

ID

16950

Name

Muhammad Asif

Subject

operating systems

Teacher

Sir. Daud Khan.

ID # 16950 Muhammad Asif

## Section "A"

- 1) interrupt.
- 2) Memory related information.
- 3) wait
- 4) Asymmetric
- 5) DS
- 6) Fg
- 7) jobs
- 8) <Ctrl-C>
- 9) All of these
- 10) usability
- 11) /Lib
- 12) long term
- 13) do not
- 14) Semaphore
- 15) Spinlock
- 16) false

ID # 16950 Muhammad Asif

- 17 True
- 18 Bounded waiting.
- 19 firmware base solution
- 20 Medium term Scheduler.

ID# 18950

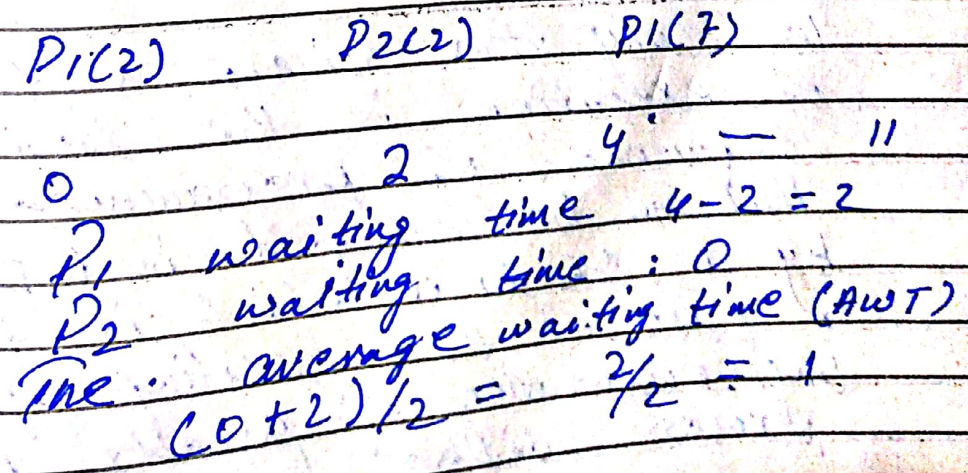
Manhannad Ash

# Q1. Procedure for Preemptive Job first scheduling.

In the shortest remaining time first scheduling algorithm, the process with the smallest amount of time remaining to execute until completion is selected to execute. Since the currently executing process is the one with the shortest amount of time remaining by definition, and since that time should only reduce as execution progress, process will always run until they complete or a new process is added that requires a smaller amount of time.

Example:-

Process	Duration	order	Arrival time
P <sub>1</sub>	4	1	0
P <sub>2</sub>	2	2	2



Q2 if a process exits and there are still threads of that process running, will they continue to run?

Ans No; threads of the process will no longer run once the process is terminated.

Because all threads in a process share the same address space. all threads are suspended at the same time. Similarly termination of a process terminates all threads within that process.

Q3 Resource sharing feature:-

Ans By default threads share common code, data and other resources which allows multiple tasks to be performed simultaneously in a single address space.

In computer science, a thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system. The implementation of threads and process differs between operating.

TD# 16950

Muhammad Hsf

Systems. but in most cases a thread is a component of a process. Multiple threads can exist within one process, executing concurrently and sharing resources such as memory, while different processes do not share these resources. In particular the threads of a process share its executable code and the values of its dynamically allocated variables and non-thread-local global variables at any given time.