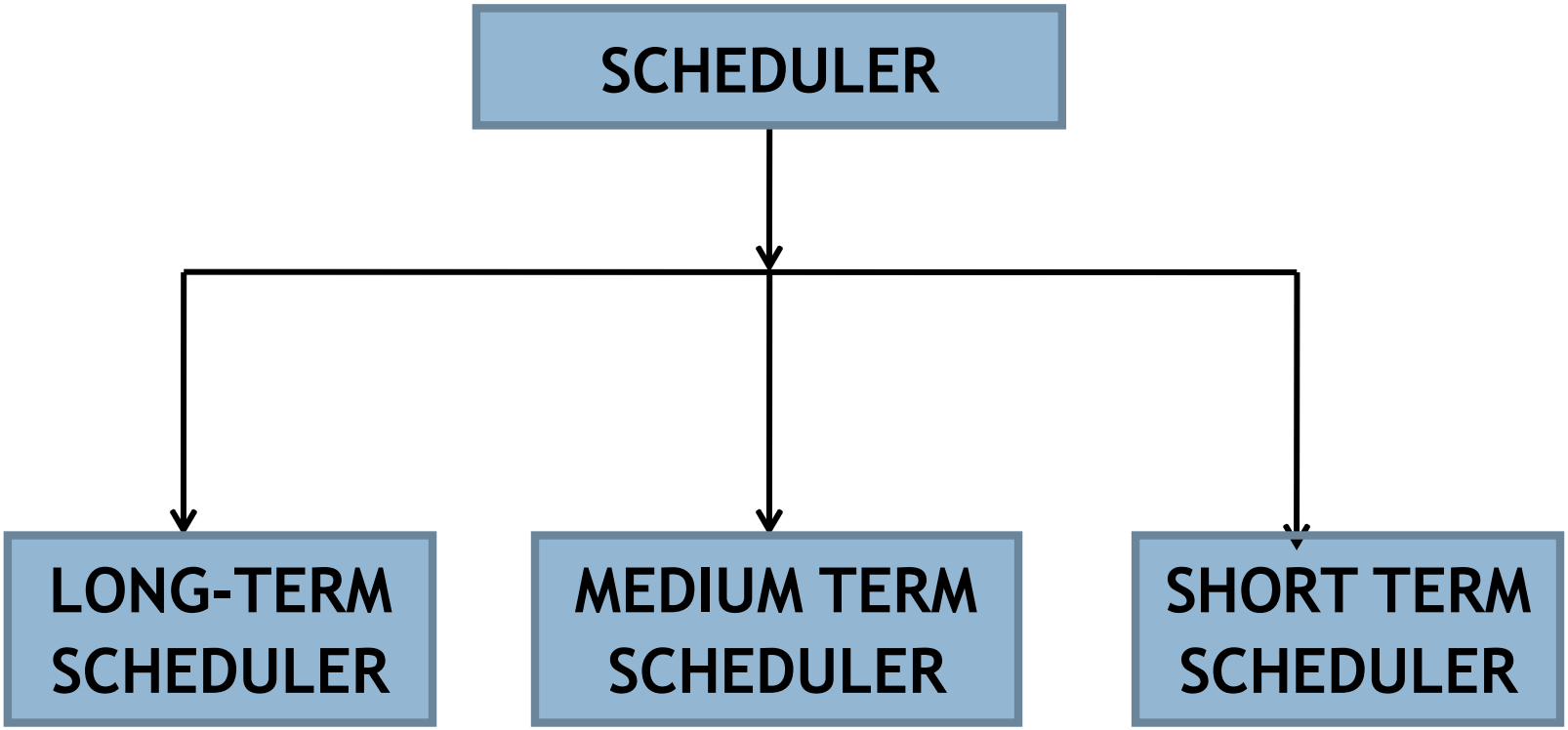
CPU SCHEDULING DEFINITION

- Scheduling is a fundamental OS function.
- Scheduling refers to set of policies and mechanisms built into the OS that governs the order in which the work has to be done by a computer system to complete the task.
- CPU Scheduling is the basis of multiprogrammed OS. By switching the CPU among processes, the OS can make the computer more productive.

