Total Pages: 9

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Scheme for Valuation/Answer Key

Scheme of evaluation (marks in brackets) and answers of problems/key

FOURTH SEMESTER B.TECH DEGREE(R,S) EXAMINATION JUNE 2022(2019 Scheme)

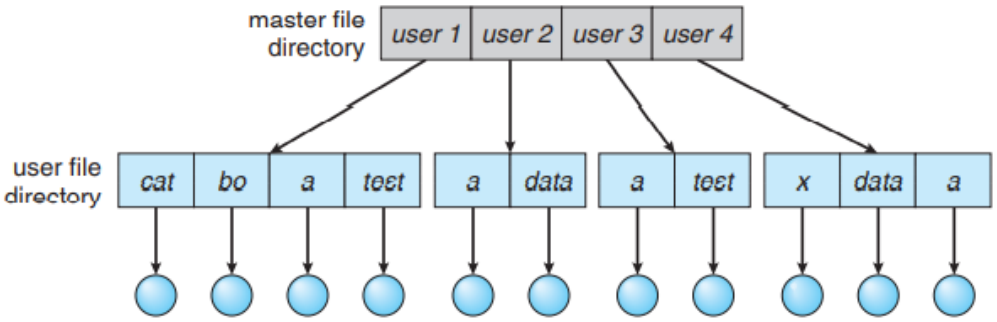
Course Code: CST206

Course Name: OPERATING SYSTEMS

Max. Marks: 100 | Duration: 3 Hours

PART A

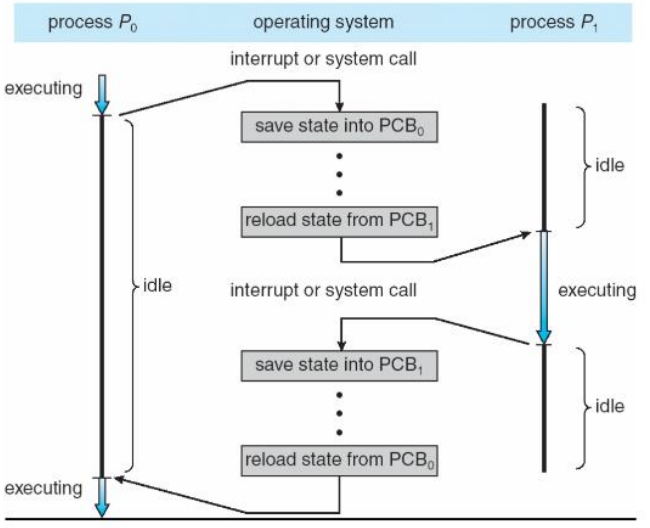
		(Answer all questions; each question carries 3 marks)	Marks
1		Bootstrap loader, located in ROM, can perform various tasks – run diagnostics to determine the state of the system, initializes all aspects of the system, locate the kernel in memory, loads it into memory and starts its execution.	3
2		Using dual mode of operations – user mode and kernel mode	3
3		3 child processes will be created and hence together with parent, it will print Forked 4 times.	3
4		Blocking send. Non blocking send, Blocking receive, Non blocking receive	3
5		When several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called a race condition. Race condition definition/explanation- 1.5 marks Example - 1.5 marks	3
6		Process termination , Resource preemption Listing - 1 mark explanation - 2 marks	3
7		If you know at compile time where the process will reside in memory, then **absolute** code can be generated at compile time - compile time binding If it is not known at compile time where the process will reside in memory, then the compiler must generate **relocatable** code. In this case, final binding is delayed until load time. - load time binding. 1.5 marks each	3
8		Logical memory size = 256 * 4KB = 2^20 B. Logical address has 20 bits.	

