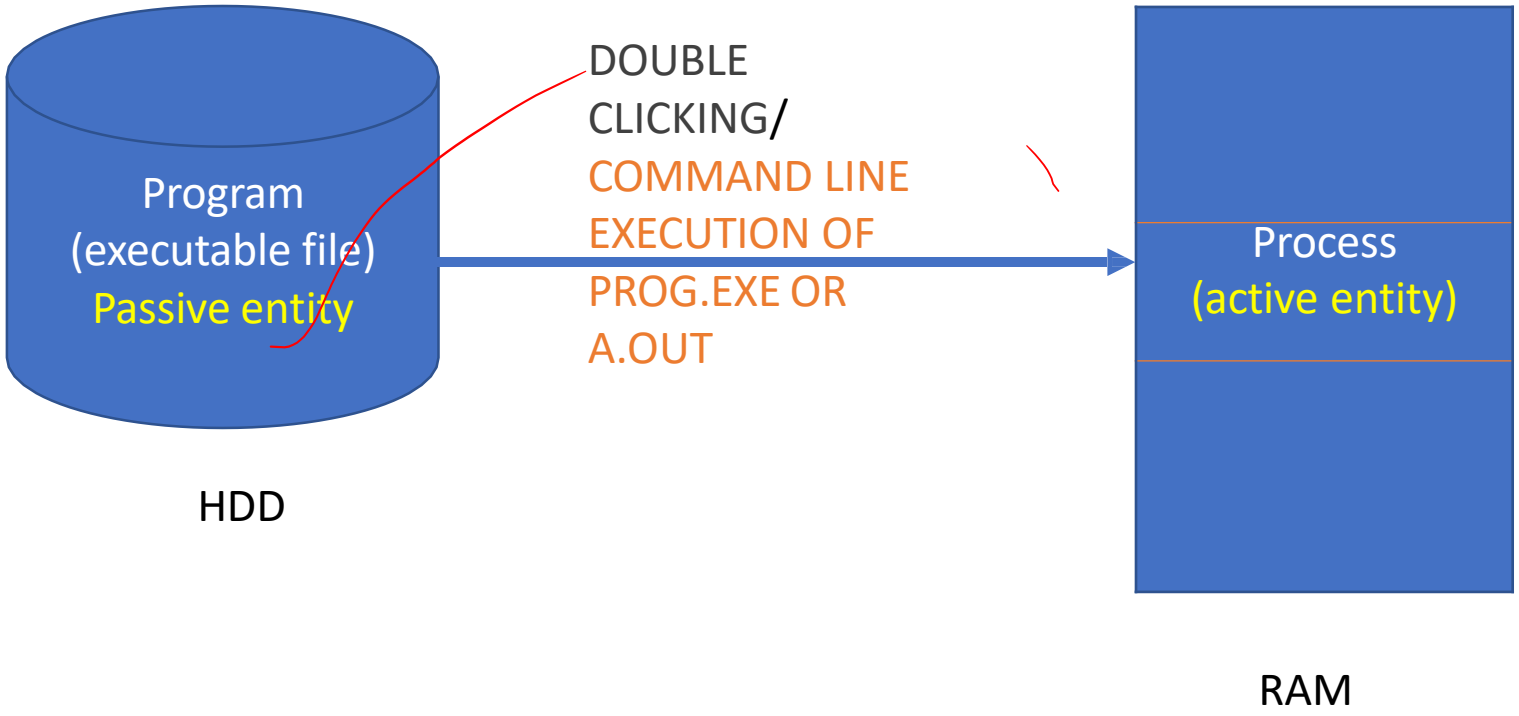
- A Program does nothing unless its instructions are executed by a CPU. A program in execution is called a process.

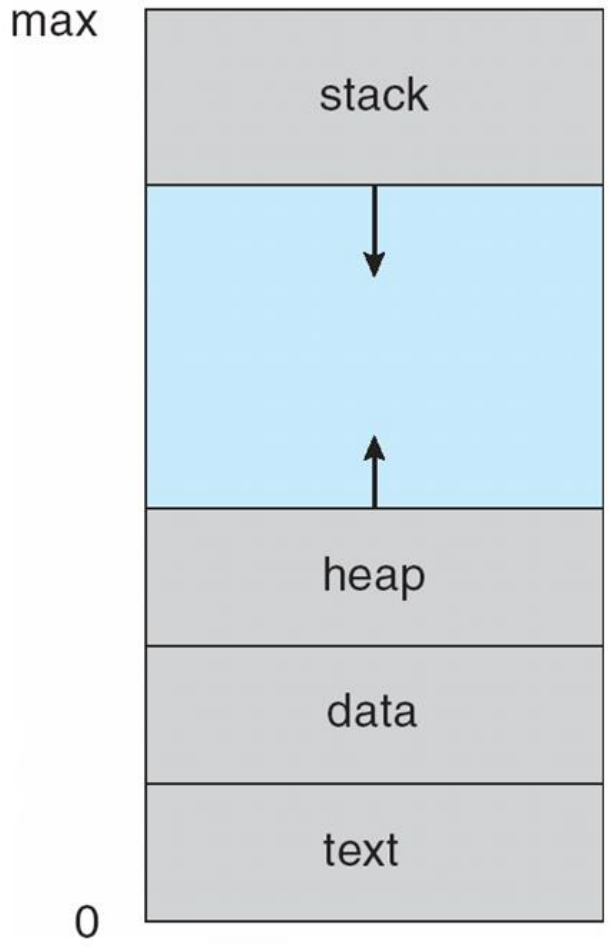
- There may exist more than one process in the system which may require the same resource at the same time. Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.

