SASTRA ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956) THINK MERIT | THINK TRANSPARENCY | THINK SASTRA	**School of Computing** **Third CIA Exam –Feb 2024** Course Code: Course Name: Duration: 90 minutes Max Marks: 50

Answer Key

PART A

1. One benefit concerns program portability. Furthermore, actual system calls can often be more detailed and difficult to work with than the API available to an application programmer

2. Process state, PID, PC, Registers, Process priority, Pointers to scheduling queues, Base and limit registers, list of open files

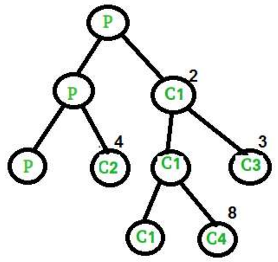
3. The long-term scheduler controls the degree of multiprogramming (the number of processes in memory). The more the number of processes in memory the better the chances of CPU getting a ready process for execution without becoming idle. It is important that the long-term scheduler make a careful selection of IO bound and CPU bound processes to keep the ready queue non-empty.

4. The parent waits for the child process to complete with the wait() system call. When the child process completes , the parent process resumes from the call to wait().The wait() system call is passed a parameter that allows the parent to obtain the exit status of the child. This system call also returns the process identifier of the terminated child so that the parent can tell which of its children has terminated. A process that has terminated, but whose parent has not yet called wait(), is known as a zombie process.

5. In shared memory, system calls are made only to create the sh memory and to attach the memory with the process. But no sy calls for sending and receiving messages. But in message qu send and receive are also system calls. So they cause r switches from user mode to kernel mode. Therefore, mes queue is better.

6. Ordinary pipes are unidirectional, allowing only one-way communication. If two-way communication is required, two pi must be used. On UNIX systems, ordinary pipes are constructe using the function
   pipe(int fd[])
   This function creates a pipe that is accessed through the int fd[descriptors: fd[0] is the read-end of the pipe, and fd[1] is the w end

7. Five processes created

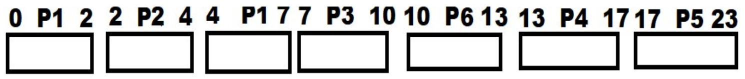


8. The next CPU burst is generally predicted as an exponential average of the measured lengths of previous CPU bursts. We c define the exponential average with the following formula

$$\tau_{n+1} = \alpha\, t_n + (1 - \alpha)\tau_n.$$

If **α** = 0, then **α**n+1 = **α**n, and recent history has no effect (curre conditions are assumed to be transient). If **α** = 1, then **α**n+1 = t and only the most recent CPU burst matters. More commonly, 1/2, so recent history and past history are equally weighted.

9. No of context switches: 7



10. Preemptive scheduling can result in race conditions when data are shared among several processes. Consider the case of two processes that share data. While one process is updating the data, it is preempted so that the second process can run. The second process then tries to read the data, which are in an inconsistent state. **Any example conveying the above meaning is acceptable**

Part B

11. SJF

Process	P1	P2	P3	P4	P5
CT	12	30	49	20	42
TT	10	27	44	14	34
WT	0	14	25	4	22

12. System calls, API, Types
System calls in fork program : fork, pipe, printf, scanf, sleep or wait

13. Multilevel queue scheduling

0-2 P1, 2-4 P2, 9-11 P3 11-13 P4, 13-15 P5					
4-5 P1, 5-9 P2, 15-19 P3 19-23 P4, 23-27 P5	P1(1)	P2(6)	P3(4)	P4(9)	P5(7)
27-29 P2, 29-34 P4,34-37 P5	P2(2)	P4(5)	P5(3)		

Process	P1	P2	P3	P4	P5
CT	5	29	19	34	37
TT	5	27	13	27	26
WT	2	19	7	15	17