		Physical memory size = 64 * 4KB = 2^18 B. Physical address has 18 bits. Offset - depends upon page size. Hence 4KB = 2^12 B. So offset = 12 bits 1 mark each	
9		The seek time is the time for the disk arm to move the heads to the cylinder containing the desired sector. The rotational latency is the additional time for the disk to rotate the desired sector to the disk head. The disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer. 1 mark each	3
10		 diagram - 1 mark explanation - 2 marks **Note : As Directory structure is not mentioned explicitly in the syllabus, full credit may be given if attempted**	3
PART B			
(Answer one full question from each module, each question carries 14 marks)			
Module -1			
11	a)	System call explanation – 3 marks Explaining system calls using APIs and system call interface – 4 marks	7
	b)	Micro-kernel architecture – diagram – 2 marks Explanation – 4 marks Communication – through message passing – 1 mark	7
12	a)	Functions of OS – Process Management, Memory Management, Storage Management, Protection and security Each 3 marks	12
	b)	Increased throughput, Economy of scale, Increased reliability	2

Module -2

13 a) Context switching explanation - 3 marks 6

Diagram and explanation – 3 marks



b) Explain each scheduler – 2 marks each 8

14 a) 9

Process	Arrival Time(ms)	CPU Burst Time(ms)	Priority
P0	0	4	3
P1	2	5	2
P2	3	1	1
P3	4	3	4

i) Pre-emptive priority scheduling

Time: 0	2	3	4	8	10	13
P0	P1	P2	P1	P0	P3	

Avg. Turn around time = $\frac{(10-0)+(8-2)+(4-3)+(13-4)}{4}$

= 6.5ms

Avg. waiting time = $\frac{(8-2)+(4-3)+0+(10-4)}{4}$

= 3.25ms

ii) Non-preemptive priority scheduling

P0	P2	P1	P3

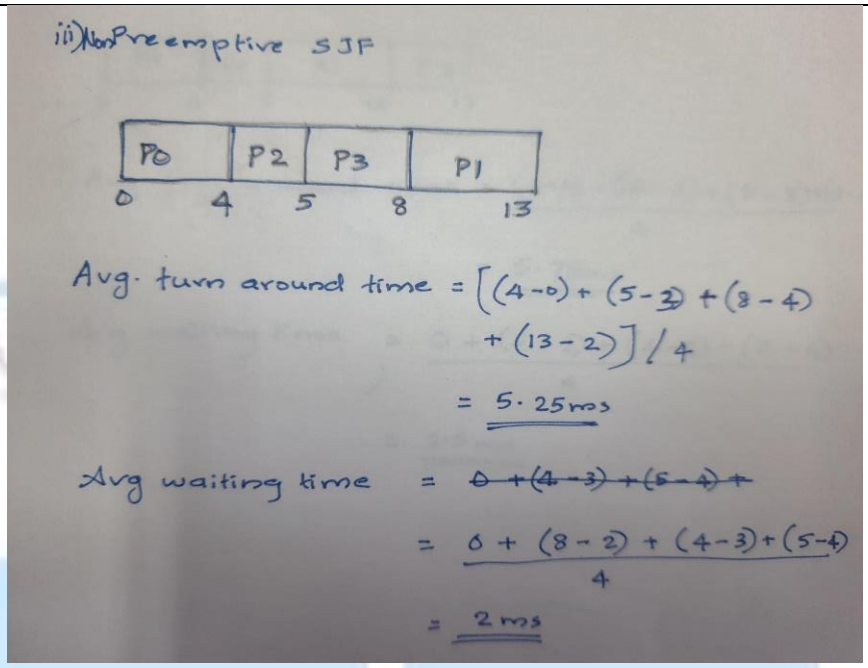
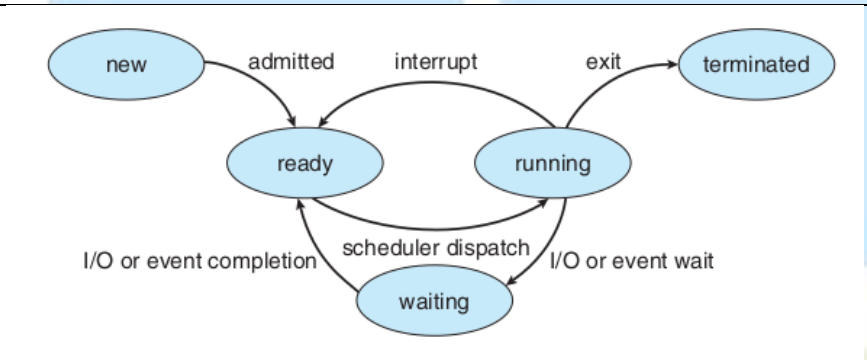
0 4 5 10 13