The operating system is responsible for the following activities in connection with Process Management:

✓ Scheduling processes and threads on the CPUs.
✓ Creating and deleting both user and system processes.
✓ Suspending and resuming processes.
✓ Providing mechanisms for process synchronization.
✓ Providing mechanisms for process communication.

1. CONCEPT OF PROCESS





Process in memory

Process: Program under execution called process

2. ATTRIBUTES AND PCB OF A PROCESS

The Attributes of the process are used by the Operating System to create the process control block (PCB) for each of them. This is also called context of the process. Attributes which are stored in the PCB are described below:

1. Process ID: When a process is created, a unique id is assigned to the process which is used for unique identification of the process in the system.

2. **Program counter**: A program counter stores the address of the last instruction of the process on which the process was suspended. The CPU uses this address when the execution of this process is resumed.

3. Process State: The Process, from its creation to the completion, goes through various states which are new, ready, running and waiting.

4. **Priority**: Every process has its own priority. The process with the highest priority among the processes gets the CPU first. This is also stored on the process control block.

ATTRIBUTES AND PCB OF A PROCESS

5. **General Purpose Registers**: Every process has its own set of registers which are used to hold the data which is generated during the execution of the process.

6. **List of open files**: During the Execution, Every process uses some files which need to be present in the main memory. OS also maintains a list of open files in the PCB.

7. **List of open devices**: OS also maintain the list of all open devices which are used during the execution of the process.

ATTRIBUTES AND PCB OF A PROCESS

Process Control Block or Task Control Block or Process Descriptor: PCB serves as repository that contains attributes of process. Attributes may include process state, process number, program counter, list of open files, registers, process priority etc. Each process in the system is represented by a PCB

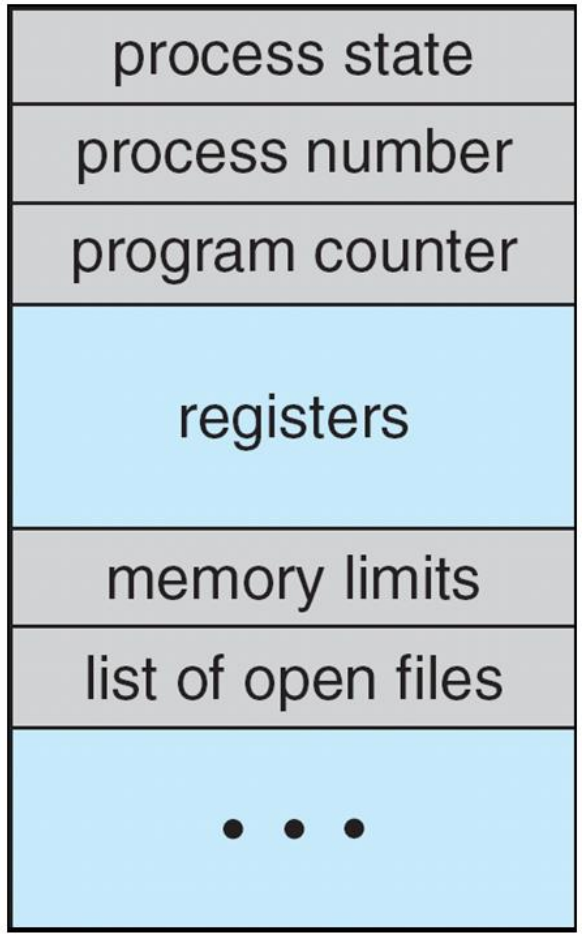


Figure- Process control block (PCB)

2. CPU SWITCH FROM PROCESS TO PROCESS

- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process.
- Context-switch time is overhead; the system does no useful work while switching.
- Time dependent on hardware support.

