UNIT-I

- Introduction
- Functions of Operating System
- Components of O.S.
- Characteristics of O.S.
- Services of O.S.
- Process Concept
- Definition of Process
- Process States
- Process State Transition
- Operations on Process
- Inter process Communication
- Process Scheduling

Operating System

- It is a program that acts as interface between user of computer and computer hardware.
- The purpose of OS is to provide an environment in which user can execute programs in convenient and efficient manner.
- The OS must ensure that correct operation are performed on the computer system.
- The OS provides certain services to program and to the user to make programming task easier.

Functions of Operating System

- Easy interaction between human and computer.
- Starting computer operation automatically when power turns ON.
- Loading and scheduling user programs along with necessary compiler.
- Controlling I/O operations.
- It schedules the processes.
- Manages use of main memory.
- Manage and manipulate files.
- Provide security to the user jobs and files.
- Manage a resource usage like software resource and hardware resource.

Characteristics of OS

- Efficiency
- Robustness
- Scalability
- Extensibility
- Portability
- Security
- Interactivity
- Usability

Process

- Process is nothing but program in execution. Each process has its own address space which typically consist of text region, data region and stack region.
- The text region stores the code that the processor executes.
- The data region stores variable and dynamically allocated memory that the process uses during execution.
- The stack region stores instructions and local variables for active procedure calls.

Process State

- The OS must ensure that each process receives a sufficient amount of process time.
- There are many processed than processors in the system. Thus at any given time, some processes can execute and some cannot.
- During its lifetime, a process moves through series of discrete process states.
- Various events can cause a process to change state i.e.
  1. Executing a process ----------☐ Running State
  2. It could Execute on processor----------☐ Ready State
  3. Waiting for some event to occur----------☐ Block State

- In uniprocessor system, only one process may be running at a time but several maybe ready or blocked.
- The OS maintains a ready list of ready processes and block list of blocked processes.
- The ready list is maintained in priority order, so that the next process to receive a process is the first one in the list (i.e. the process with the highest priority).
- When a user runs a program ,processes are created and inserted into ready list .A process moves towards the head of the list as other processes complete their turn using a processor.

Process State Transition

- Change of process from one state to another.
- The OS manages state transition to best serve processes in system. To prevent any process from monopolizing the system, either accidently or maliciously, OS set hardware interrupting clock to allow a process to run specific interval.
- Four possible state transitions are :
  1. **Ready to Running ----------□Dispatch**
  2. **Running to Ready----------□Process's Quantum Expires**
  3. **Running to Block----------□Process Block**
  4. **Block to Ready ----------□Process Wakeup**