Avg. turn around time = $\frac{(4-0)+(10-2)+(5-3)+(13-4)}{4}$

= 5.75ms

Avg waiting time = $\frac{0+(5-2)+(4-3)+(10-4)}{4}$

= 2.5ms

		 iii) Non Preemptive SJF P0 | P2 | P3 | P1 0 4 5 8 13 Avg. turn around time = [(4-0) + (5-2) + (8-4) + (13-2)] / 4 = 5.25 ms Avg waiting time = ~~0 + (4-3) + (5-4) +~~ = 0 + (8-2) + (4-3) + (5-4) / 4 = 2 ms 3 marks each for correct chart and calculations	
	b)	 Diagram - 2 marks Explanation - 3 marks	5
		Module -3	
15	a)	Critical section - 1.5 marks 3 conditions and explanations - 4.5 marks	6
	b)	Deadlock detection - with cycle (single instance, multiple instance cases.) - 4 marks Deadlock avoidance - draw claim edges and describe cycles (single instance case) - 4 marks	8

16 a) 5

The structure of Philosopher *i*:

```
do {
    wait (chopstick[i] );
     wait (chopStick[ (i + 1) % 5] );

                //  eat

     signal (chopstick[i] );
     signal (chopstick[ (i + 1) % 5] );

                //  think

} while (TRUE);
```

Pseudocode - 3 marks

Deadlock and starvation possible with this code; - explanation 2 marks

b) 9

	Allocation	*Max*	*Available*	Need
	A B C D	*A B C D*	*A B C D*	A B C D
P_0	0 0 1 2	0 0 1 2	1 5 2 0	0 0 0 0
P_1	1 0 0 0	1 7 5 0		0 7 5 0
P_2	1 3 5 4	2 3 5 6		1 0 0 2
P_3	0 6 3 2	0 6 5 2		0 0 2 0
P_4	0 0 1 4	0 6 5 6		0 6 4 2

1. Work = 1 5 2 0 , Finish = F F F F F
2. Take P0. Need0 (0 0 0 0)<= Work.
   Work = 1 5 2 0 + 0 0 1 2 = 1 5 3 2,
   Finish = T F F F F

3. Take P3. Need3 (0 0 2 0) <= Work
   Work = 1 5 3 2 + 0 6 3 2 = 1, 11, 6, 4
   Finish = T F F T F

4. Take P2. Need2 (1 0 0 2) <= Work
   Work = 1, 11, 6 ,4 + 1 3 5 4 = 2, 14 ,11, 8
   Finish = T F T T F

5. Take P4 . Need4 (0 6 4 2) <= Work
   Work = 2, 14 ,11, 8 + 0 0 1 4 = 2, 14, 12, 12
   Finish = T F T T T

6. Take P1. Need1 (0 7 5 0) <= Work
   Work = 2, 14, 12, 12 + 1 0 0 0 = 3, 14, 12,12
   Finish = T T T T T

Safe sequence <P0, P3, P2, P4, P1>.
State is safe.

Checking for the safe state : 4 Marks

Note : More than one safe sequence are there. Mark can be given to any safe sequence.

State at time T0 is safe.
Request arrives : P1 <0,4,2,0>

1. Check whether Request(P1) <= Need1. <0,4,2,0> <= <0 7 5 0> . Request is valid.
2. Check whether Request(P1) <= Available. <0,4,2,0> <= <1 5 2 0>. Hence can try to allocate.
3. Pretend to allocate resources and arrive at the new state.
   State at time T1. (Pretend Resource is allocated)
   Allocation (P1) : <1 0 0 0> + <0 4 2 0> = <1 4 2 0>
   Need (P1) : <0 7 5 0> - <0 4 2 0> = < 0 3 3 0>
   Available : <1 5 2 0> - <0 4 2 0> = < 1 1 0 0>

<P0, P2, P3, P1, P4> is a safe sequence. Hence request can be granted.