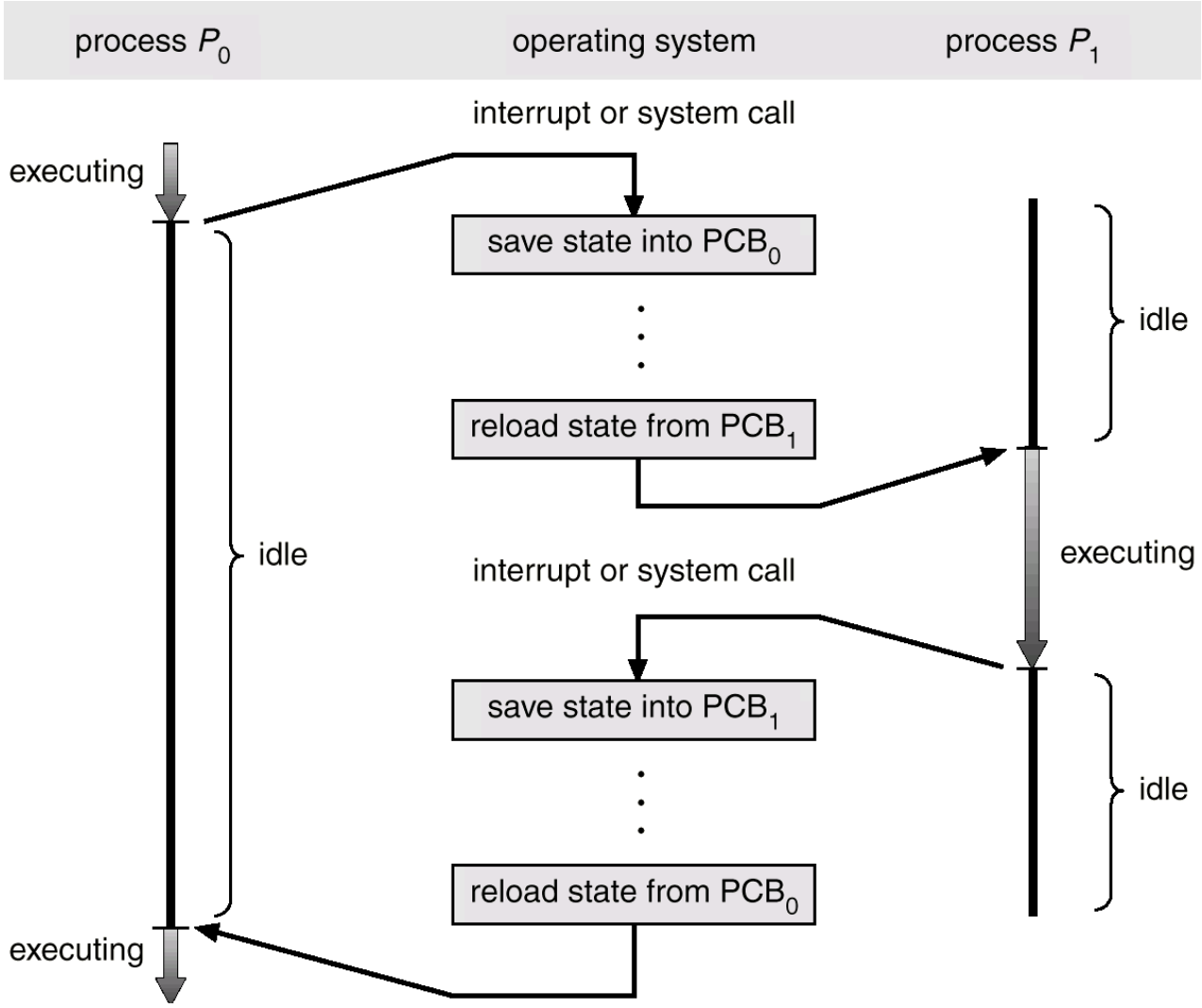
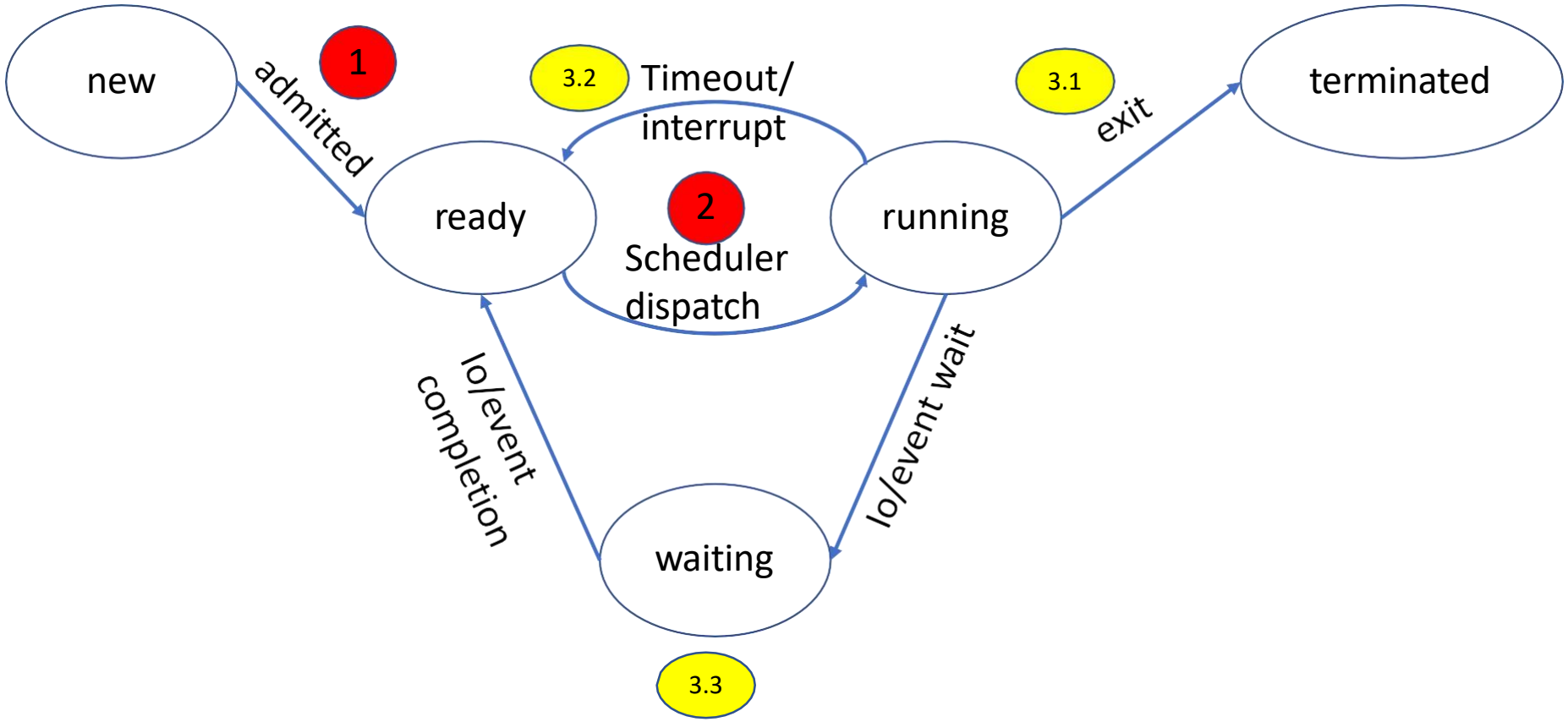


