AKGEC/IAP/FM/02

Ajay Kumar Garg Engineering College, Ghaziabad

Department of MCA

Sessional Test-2 Solution

Course:	MCA	Semester:	III
Session:	2017-18	Section:	MCA-1 & 2
Subject:	OS	Sub Code:	RCA-301
Max Marks:	50	Time:	2 hour

Note: Answer **all** the sections.

Section-A

A. Attempt **all** the parts. **(5 X 2 = 10)**

1. What is a process? What are attributes of a process?

Answer:- A program in execution is known as process.
Attributes of process are –
Program counter, stack, data section.

2. What is context switching?

Answer:- Switching the CPU to another process requires saving the state of the old process & loading the saved state for the new process. This task is known as context switching.

3. What are necessary conditions for deadlock to occur?

Answer:- a) Mutual Exclusion
b) Hold & wait
c) No preemption
d) Circular wait

4. Define binary semaphores.

Answer:- They are those semaphores that have only two values 0 & 1.

5. What is the relationship between threads and processes?

Answer: a) Like processes, threads share CPU & only one thread is active at a time.

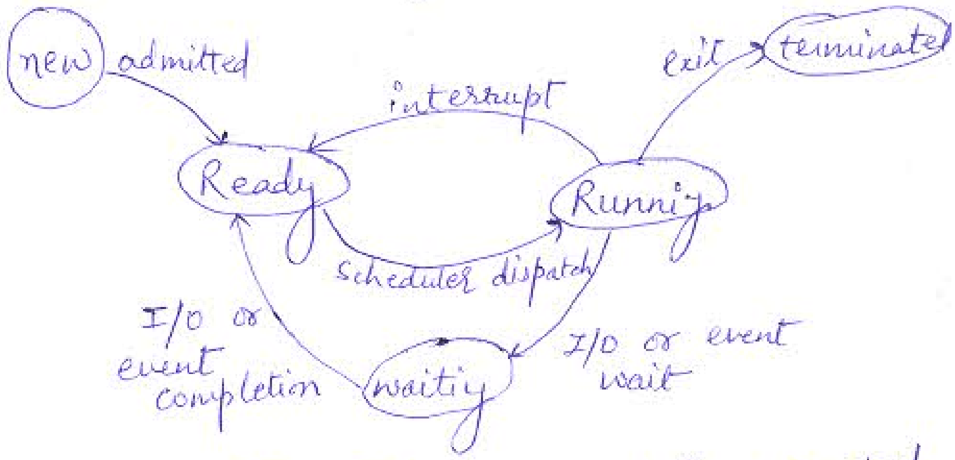
b) Like processes, threads within a process execute sequentially.

Section-B

B. Attempt **all** the parts. **(5 X 5 = 25)**

6. Draw the process state diagram and describe the various process states.

Answer:- A process goes through various process states -



a) New state :- The process being created.

b) Running state :- A process is using CPU at that instant.

c) Waiting / Blocked state :- A process is said to be blocked if it is waiting for some event to happen such as I/O completion etc

d) Ready state :- waiting to get processor allocated.

e) Terminated state :- The process has finished execution.

7. Explain different kinds of threaded models.

Answer:- 1. Many to One Model

a) In this many user level threads are mapped to a single kernel thread.

b) Thread management is handled by thread library in user space.

c) If a blocking system call is made, entire process blocks.

2. One-to-One Model :-

a) This model creates a seperate kernel thread to handle each user thread.

b) Overhead of managing this model is more significant.

c) Most implementations of this model place a limit on how many threads can be created.

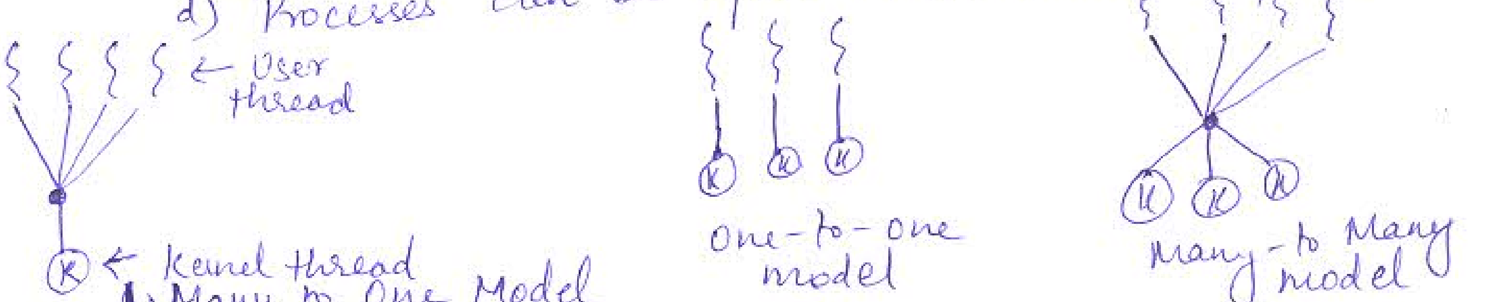
3. Many-to-Many model -

a) This model multiplexes any number of user threads onto an equal or smaller number of kernel threads, combining the best features of above two models.

b) Users have no restriction on the number of threads created.

c) Blocking kernel system calls do not block entire process.

d) Processes can be split across multiple processors.