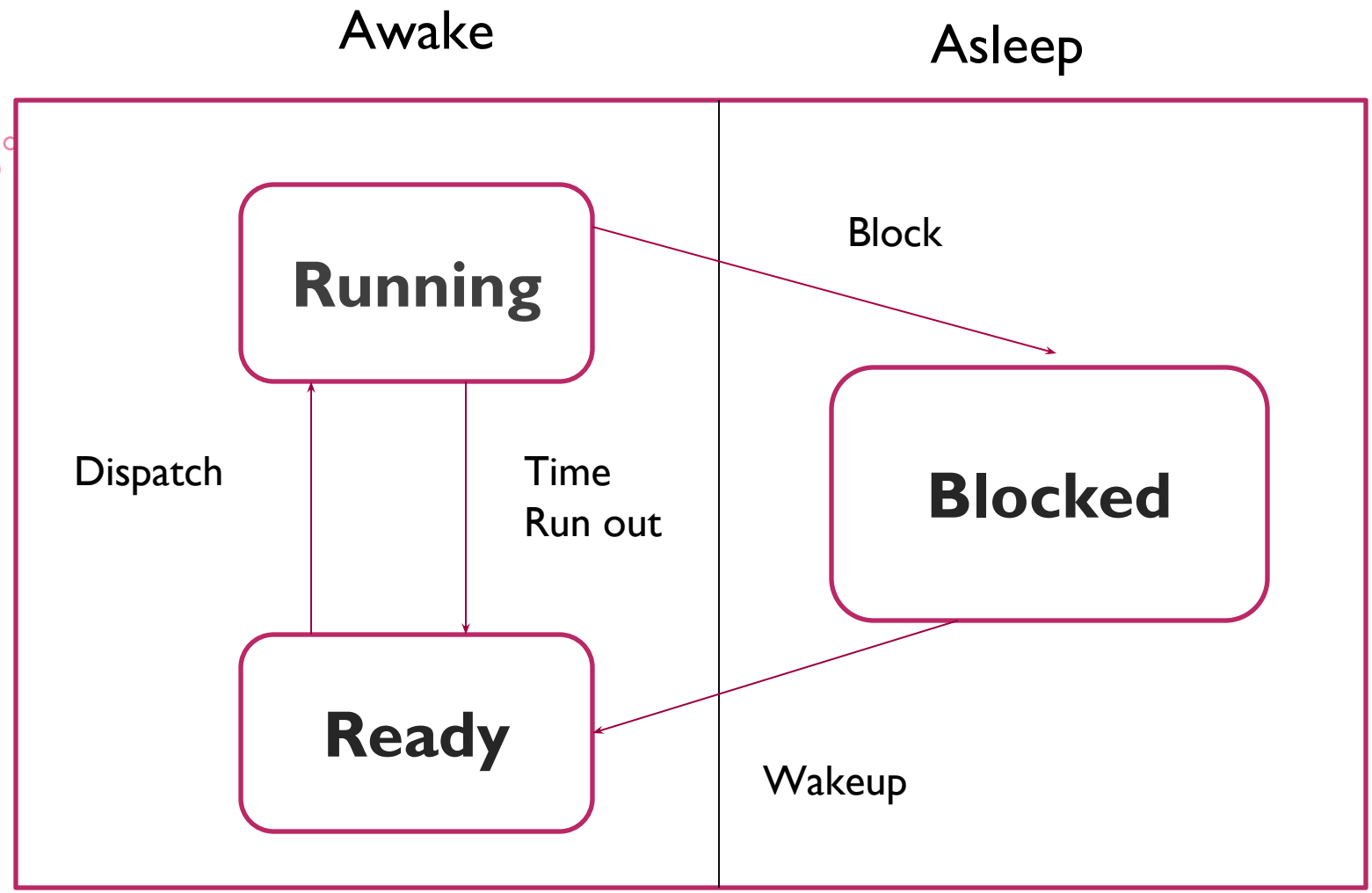


Fig. Process State Transition

Information Management

- It provides the facility to store,retrieve,modify or remove the information from files/directories.
- These system services manages the organization of information into files and directories by allocating memory space to them.
- It also ensure that co-programs have access to information and occupied memory space; driving various services.

Some of system services (System call) of IM

- Create a file
- Create a directory
- Open a file for read/write purpose
- Explore a directory
- Close a file
- Read data from file to buffer
- Write data from file to buffer
- Move file pointer
- Create a Link
- Change working directory

Process Management

- **It is used to keep track of all running programs which are called processes**
- **In multiuser O.S., no. of users located at different terminals of network, may execute same or different program at a time.**
- **The PM of such O.S. keeps track of all running programs called processes, schedule them and dispatch them one after another while doing so, it gives an impression to each user that it has the full control of CPU.**
- **The PM module of single O.S. are less compatible than multiuser O.S.**

Some of System Services (System Calls) of PM

- Running a process
- Ready a process
- Block a process
- Terminate a process
- Suspend a process
- Delay a process
- Change the priority of process
- Generate a process
- Dispatch a process
- Fork a process

Memory Management

- It keeps track of all memory locations, determine memory location policy and uses various techniques & algorithm for it.
- When job is executed the OS loads the job in main memory from disk.
- But before loading it in memory, it should know that how much of free memory is available and how much of memory should be allocated to the job.
- For this, the OS keeps list of all free memory locations. Before a program is loaded in memory the OS consults with this list.
- It loads the program into memory and modified the list.

- When the program is executed completely, it removes program into memory and modified the list.
- For this , Os determines memory allocation/deallocation policies and uses various technique and algorithm for it.
- The system call in it are :
  1. To allocate a chunk of memory to process.
  2. To free chunk of memory from a process.

Components of OS

- The following are components of operating system

1. Graphical User Interface (GUI)
2. Kernel
3. Interrupt
4. Memory Management

5. **Graphical User Interface(GUI):**

   It allows the user to point and click, drag and drop icons to request services from the OS.