Fig: Diagram showing CPU switch from process to process

1. PROCESS STATE LIFE CYCLE MODEL

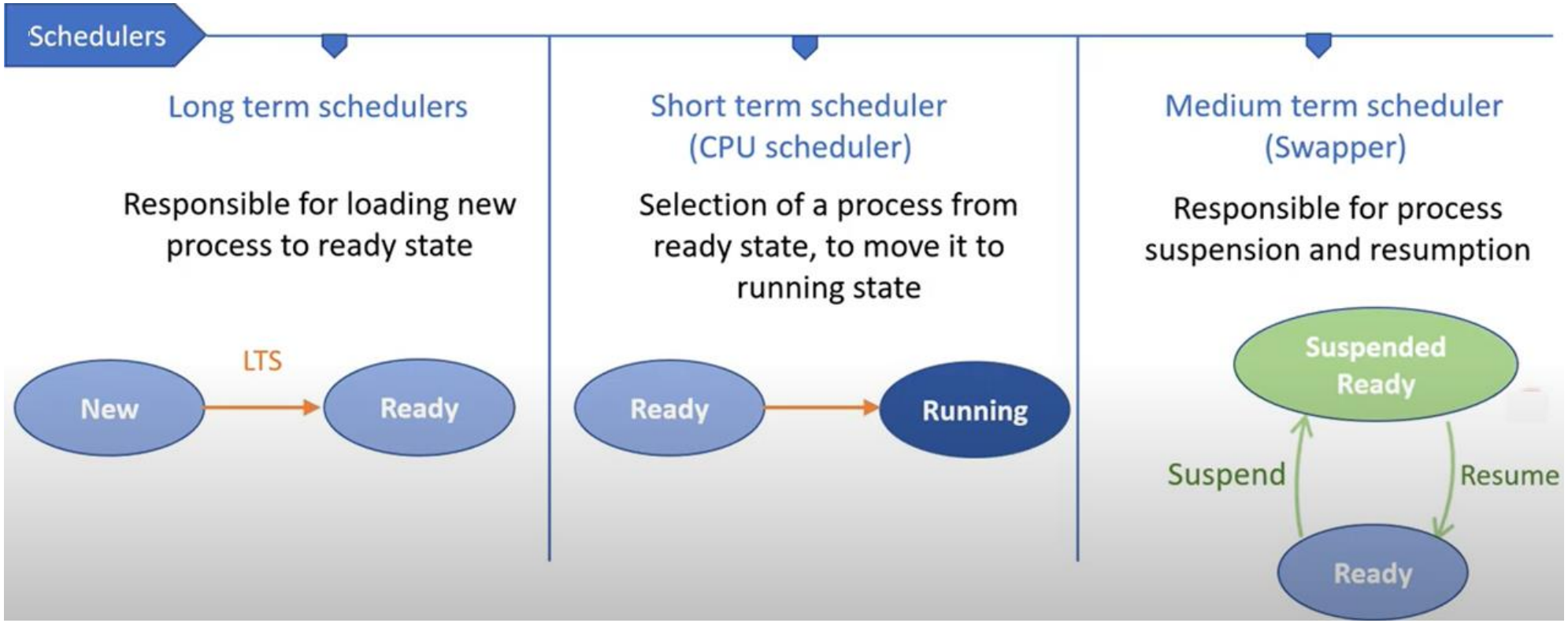
- As a process executes, it changes *state*
  - new: The process is being created.
  - ready: The process is waiting to be assigned to a processor.
  - running: Instructions are being executed.
  - waiting: The process is waiting for some event (I/O operations) to occur.
  - terminated: The process has finished execution.