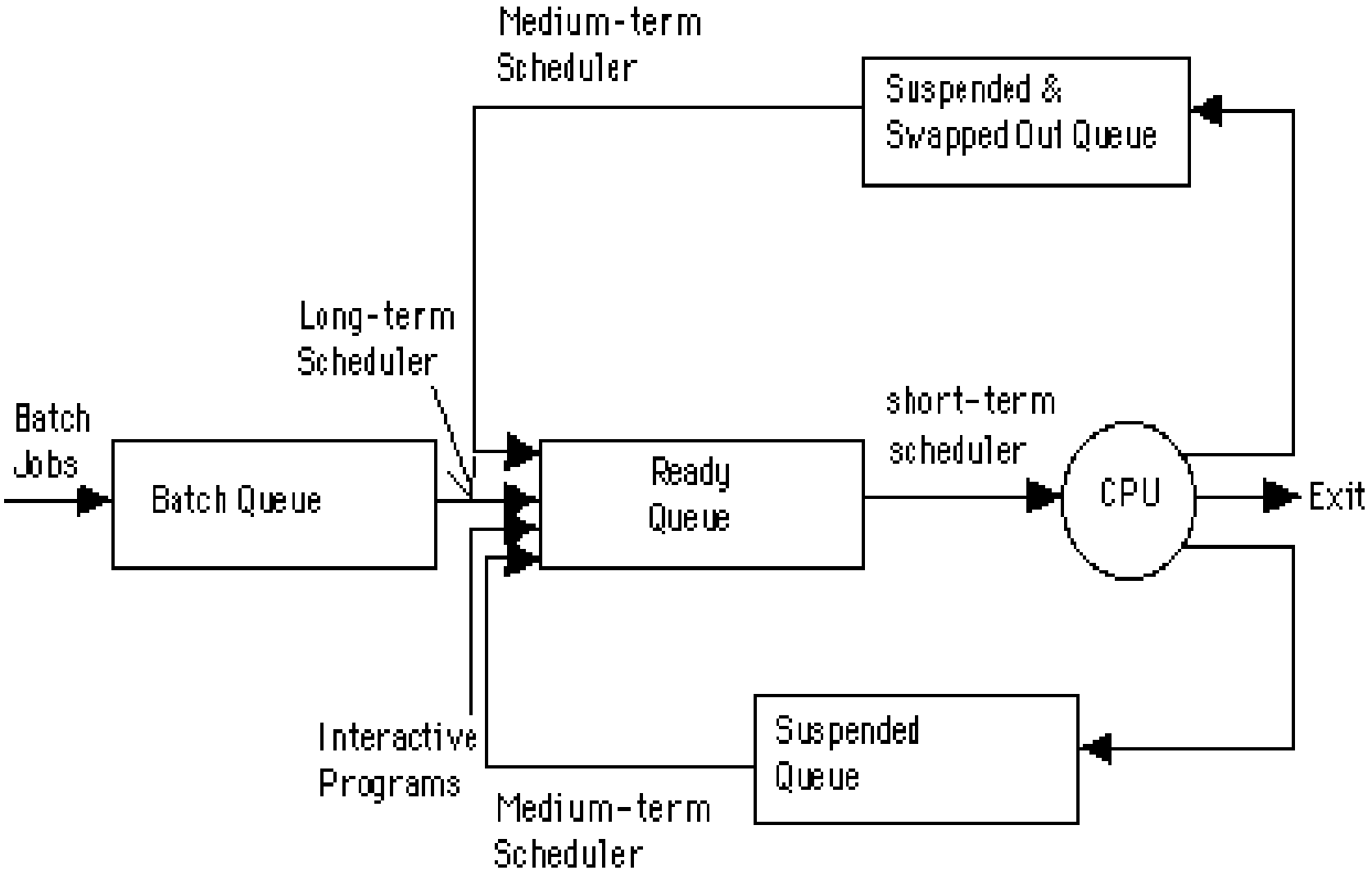
SCHEDULER

- A scheduler is an OS module that selects the next job or process to be admitted into the system.
- Thus, a scheduler selects one of the process from among the processes in the memory that are ready to execute and allocates CPU to it.
- In a complex OS three types of schedulers exist. These are :
  1. Long-term scheduler
  2. Medium term scheduler
  3. Short-term scheduler