8. What is PCB? What kind of information is stored in PCB?

Answer:- For each process there is a Process Control Block (PCB) which stores process specific information in it. Details stored in PCB are as follows;-

a) Process state - Running, waiting etc.

b) Process Id & parent process Id.

c) CPU registers & Program counter - These need to be saved and restored when swapping processes in & out of CPU

d) CPU scheduling information - Such as priority information & pointers to scheduling queues.

e) Memory mgt. info - Eg. page tables / segment tables.

f) Accounting information - user & kernel CPU time consumed, account numbers, limit etc.

g) I/O status information - Devices allocated, open file tables etc.

9. Explain the differences in the degree to which the following scheduling algorithms discriminate in favor of short processes: a) First Come First Served b) Round Robin c) Multilevel feedback queues.

Answer:-

(a) FCFS :- discriminates against short jobs since any short jobs arriving after long jobs will have a longer waiting time.

(b) Round Robin :- Treats all jobs equally (giving them equal bursts of CPU time) so short jobs will be able to leave the system faster since they will finish first.

(c) Multilevel feedback queues - It depends on exact algorithm used to move processes between queues. If the scheduling algorithm tends to move I/O bound processes to higher priority queues, then short I/O bound processes will be favored over short CPU bound processes, both of which will be favored over longer processes.

10. Consider the following snapshot of a system:

	Max	Allocation	Available
	A B C	A B C	A B C
P_0	0 0 1	0 0 1	
P_1	1 7 5	1 0 0	
P_2	2 3 5	1 3 5	
P_3	0 6 5	0 6 3	
Total		2 9 9	1 5 2

Answer the following questions using the banker's algorithm:

a) What is the content of the matrix Need?

b) Is the system in a safe state?

Answer:-

(a) Need Matrix = Max - Allocation

	A	B	C
P_0	0	0	0
P_1	0	7	5
P_2	1	0	0
P_3	0	0	2

(b) Banker's safety Algorithm —

Finish [i] = False $\forall$ $i \in \{0,1,2,3\}$

work = (1,5,2)

For P_0 $Need_0 <$ work $\therefore$ Finish [0] = F & work = (1,5,3)

For P_2 $Need_2 <$ work $\therefore$ Finish [2] = F & work = (1,5,3) + (1,3,5) = (2,8,8)

For P_3 $Need_3 <$ work $\therefore$ Finish [3] = F & work = (2,8,8) + (0,6,3) = (2,14,11)

For P_1 $Need_1 <$ work $\therefore$ Finish [1] = F & work = (2,14,11) + (1,0,0) = (3,14,11)

$\therefore$ System is safe and safe sequence is $\langle P_0, P_2, P_3, P_1 \rangle$

Section-C

C. Attempt **all** the parts. (2 X 7.5 = 15)

11. Explain readers' writers' problem. Give the solution of this problem with the help of semaphores.

Answer:- An object is shared among several process.

a) Some read from it and they are known as reader processes.

b) Some read & write to it. They are known as writer processes.

There is no limit to how many readers can access the object simultaneously. But when a writer accesses the data, it needs exclusive access.

Shared data :-

```
semaphore mutex = 1
semaphore wrt = 1
int readcount = 0
```

write process :-

```
do {  wait (wrt);
            // writing is performed
      signal (wrt);
   } while (true);
```

Reader process :-

```
do{
    wait (mutex);
    readcount ++;
```

```
if (readcount ==1) wait (wrt);
signal (mutex);
  // reading is performed
wait (mutex);
  readcount--;
  if (readcount ==0) signal(wrt);
  signal (mutex);
} while (true);
```

12. Consider the following set of processes, with the length of the CPU burst given in milliseconds.

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5 all at time 0.

a) Draw two Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).

b) What is the turnaround time of each process for each of the scheduling algorithms in part a?

c) Which of the algorithms of part a results in the minimum average waiting time (over all processes)?

Answer:-

(a) Gantt chart for non-preemptive priority algo

P_2	P_5	P_1	P_3	P_4

0 1 6 16 18 19

Gantt chart for RR

P_1	P_2	P_3	P_4	P_5	P_1	P_3	P_5	P_1	P_5	P_1	P_5	P_1	P_5	P_1

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 19

(b) $\underline{WT}$ in $\underline{RR}$ $P_1 = (0-0)+(5-1)+(8-6)+(10-9)+(12-11)+(14-13)$

$= 0+4+2+1+1+1 = 9$

WT for $P_2 = (1-0) = 1$

WT for $P_3 = (2-0)+(6-3) = 2+3 = 5$

WT for $P_4 = (3-0) = 3$

WT for $P_5 = (4-0) + (7-5) + (9-8) + (11-10) + (13-12)$

$= 4 + 2 + 1 + 1 + 1 = 9$

$$\left.\begin{aligned} \text{TAT for } P_1 &= 9 + 10 = 19 \\ P_2 &= 1 + 1 = 2 \\ P_3 &= 2 + 5 = 7 \\ P_4 &= 1 + 3 = 4 \\ P_5 &= 5 + 9 = 14 \end{aligned}\right\} \text{For RR Algo.}$$

<u>For Priority Algorithm</u> :-

WT for $P_1 = 6$

$P_2 = 0$

$P_3 = 16$

$P_4 = 18$

$P_5 = 1$

TAT for $P_1 = 6 + 10 = 16$

$P_2 = 0 + 1 = 1$

$P_3 = 16 + 2 = 18$

$P_4 = 18 + 1 = 19$

$P_5 = 1 + 5 = 6$

(c) Average WT for priority $= \dfrac{6 + 0 + 16 + 18 + 1}{5}$

$= 41/5 = 8.2$

Average WT for RR $= \dfrac{9 + 1 + 5 + 3 + 9}{5}$

$= \dfrac{27}{5} = 5.4$

∴ RR algorithm shows the best result for min. Av. WT criteria.