1. PROCESS STATE LIFE CYCLE MODEL



3. PROCESS SCHEDULING

To meet the objectives of Multiprogramming, Schedulers are responsible for selecting a process for scheduling. **Three types of scheduler are:**



QUIZ QUESTION !

The Degree of multiprogramming is controlled by which one?

a. Short term scheduler
b. Long term scheduler.
c. Medium term scheduler.

QUIZ QUESTION 2

Because of the short time between executions, the short-term scheduler must be fast. If it takes 20 milliseconds to decide to execute a process of 100 milliseconds, then ? percent of the CPU is being used (wasted) simply for scheduling the work?

CPU BOUND AND I/O BOUND PROCESS

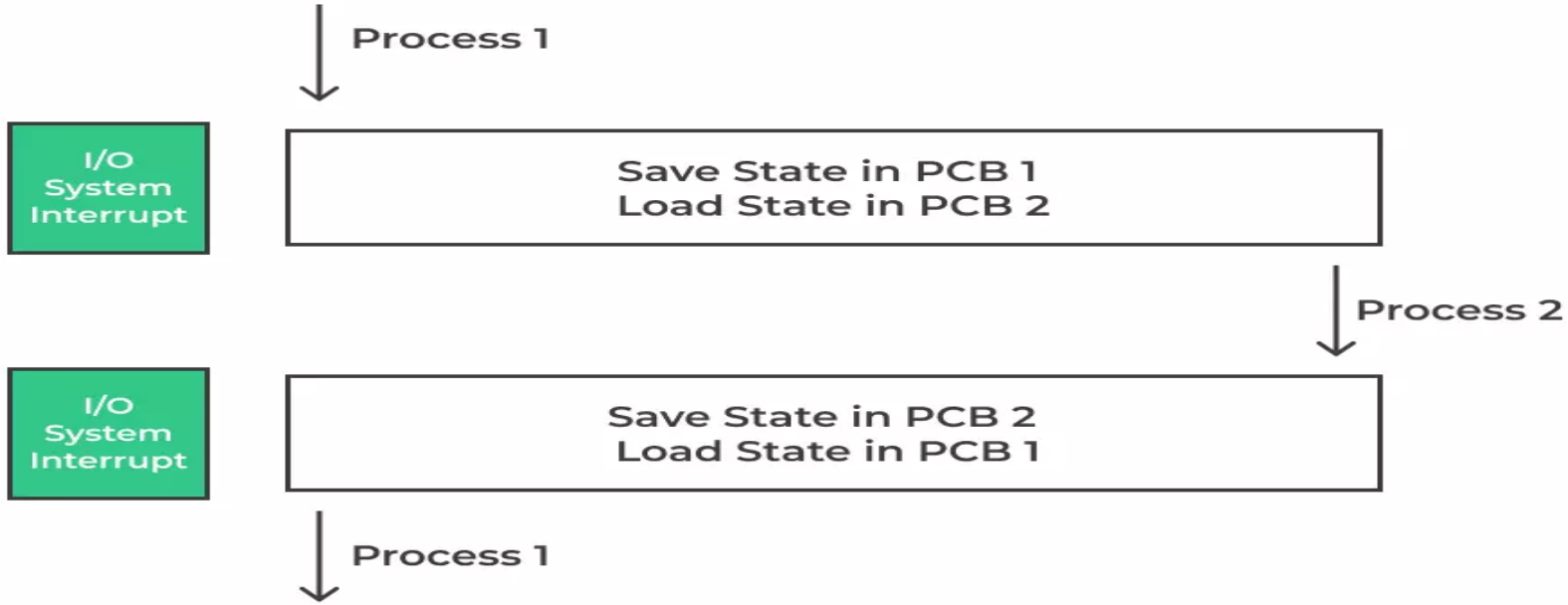
- **A CPU-bound process** requires more CPU time or spends more time in the running state.
-
- **An I/O-bound process** requires more I/O time and less CPU time. An I/O-bound process spends more time in the waiting state.

