SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing
DEPARTMENT OFCOMPUTING TECHNOLOGIES
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu
Academic Year: 2021-2022 (EVEN)

Test: CLAT-1 **Date: 4-4-2022**
Course Code & Title: 18CSC205J: Operating systems **Duration:** 1 Period
Year & Sem: II & IV **Max. Marks:** 25 Marks

Course Outcomes (CO):	*At the end of this course, learners will be able to:*
CO-1 :	*Express the fundamental concepts in operating system*

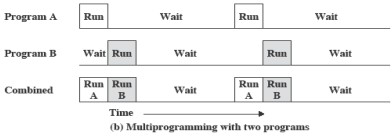
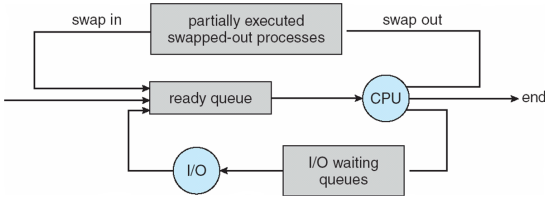
Program Outcomes (PO)												PSO		
1	2	3	4	5	6	7	8	9	10	11	12			
Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
3		*3*										*2*		

Part - A
(5 x 1 = 5 Marks) Instructions: Answer all

Q. No	Question	Marks	BL	CO	PO	PI Code
1	Time Sharing technique handles a) Single Interactive Job **b) Multiple Iterative Job** c) Recent Interactive Job d) Old Interactive Job	1	1	1	1	1.6.1
2	Interrupts make an Operating System more a) Secure b) Slow c) Fast **d) Flexible**	1	1	1	1	1.6.1
3	Work can be _____ while using multiprogramming batch processing. a) Rigid **b) Expensive** c) Reliable d) Flexible	1	1	1	1	1.6.1
4	When you start up the computer the boot up storage at which the BIOS versions manufacturer and data are displayed on the monitor is called a) Bootstrap **b). Power on self-test (POST)** c) System configuration d). Kernel loading	1	1	1	1	1.6.1
5	What is dispatch latency? **A. The time taken by the dispatcher to stop one process and start another** B. The time taken by the processor to write a file into disk C. The whole time taken by all processor D. None of Above	1	1	1	1	1.6.1

Part – B
(5 x 4 = 20 Marks) Instructions: Answer any 5

6	Compare Modes of operation in operating system.
	• **User Mode**
	  • **User program executes in user mode**
	  • **Certain areas of memory are protected from user access**
	  • **Certain instructions may not be executed**
	  •
	• **Kernel Mode**
	  • **Monitor executes in kernel mode**
	  • **Privileged instructions may be executed**
		4	4	1	2	2.6.4

	• **Protected areas of memory may be accessed**					
7	Write short notes on Multiprogramming Multiprogramming • also known as multitasking • memory is expanded to hold three, four, or more programs and switch among all of them  (b) Multiprogramming with two programs There must be enough memory to hold the OS (resident monitor) and one user program When one job needs to wait for I/O, the processor can switch to the other job, which is likely not waiting for I/O	**4**	**3**	**1**	**1**	**1.7.1**
8	Relate Process Control Block with respect to process. Information associated with each process (also called **task control block**) Process state – running, waiting, etc Program counter – location of instruction to next execute CPU registers – contents of all process-centric registers CPU scheduling information- priorities, scheduling queue pointers Memory-management information – memory allocated to the process Accounting information – CPU used, clock time elapsed since start, time limits I/O status information – I/O devices allocated to process, list of open files	**4**	**3**	**1**	**2**	**2.6.2**
9	With a neat sketch interpret addition of medium-term scheduling **Medium-term scheduler** can be added if degree of multiple programming needs to decrease ● Remove process from memory, store on disk, bring back in from disk to continue execution: **swapping**  	**4**	**3**	**1**	**3**	**2.6.2**
10	Illustrate inter-process communication **Processes within a system may be *independent* or *cooperating*** **Cooperating process can affect or be affected by other processes, including sharing data** **Reasons for cooperating processes:** ◦ **Information sharing** ◦ **Computation speedup** ◦ **Modularity** ◦ **Convenience** **Cooperating processes need interprocess communication (IPC)**	**4**	**2**	**1**	**1**	**1.7.1**

	Two models of IPC ◦ **Shared memory** ◦ **Message passing**					

11	With a suitable example illustrate IPC POSIX Producer
	`#include <stdio.h>`
	`#include <stdlib.h>`
	`#include <string.h>`
	`#include <fcntl.h>`
	`#include <sys/shm.h>`
	`#include <sys/stat.h>`
	
	`int main()`
	`{`
	`/* the size (in bytes) of shared memory ob`
	`const int SIZE = 4096;`
	`/* name of the shared memory object */`
	`const char *name = "OS";`
	`/* strings written to shared memory */`
	`const char *message_0 = "Hello";`
	`const char *message_1 = "World!";`
	
	`/* shared memory file descriptor */`
	`int shm_fd;`
	`/* pointer to shared memory obect */`
	`void *ptr;`
	
	`/* create the shared memory object */`
	`shm_fd = shm_open(name, O_CREAT | O_RDWR`
	
	`/* configure the size of the shared mem`
	`ftruncate(shm_fd, SIZE);`
	
	`/* memory map the shared memory object`
	`ptr = mmap(0, SIZE, PROT_WRITE, MAP_SHA`
	
	`/* write to the shared memory object */`
	`sprintf(ptr,"%s",message_0);`
	`ptr += strlen(message_0);`
	`sprintf(ptr,"%s",message_1);`
	`ptr += strlen(message_1);`
	
	`return 0;`
	`}`
		4	**2**	**1**	**2**	**1.7.1**