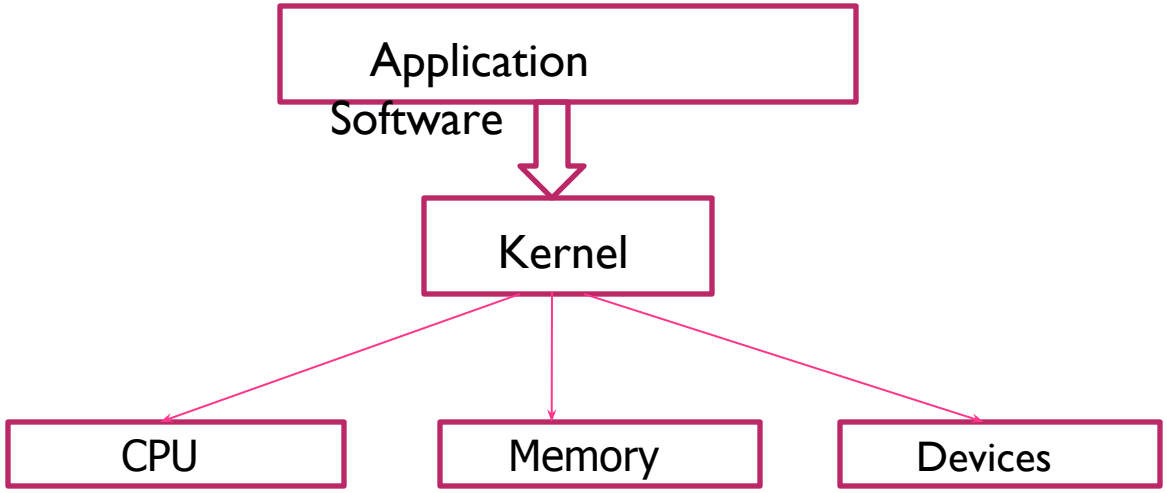
   E.g: Microsoft Windows XP provides GUI through which users can issue commands; alternatively the user can open a command prompt window that accepts type command.

2. **Kernel:**

The software that contains the core components of OS is referred to as the Kernel. The Kernel gets control when :

- **An exceptional situation arises in the system.**
- **An OS policy module explicitly invoke kernel mechanism.**

The OS policy module typically transfers control to the kernel through a software interrupt.



3. **Interrupt :**

Interrupt enables to respond to signal from hardware. The OS may specify set of instructions called as interrupt handler to be executed in response to each type of interrupt. This allows the OS to gain control of the processor to manage system and resources. A processor may generate an interrupt as a result of executing a process instructions.

4. **Memory Management :**

The memory management is an OS component concerned with the system's organization scheme and memory management strategies. The memory manager determines how available memory space is allocated to process and how to respond to changes in a process memory. It also interacts with special purpose memory management hardware to improve performance.