DISPATCHER

- It is the module that gives control of the CPU to the process selected by the short-time scheduler.

- A dispatcher switches execution from one process to another process called context switching. It also setup user registers, memory mapping, etc.

- Dispatch latency: amount of time taken by the system to stop one process and give permission to another process to being execution.

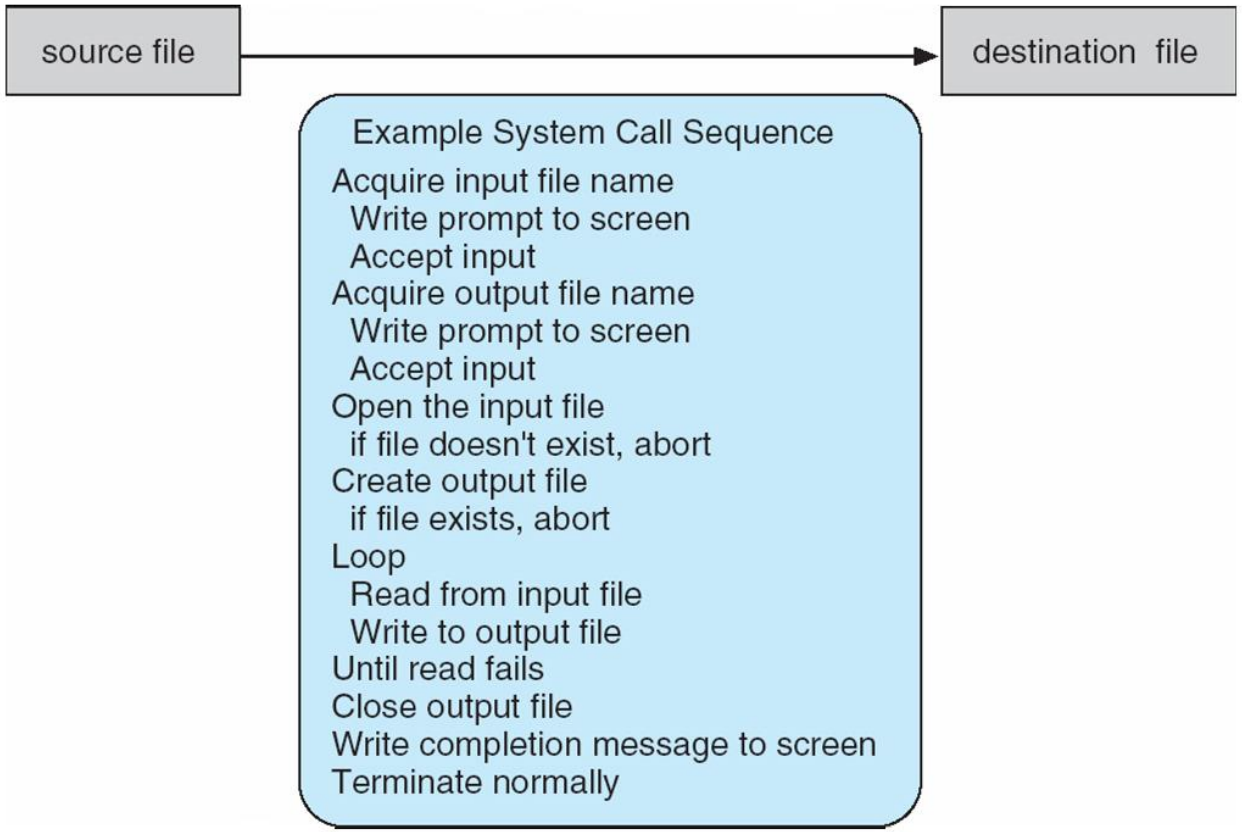


4.1 SYSTEM CALLS IN OS

- A system call is a method for a computer program to request a service from the kernel of the operating system on which it is running. It is the only method to access the kernel system. All programs or processes that require resources for execution must use system calls.
- Even simple programs may make heavy use of the operating system. Frequently, systems execute thousands of system calls per second.
- Mostly accessed by programs via a high-level Application Programming Interface (API) rather than direct system call use.