Executing Resource request Algorithm and then checking for safe state : 5 marks

Module -4

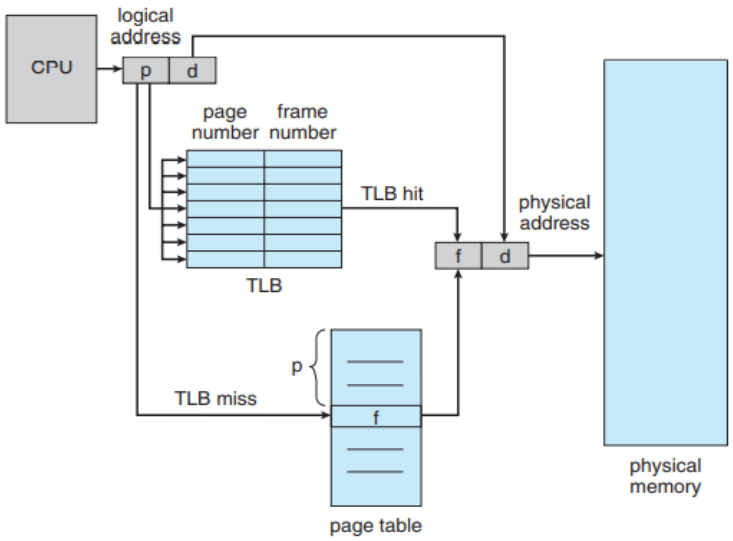
17	a)		9

	9	2	3	1	2	5	3	4	6	9	9	1	0	5	4	6	2	3	0	1	Page Faults
FIFO	**9**	9	9	**1**		1		1	**6**	6		6	**0**	0	0	**6**	6	6	**0**	0	17
		2	2	2		**5**		5	5	**9**		9	9	**5**	5	5	**2**	2	2	**1**	
			3	3		3		**4**	4	4		**1**	1	1	**4**	4	4	**3**	3	3	

	9	2	3	1	2	5	3	4	6	9	9	1	0	5	4	6	2	3	0	1	
Optimal	**9**	9	9	**1**		1		1	1	1			1		1	1	1	1			13
		2	2	2		**5**		5	5	5			5		**4**	**6**	**2**	**3**			
			3	3		3		**4**	**6**	**9**			**0**		0	0	0	0			

	9	2	3	1	2	5	3	4	6	9	9	1	0	5	4	6	2	3	0	1	
LRU	**9**	9	9	**1**		1	**3**	3	3	**9**		9	9	**5**	5	5	**2**	2	2	**1**	18
		2	2	2		2	2	**4**	4	4		**1**	1	1	**4**	4	4	**3**	3	3	
			3	3		**5**	5	5	**6**	6		6	**0**	0	0	**6**	6	6	**0**	0	

3 marks each

	b)	Pure paging - results in internal fragmentation. Pure segmentation - results in external fragmentation with explanation. 2.5 marks each.	5
18	a)	Virtual memory concept - 3 marks Demand paging explanation - 3 marks	6
	b)	 Diagram - 3 marks Explanation - TLB acts as cache, associative memory, address translation - 5 marks	8

Module -5

19	a)	Access methods - sequential , direct with explanations. 2 marks each	4
	b)	Linked allocation - 5 marks Indexed allocation - 5 marks. Explanations with required diagrams.	10
20	a)	3 marks each for explanation and answer. FCFS: 100 -> 20->89 -> 130 ->45 -> 120 -> 180 Head movement : 410 cylinders SSTF: 100 -> 89 -> 120 -> 130 -> 180 -> 45 ->20 Head movement: 262 cylinders CSCAN : Method-1 If direction of head movement towards 199 100 → 120 → 130 → 180 → 199 → 0 → 20 → 45 → 89 Total Head movement= 387 Cylinders Method-2 If direction of head movement towards 0 100 → 89 → 45 → 20 → 0 → 199 → 180 → 130 → 120 Total Head movement= 378 Cylinders Three Marks can be given to any one method	9
	b)	Owner, Group, Universe Explanation – 3 marks Example - 2 marks	5
		*********	