Operating Systems/ CS2002/CSE&IT/5th/2017

Mid-Semester Examination

School of Computer Engineering
KIIT University, Bhubaneswar-24

Time: 1.5hrs ***Full Mark: 25***

(Answer Any five Questions including Q.No.1)

Q1)Answer all questions. [1×5]

(a)Why the FCFS scheduling is always non preemptive type?

(b) Two processes P_0 and P_1, share the following variables:

```
boolean key;
boolean lock ; /* initially false */
```

Structure of the processes is given below:

```
do {
key = True;
while(key == True)
{
Swap(&lock, &key);
}
// critical section
lock = True;
// remainder section
} while(True);
```

The Swap() is as follows:

```
void Swap(boolean *a, boolean *b)
{
boolean x = *a;
*a = *b;
 *b = x;
}
```

Whether the above mentioned algorithm ensure mutual exclusion requirement? Justify.

(c)What are the benefits of threads over processes?

(d) If Round Robin is used with a time quantum of 2 seconds, what will be the turnaround time for the process P2 ?

Process	Next CPU Burst Time
P1	9 min
P2	1 sec

(e)What is aging priority? Why is it used?

Q2) (a)Compare among different schedulers that can exist in an operating system? [2.5] (b)Consider the following snapshot of the system: [2.5]

Process	Next CPU Burst Time(ms)	Arrival Time
P1	10	0
P2	5	1
P3	2	2
P4	1	3

Using shortest remaining time first scheduling, find the waiting time of each process.

Q3) What are the various states of a process? Explain about the state transitions of a process during its life. [5]

Q4)What are the conditions for a solution to critical section problem? Design a solution to critical section problem involving 2-processes. Justify that the solution is satisfying the conditions. [5]

Q5) Consider the following processes arrived in a system. [5]

Process	Next CPU Burst Time(ms)	Arrival Time
A	4	0
B	5	1
C	6	2
D	3	3
E	1	4
F	4	5

Calculate the average waiting time of the processes if the scheduling algorithm is Round Robin with time slice length as 2 ms.

Q6) What is busy waiting? How a semaphore can be implemented to have no busy waiti

[5]

Q7)Explain Dinning philosopher problem. Develop a deadlock free semaphore based solution to solve the dinning philosopher problem. Whether the solution is free from starvation? [5]