AN EXAMPLE OF SYSTEM CALL USAGE

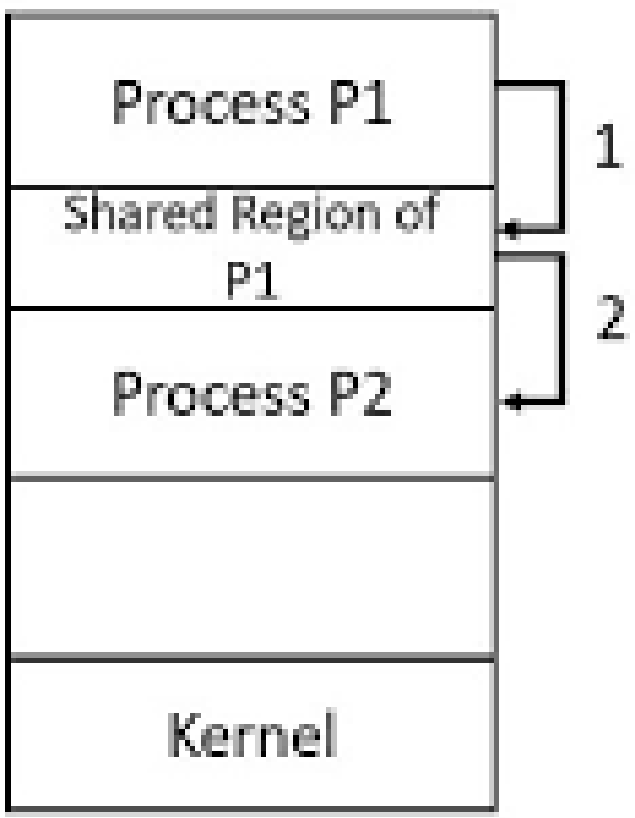
- System call sequence to copy the contents of one file to another file



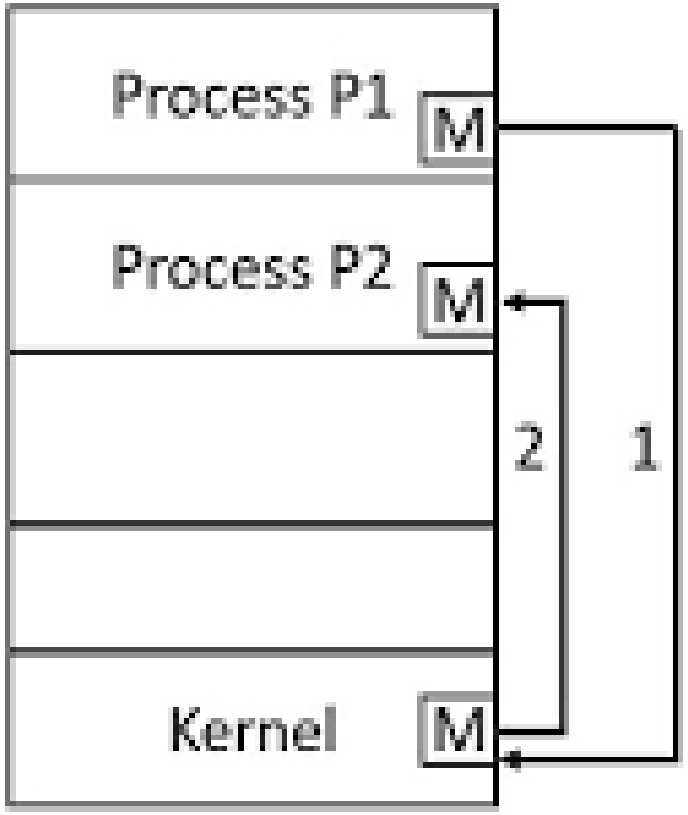
TYPES OF SYSTEM CALLS

Types of System Calls	Windows	Linux
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Management	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()

COMMUNICATION MODELS



Shared Memory System



Message Passing System

4.2. WHAT IS INTERRUPT IN OS?

- An interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high-priority process requiring interruption of the current working process. In I/O devices, one of the bus control lines is dedicated for this purpose and is called the ***Interrupt Service Routine*** (ISR0).
- When a device raises an interrupt at the process, the processor first completes the execution of an instruction. Then it loads the ***Program Counter*** (PC) with the address of the first instruction of the ISR. Before loading the program counter with the address, the address of the interrupted instruction is moved to a temporary location. Therefore, after handling the interrupt, the processor can continue with the process.