TYPES OF SCHEDULER



SCHEDULERS


Medium-term
Scheduler
Suspended &
Swapped Out Queue
Long-term
Scheduler
Batch
Jobs
Batch Queue
Ready
Queue
short-term
scheduler
CPU
Exit
Interactive
Programs
Suspended
Queue
Medium-term
Scheduler

LONG-TERM SCHEDULER

- The long term scheduler works with batch queue and selects the next batch job to be executed. Thus it plans the CPU scheduling for batch jobs.
- Processes, which are resource intensive and have a low priority are called batch jobs. These jobs are executed in a group or bunch. For example, a user request for printing a bunch of files.
- Long term scheduler selects the processes or job from secondary storage device e.g. a disk and loads them into the memory for execution. It is also known as Job Scheduler
- The frequency of execution of a long term is usually low.
- The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound.
- It also controls the degree of multiprogramming.
- If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.
- On some systems, the long-term scheduler may not be available or minimal.

I/O BOUND PROCESS

- I/O bound processes are those that spend most of their time in I/O than computing.

CPU BOUND PROCESS

- CPU bound processes are those that spend most of their time in computations rather than generating I/O requests.

MEDIUM TERM SCHEDULER

- Medium-term scheduling is a part of **swapping**.
- It removes the processes from the memory. It *reduces the degree of multiprogramming*.
- The medium-term scheduler is in-charge of handling the *swapped out-processes*.
- A running process may become suspended because of an I/O request or by a system call.
- Such a suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other processes, the suspended process is moved to the secondary storage. This process is called ***swapping***, and the process is said to be ***swapped out or rolled out***.
- Swapping may be necessary to improve the process mix.

SHORT TERM SCHEDULER

- It is also called as **CPU scheduler**.
- it selects from among the ready processes that are residing in the main memory and allocates CPU to one of them.
- It is the change of ready state to running state of the process.
- As compared to long-term schedulers, a Short-term scheduler has to work very often i.e., the frequency of execution of Short-term schedulers is high.
- The Short-term scheduler must select a new process for CPU frequently.
- A process may execute for only a few milliseconds before waiting for an I/O request. Often, the short-term scheduler executes atleast once every 100 milliseconds. Because of the brief time between executions, the short-term schedulers are fast.
- Short-term schedulers, also known as *dispatchers*, as they make the decision of which process to execute next.
- Short-term schedulers are faster than long-term schedulers.

Comparison among Scheduler

S.N.	Long-Term Scheduler	Short-Term Scheduler	Medium-Term Scheduler
1	It is a job scheduler	It is a CPU scheduler	It is a process swapping scheduler.
2	Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between both short and long term scheduler.
3	It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming	It reduces the degree of multiprogramming.
4	It is almost absent or minimal in time sharing system	It is also minimal in time sharing system	It is a part of Time sharing systems.
5	It selects processes from pool and loads them into memory for execution	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

