Assignment 3

Q1 – Written question (2 marks)

Explain why disabling interrupts is not appropriate for solving synchronization issues in multiprocessor systems.

Firstly, Process Synchronization means maintaining data consistency and ensure synchronized execution of cooperating processes. Solving synchronization issue is about handling problems that arose while multiple process executions. Such as the solution of critical section problem and dinning philosopher problem.

Secondly, the disabling interrupt let process disables all interrupts just before entering its CS and re-enable them just before leaving the CS. Basically, it works to prevent preemption of kernel-mode code.

However, disabling interrupts is insufficient for the synchronization issues in multiprocessor systems because the main line can be executing on both processors at the same time, so interrupts are not involved. What is more, one processor cannot block interrupts on the other.

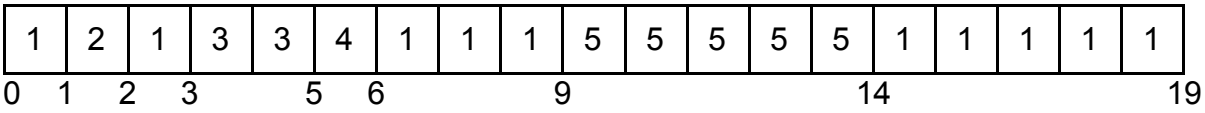
Q2 – Written question (4 marks)

Consider the following set of processes (all times given in seconds):

Process	Arrival Time	Burst Time
P1	0	10
P2	1	1
P3	3	2
P4	5	1
P5	9	5

Draw a Gantt chart to illustrate the execution of these processes using the SRTN scheduling algorithm. Use FCFS to break ties. What is the average wait time?

process	arrival time	burst time	start time	finish time	turnaround	waitting
P1	0	10	0	19	19	9
P2	1	1	1	2	1	0
P3	3	2	3	5	2	0
P4	5	1	5	6	1	0
P5	9	5	9	14	5	0

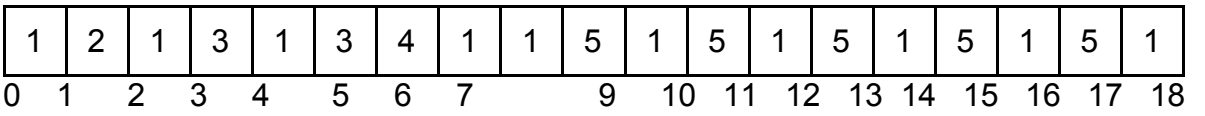


According to the scheduling table, the average waiting time is 9/5 = 1.8 sec.

Q3 – Written question (4 marks)

For the same set of processes as in Q2, draw a Gantt chart to illustrate the execution of these processes using the RR scheduling algorithm with 1 sec quantum. Use FCFS to break ties. How many context switches are there?

process	arrival time	burst time	start time	finish time	turnaround	waitting
P1	0	10	0	19	19	9
P2	1	1	1	2	1	0
P3	3	2	3	6	3	1
P4	5	1	6	7	2	1
P5	9	5	9	18	9	4



Context switch time = 18

Q4 – Written question (4 marks)

For the same set of processes as in Q2, assume that P1 and P5 are high priority processes, P2 and P4 are medium priority processes, and P3 is a low priority process. Draw the Gantt chart to illustrate the execution of these processes using the multi-level feedback queue scheduling algorithm. The high priority level queue employs RR with 2s quantum, the medium priority level queue employs RR with 4s quantum, and the low priority level queue employs FCFS.

process	arrival time	burst time	start time	finish time	turnaround	waitting
P1	0	10	0	19	19	9
P2	1	1	2	3	2	1
P3	3	2	3	5	2	0

P4	5	1	5	12	7	6
P5	9	5	8	15	6	1

1	1	2	3	3	1	1	1	1	5	5	5	4	5	5	1	1	1	1

0 3 4 6 8 10 12 13 15
19

Question 6

medium.txt

threads	observed timing(s)	observed speed up compared to original	expected speed up
original program	21.372	1.0	1.0
1	18.252	3	1.0
2	11.627	10	2.0
3	10.5621	11	3.0
4	9.451	12	4.0
8	8.000	13	8.0
16	7.0292	14	16.0

hard2.txt

threads	observed timing	observed speed up compared to original	expected speed up
original program	7.7346	1.0	1.0
1	5.621	2	1.0
2	4.8672	3	2.0
3	4.2443	3.5	3.0
4	3.7879	4	4.0
8	2.8754	5	8.0
16	1.7654	6	16.0

hard.txt

threads	observed timing	observed speed up compared to original	expected speed up
original program	7.7439	1.0	1.0
1	6.7832	1	1.0
2	5.6478	2	2.0
3	4.543	3	3.0
4	4.3435	3.5	4.0
8	3.234	4	8.0
16	2.345	5	16.0

Explanation: The result from these diagrams does not match the expected results very accurately. At the beginning the running time decrease faster than the expected speeds. However, when the input thread is bigger than 4, the running time decrease slower. The reason is that, with user input lager thread numbers. the program needs to create more threads, and do more operations such as checking the status, that takes some more time.