TYPES OF INTERRUPT

1. Hardware Interrupts

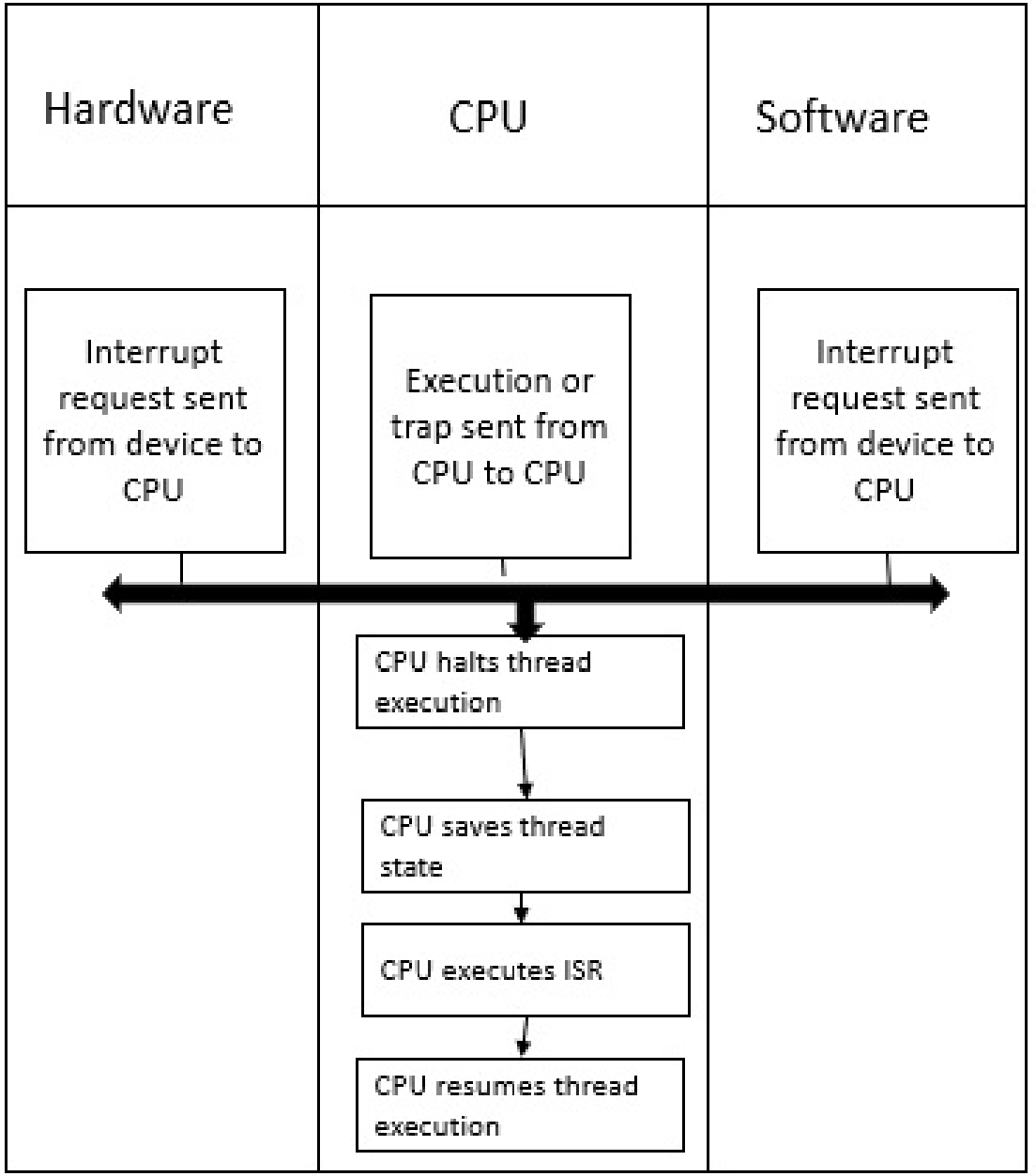
- A hardware interrupt is a condition related to the state of the hardware that may be signaled by an external hardware device, e.g., an interrupt request (IRQ) line on a PC, or detected by devices embedded in processor logic to communicate that the device needs attention from the operating system. For example, pressing a keyboard key or moving a mouse triggers hardware interrupts that cause the processor to read the keystroke or mouse position.

2. Software Interrupts

- The processor requests a software interrupt upon executing particular instructions or when certain conditions are met. Software interrupts may also be unexpectedly triggered by program execution errors. These interrupts are typically called traps or exceptions.

HOW INTERRUPT HANDLING IS DONE ?

- Whenever an interruption occurs the processor finishes the current instruction execution and starts the execution of the interrupt known as interrupt handling.
- The interrupt handling mechanism of an operating system accepts a number which is an address and then selects what specific action to be taken which is already mentioned in the interrupt service routine. In most architecture, the address is stored in a table known as a vector table. Interrupt handling by OS scheme is shown as follows:



THANK YOU
?