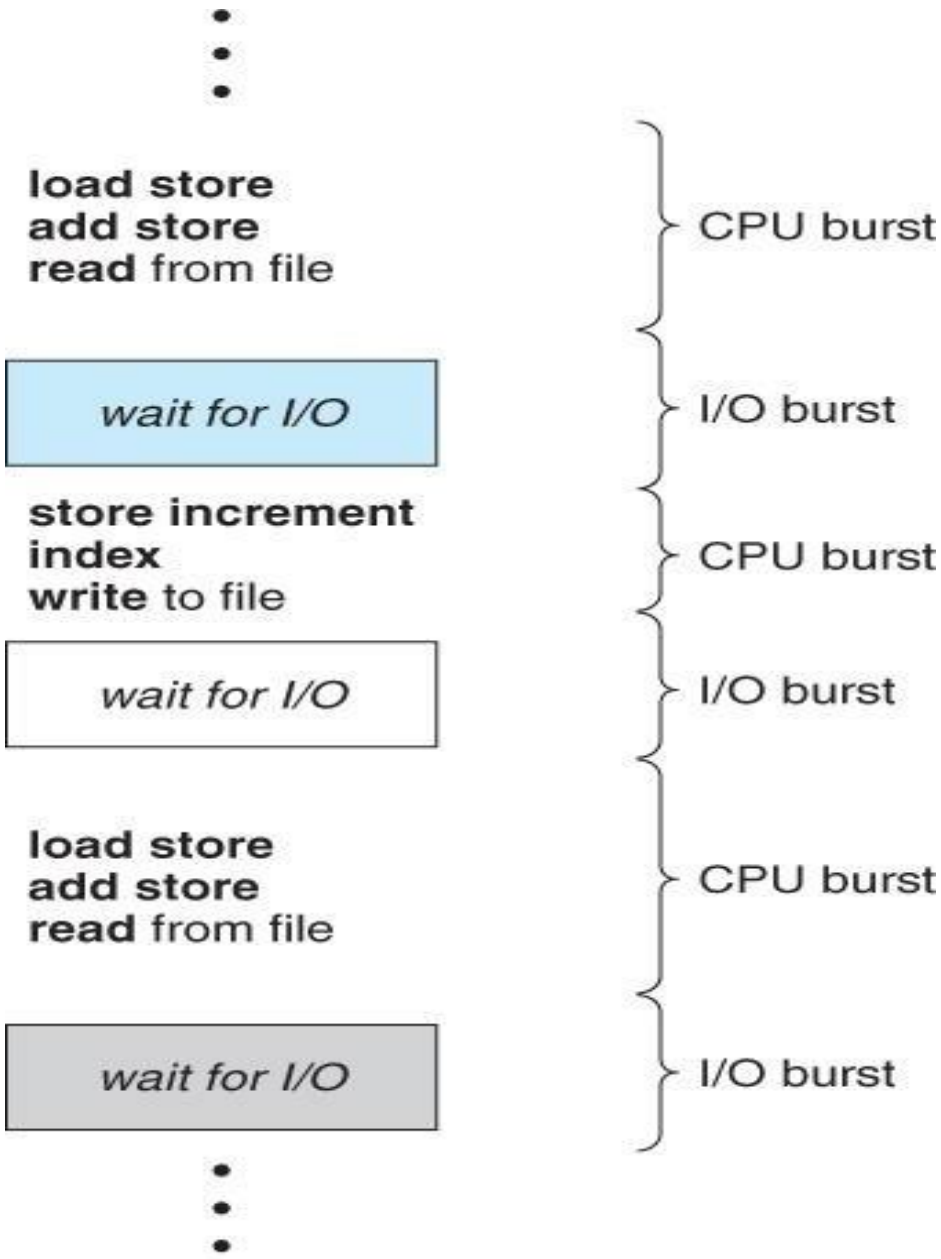
CPU-I/O BURST CYCLE

- Almost all programs have some alternating cycle of CPU number crunching and waiting for I/O of some kind. (Even a simple fetch from memory takes a long time relative to CPU speeds.)
- In a simple system running a single process, the time spent waiting for I/O is wasted, and those CPU cycles are lost forever.
- A scheduling system allows one process to use the CPU while another is waiting for I/O, thereby making full use of otherwise lost CPU cycles.
- The challenge is to make the overall system as "efficient" and "fair" as possible, subject to varying and often dynamic conditions, and where "efficient" and "fair" are somewhat subjective terms, often subject to shifting priority policies.

CPU-I/O Burst Cycle

- Almost all processes alternate between two states in a continuing ***cycle***, as shown in Figure below *:*
  - ***A CPU burst of performing calculations, and***
  - ***An I/O burst, waiting for data transfer in or out of the system.***

FIGURE : ALTERNATIVE SEQUENCE OF CPU AND I/O BURSTS



CPU SCHEDULER

- Whenever the CPU becomes idle, it is the job of the CPU Scheduler (a.k.a. the short-term scheduler) to select another process from the ready queue to run next.
- The storage structure for the ready queue and the algorithm used to select the next process are not necessarily a FIFO queue. There are several alternatives to choose from, as well as numerous adjustable parameters for each algorithm, which is the basic subject of this entire chapter.