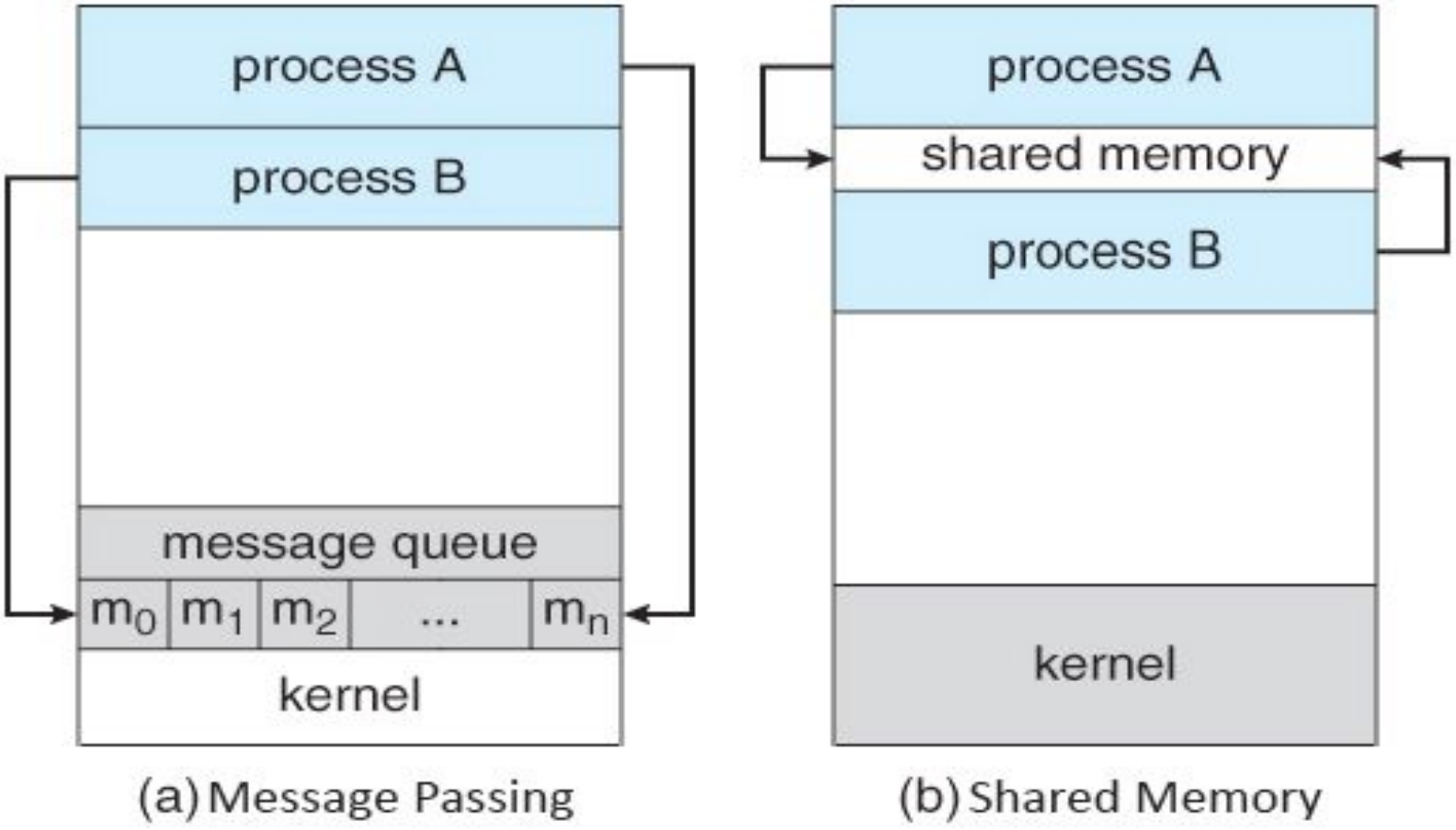
Operations on process

- Create Process
- Destroy Process
- Suspend Process
- Resume Process
- Change Process Priority
- Block Process
- Wakeup Process
- Dispatch Process
- Inter process Communication

Inter process Communication:

- **Interprocess communication (IPC) is a set of programming interfaces that allow a programmer to coordinate activities among different program processes that can run concurrently in an operating system. This allows a program to handle many user requests at the same time. The IPC interfaces make this possible.**
- **To enable a process to communicate with another process.**
- **It is common for process to communicate with one another in multi programmed and networked environment.**
- **IPC is also essential for process that must co-ordinate activities to achieve a common goal.**
- **IPC woks in following two fields:**
  - **Signal**
  - **Message passing**

Diagram of IPC:



- **The two communications models are contrasted in the figure above:**
- **In the message-passing form, communication takes place by way of messages exchanged among the cooperating processes.**
- **In the shared-memory model, a region of memory which is shared by cooperating processes gets established. Processes can be then able to exchange information by reading and writing all the data to the shared region.**
- **IPC usually utilizes shared memory that requires communicating processes for establishing a region of shared memory. Typically a shared-memory region resides within the address space of any process creating the shared memory segment. Other processes that wish for communicating using this shared-memory segment must connect it to their address space.**

Signal :

- Signals are software interrupts that notify a process that an event has occurred.
- Signal does not support data exchange between processes.
- A system signal's depend on the O.S. and software generated interrupts supported by particular processor.
- When signal occurs, O.S. first determines which process should receive the signal and how that process will respond to the signal.
- Processes may catch, ignore or mask a signal.

Message Passing :

- Message can be passed in one direction at a time.
- One process is sender and other is receiver.
- Message passing may be bidirectional, it means that process can act as either a sender or receiver while participating in IPC.
- A blocking send must wait for the receiver to receive the message, requiring that receiver notify the sender when the message is received. Blocking send is example of synchronous communication.
- A non blocking send enables sender to continue with other processing even if the receiver has not yet received the message. This require a message buffering mechanism to hold message until receiver receives it. It is asynchronous communication.

