- **First type should be bold (taken directly from a past midterm)**
- Should be underlined (taken from tutorial and stuff)
- 3rd type could be questions we make ourself -not important

Chapter #3

Q1) What is a process state? Mention the various states of a process?
Answer: *As a process executes, it changes states. The state of a process is defined by the current activity of that process. Each process may be in one of the following states:*

I. *New*
II. *Running*
III. *Waiting*
IV. *Ready*
V. *Terminated*

Q2) What is PCB(Process Control Block)?
Answer: *Each process is represented in the OS by a PCB, it contains the following information:*

I. *Process State*
II. *Program Counter: The counter indicates the address of the next instruction to be executed for this process.*
III. *CPU registers*
IV. *CPU scheduling information*
V. *Memory-Management information*
VI. *I/O status information*

Q3) What is the use of job, ready and device queues?
Answer:

I. *Job Queue: As a process enters a system, they are put into a job queue. This queue consists of all jobs in the system.*
II. *Ready Queue: The processes that are residing in main memory and are ready & waiting to execute are kept in ready queue.*
III. *Device Queue: The list of processes waiting for I/O device is kept in the device queue. Each device has its own device queue.*

Q4) What is meant by Context Switch?
Answer: *Switching the CPU to another process requires saving the state of old process & loading the saved state of the new process. This task is known as Context Switch.*
The context of a process is represented in the PCB of the process.

Q4) **What are short, long and medium-term scheduling?**

Long term scheduler *determines which programs are admitted to the system for processing. It controls the degree of multiprogramming. Once admitted, a job becomes a process.*
Medium term scheduling *is part of the swapping function. This relates to processes that are in a blocked or suspended state. They are swapped out of real-memory until they are ready to execute. The swapping-in decision is based on memory-management criteria.*
Short term scheduler*, also know as a dispatcher executes most frequently, and makes the finest-grained decision of which process should execute next. This scheduler is invoked whenever an event occurs. It may lead to interruption of one process by preemption.*

Chapter #4

Q1) Define Threads?
Answer: *Thread is a basic unit of CPU utilization, it comprises a thread ID, a program counter, a register set, and a stack.*

Q2) Give two different scenarios of thread cancellation?
Answer: *A thread that is to be cancelled is often referred to as the target thread.*

I. *Asynchronous cancellation: One thread immediately terminates the target thread.*
II. *Deferred cancellation: The target thread can periodically check if it should terminate, allowing the target thread an opportunity to terminate itself in an orderly fashion.*

Q3) What is the use of Fork and Exec system call?
Answer: *Fork is a system call by which a new process is created. Exec system call is used after a fork by one of the two processes to replace the process memory space with a new program.*

Q4) Define thread pool?
Answer: *Is to create a number of threads at process startup and place them into a pool, where they sit and wait for work.*

Q5) Define dispatcher?
Answer: *The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler. This function involves*

I. *Switching context*
II. *Switching to user mode*
III. *Jumping to the proper location in the user program to restart that program.*

Q6) Define Dispatch latency?
Answer:. *The time it takes for the dispatcher to stop one process and start another running.*

Q7) What are the Scheduling Criteria?
Answer:

I. *CPU utilization*
II. *Throughput*
III. *Turnaround time*
IV. *Waiting time*
V. *Response time*

Q8) Define multilevel queue- scheduling algorithm.
Answer: *A multilevel queue- scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority, or process type. Each queue has its own scheduling algorithm.*

Q9)What are the benefits of multithreaded programming?
Answer: The benefits of multithreaded programming can be broken down into three major categories:

I. Responsiveness
II. Resource sharing
III. Economy
IV. Scalability

Q10) Define CPU scheduling?
Answer: CPU scheduling is the process of switching the CPU among various processes. CPU scheduling is the basis of multiprogrammed OS's. by switching the CPU among processes, the OS can make the computer more productive.

MCQ's

Q11) A scheduler which selects processes from secondary storage device is called

(A) Short term scheduler. (B) Long term scheduler.

(C) Medium term scheduler. (D) Process scheduler.

Ans: (C)

Q12) Interval between the time of submission and completion of the job is called

(A) Waiting time (B) Turnaround time

(C) Throughput (D) Response time

Ans: (B)

Q13) Q.10 The scheduling in which CPU is allocated to the process with least CPU-burst time is called

(A) Priority Scheduling (B) Shortest job first Scheduling

(C) Round Robin Scheduling (D) Multilevel Queue Scheduling

Ans: (B)

Q14) The “turn-around” time of a user job is the

(A) time since its submission to the time its results become available.

(B) time duration for which the CPU is allotted to the job.

(C) total time taken to execute the job.

(D) time taken for the job to move from assembly phase to completion phase.

Ans: (C)

Q15) Program ‘preemption’ is

(A) forced de allocation of the CPU from a program which is executing on the CPU.

(B) release of CPU by the program after completing its task.

(C) forced allotment of CPU by a program to itself.

(D) a program terminating itself due to detection of an error.

Ans: (A)

Q16) Which scheduling policy is most suitable for a time-shared operating system

(A) Shortest-job First. (B) Elevator.
(C) Round-Robin. (D) First-Come-First-Serve.

Ans: (C)

Q17) Differentiate between pre-emptive and non-pre-emptive scheduling.

Ans: In a pre-emptive scheduling approach, CPU can be taken away from a process if there is a need while in a non-pre-emptive approach if once a process has been given the CPU, the CPU cannot be taken away from that process, unless the process completes or leaves the CPU for performing an Input Output.

Pre-emptive scheduling is more useful in high priority process which requires immediate response, for example in real time system. While in nonpreemptive systems, jobs are made to wait by longer jobs, but treatment of all processes is fairer.

Q18) Explain any three policies for process scheduling that uses resource consumption information.

Answer: Three policies for process scheduling are described below in brief:

1. First-come First-served (FCFS) (FIFO)
– Jobs are scheduled in order of arrival
– Non-preemptive

• Problem:
– Average waiting time can be large if small jobs wait behind long ones
– May lead to poor overlap of I/O and CPU and convoy effects

2. Shortest Job First (SJF)
– Choose the job with the shortest next CPU burst
– Provably optimal for minimizing average waiting time

• Problem:
– Impossible to know the length of the next CPU burst

3. Round Robin(RR)
• Problem:
– Jobs are scheduled in order of arrival
– Non-preemptive
– Average waiting time can be large if small jobs wait behind long ones
– May lead to poor overlap of I/O and CPU and convoy effects
– Choose the job with the shortest next CPU burst
– Provably optimal for minimizing average waiting time
– Impossible to know the length of the next CPU burst
– Often used for timesharing

– Ready queue is treated as a circular queue (FIFO)
– Each process is given a time slice called a quantum
– It is run for the quantum or until it blocks
– RR allocates the CPU uniformly (fairly) across all participants. If average queue length is n, each participant gets 1/n
– As the time quantum grows, RR becomes FCFS
– Smaller quanta are generally desirable, because they improve response time
– Context switch overhead of frequent context switch