PREEMPTIVE & NON-PREEMPTIVE SCHEDULING

- **Preemptive Scheduling**
- CPU scheduling decisions take place under one of four conditions:
  - When a process switches from the running state to the waiting state, such as for an I/O request or invocation of the wait( ) system call.
  - When a process switches from the running state to the ready state, for example in response to an interrupt.
  - When a process switches from the waiting state to the ready state, say at completion of I/O or a return from wait( ).
  - When a process terminates.
- For conditions 1 and 4 there is no choice - A new process must be selected.
- For conditions 2 and 3 there is a choice - To either continue running the current process, or select a different one.
- If scheduling takes place only under conditions 1 and 4, the system is said to be ***non-preemptive***, or ***cooperative***. Under these conditions, once a process starts running it keeps running, until it either voluntarily blocks or until it finishes. Otherwise the system is said to be ***preemptive***.

- Under non-preemptive scheduling, once the CPU has been allocated to a process, the process keeps the CPU untill it releases the CPU either by terminating or by switching to the waiting state.
- It is the only method that can be used on certain hardware platforms, because it does not require the special hardware platforms. For example: timer needed for preemptive scheduling.
- Unfortunately, preemptive scheduling incurs a cost.
- Consider the case of two processes sharing data. One may be in the midst of updating the data, when it is preempted and the second process is run. The second process may try to read the data, which are currently in an inconsistent state. New mechanisms thus are needed to co-ordinate access to shared data.

DISPATCHER

- Another component involved in the CPU scheduling is the dispatcher.
- The **dispatcher** is the module that gives control of the CPU to the process selected by the scheduler.
- This function involves:
  - Switching to user Switching context.
  - mode.
  - Jumping to the proper location in the newly loaded program.
- The dispatcher should be as fast as possible, given that it is invoked during every process switch.

  What is dispatch latency? (VVIMP 2marks)
- The dispatcher needs to be as fast as possible, as it is run on every context switch. The time consumed by the dispatcher is known as **dispatch latency.**

SCHEDULING CRITERIA (5 MARKS)

- In choosing, which algorithm to use in a particular situation, we must consider the properties of the various algorithms.
- Many criteria have been suggested for CPU Scheduling algorithm.
- The characteristics used for comparision can make a substantial difference in the determination of the best algorithm.
- The criteria include the following:

1. CPU Utilization :

- This is the percentage of time that the CPU/Processor is busy.
- CPU Utilization may range from 0 to 100%.
- In a real system, it should range from 40% (for a lightly loaded system) to 90% (for a heavy loaded system).
- It is system oriented and performance related.

- 2. Throughput :
- The scheduling policy should attempt to maximize the number of process completed per unit of time.
- This is the measure of how much work is being performed.
- If the CPU is busy executing the processes, then work is being done.
- ***One measure of work is the number of processes completed per unit time is called throughput.***
- For long processes, this rate may be 1 process per hour; for short processes/transactions throughput might be 10 processes per second.
- 3. Turn around time :from the point of view of a particular process, the important criterion is how long it takes to execute that process.
- ***The interval of time between the submission of a process and its completion is the turn around time.***

$$\text{Turn Around Time} = \text{actual execution time} + \text{time spent waiting for resources including the processor and doing I/O}$$

- This is an appropriate measure for a batch job.

4. Waiting time

- The CPU scheduling algorithm does not affect the amount of time during which a process executes or does I/O, it affects only the amount of time that a process spends waiting in the ready queue.
- Waiting time is the sum of the periods spent waiting in the ready queue.

5. Response Time

- The measure of time from the submission of a request until the first response is produced.
- This measure called response time, is the amount of time it takes to start responding, but not the time that it takes to output that response.

4. Waiting Time

- It is the average period of time a process spends waiting.
- It is a time spent in waiting for a resource allocation. Therefore, waiting time is the penalty imposed for sharing resources with others.
- Waiting time presents a more accurate measure of performance as compared to turnaround time because it does not include the time a process is executing on the CPU or performing I/O. It includes only the time a process spends waiting.
- Waiting time can be expressed as turnaround time minus the actual execution time.
- $W(x)=T(x)-x$
- Where x is the service time, W(x) is the waiting time of job requiring x units of service and T(x) is the job's turnaround time.