Process Scheduling :

- To keep track of status of process.
- It decides which process gets a processor, when and for how long.
- It is a traffic controller.
- Two objectives of Process Scheduling:
  - The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization.
  - The objective of time sharing is to switch the CPU among processes so frequently that users can interact.

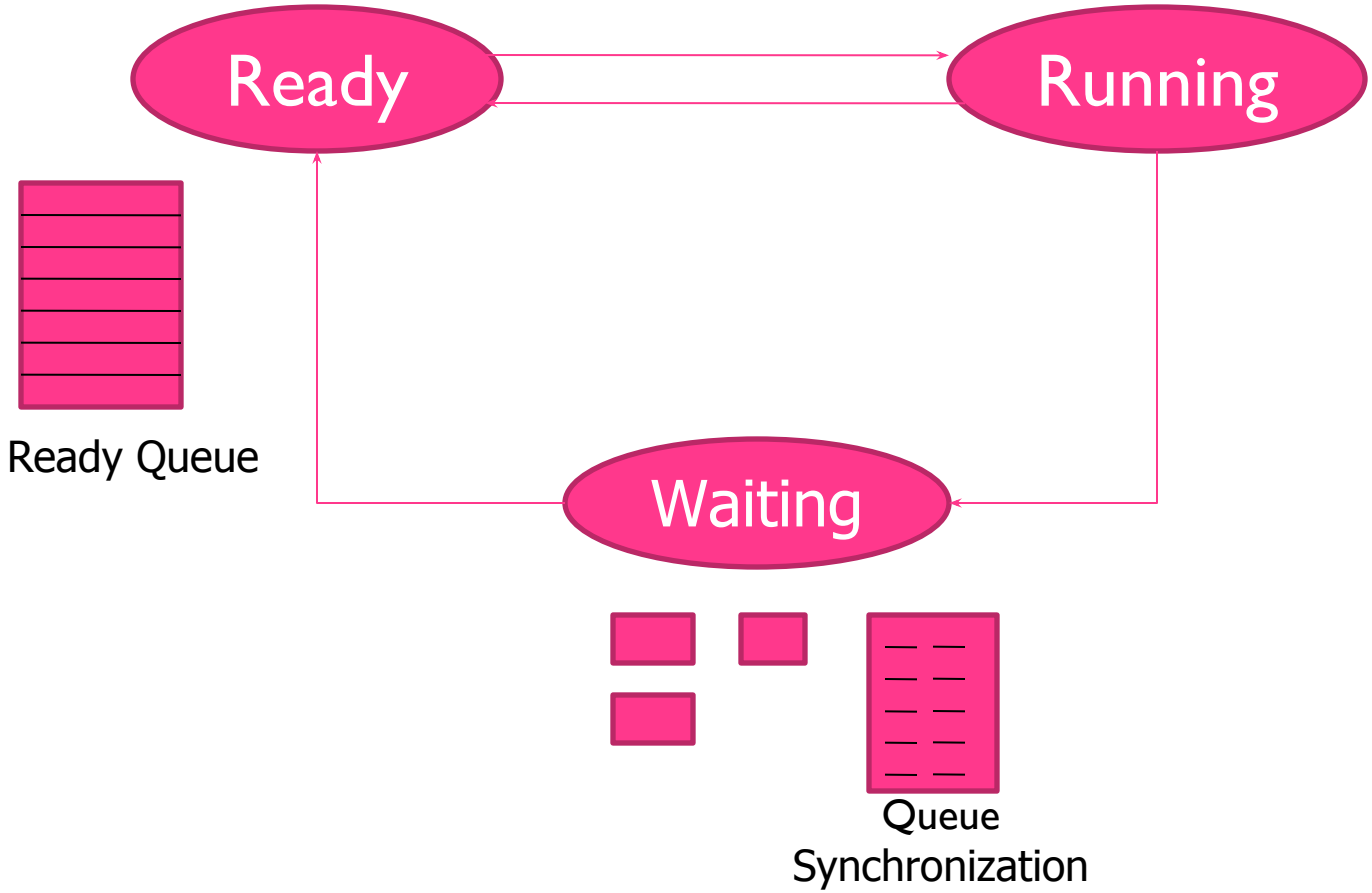
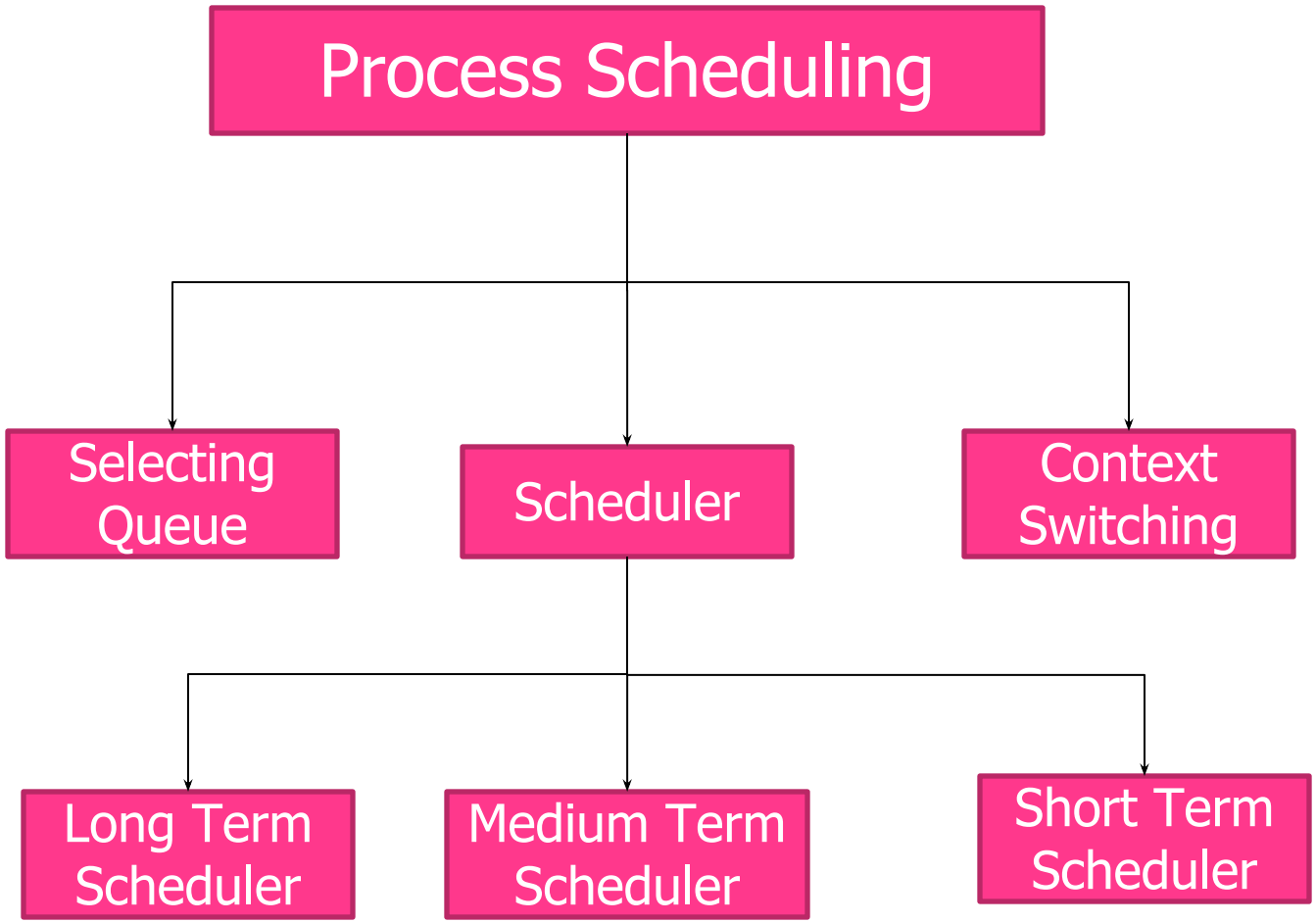


Fig : Process Synchronization


Process Scheduling
Selecting Queue
Scheduler
Context Switching
Long Term Scheduler
Medium Term Scheduler
Short Term Scheduler

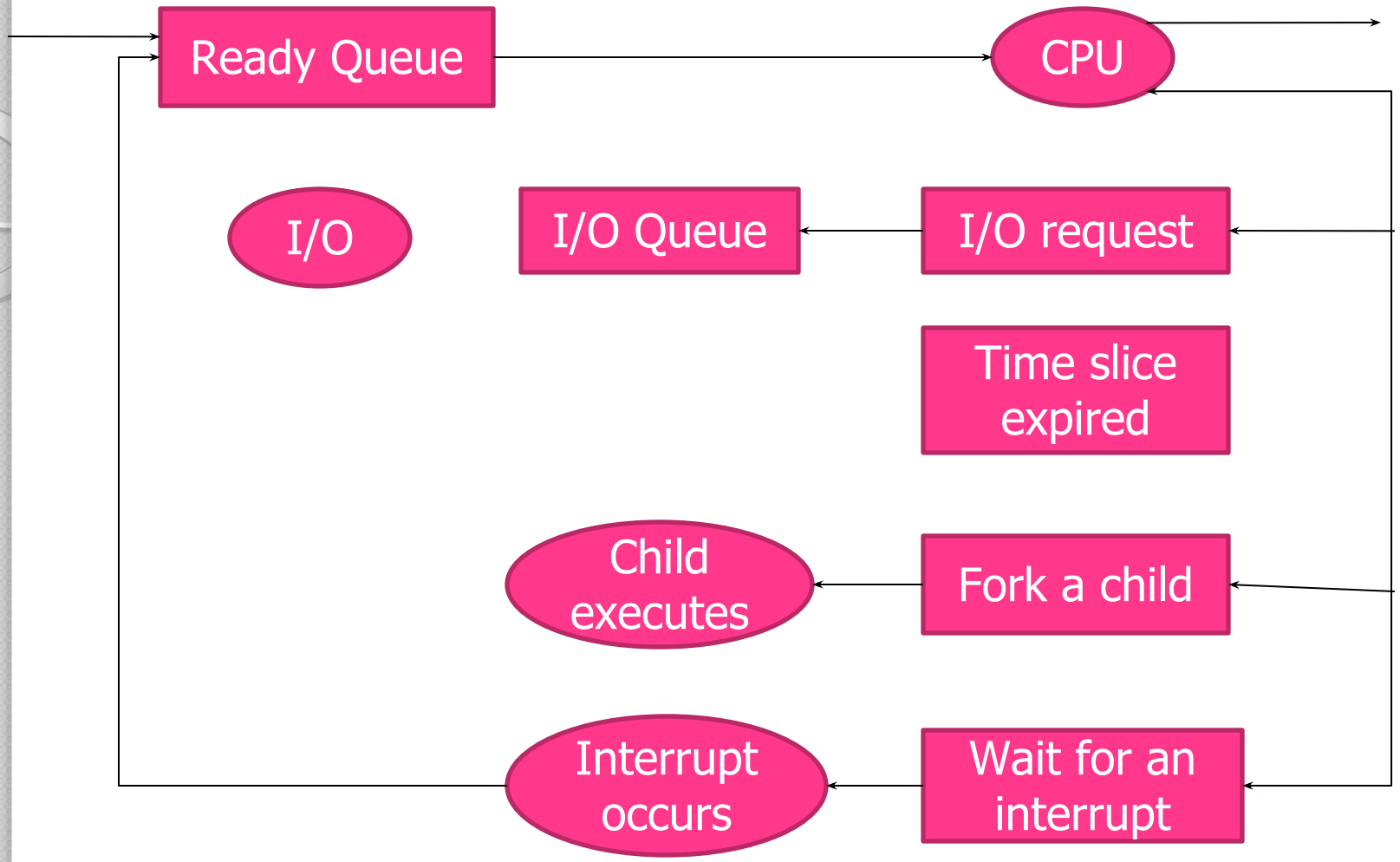


Fig : Queuing diagram representation of Process Scheduling

Assignment Question on UNIT-I

Q.1 . What is Operating system? & Explain Components of operating system. 4

Q.2. Explain services of operating system in detail. 6

Q.3. Explain Characteristics of operating system. 6

Q.4. Explain operation on process? 4

Q.5. Explain Inter process Communication in detail. 5

Q.6. Explain process, process state, process state transition with example. 6

Q.7. Explain Process scheduling? Explain. 5