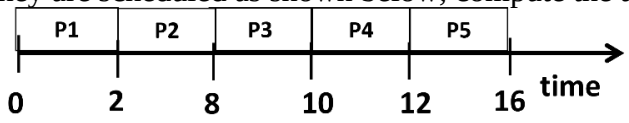
RV UNIVERSITY Go, change the world an initiative of RV EDUCATIONAL INSTITUTIONS **School of Computer Science & Engineering** **B.Tech(H) Program**	**MID-TERM: ANSWER SCHEME** ✅ **CIE** **Academic Year: 2023-24** **Term: II** **Semester: Sem 3** **Section: A, B and C** **Date: 18 Oct 23 (9:15 am to 11:15 am)** **Course Code: CS2100** **Course Name: Operating Systems and Systems Software** **Max Marks: 30 (Weightage: 20 Marks)**

Instructions to the students:1. Answer all the questions in Part A. **2.** Part-B any 10 out of 12 questions can be answered. The best 10 answers will be considered for evaluation. **3.** No further clarifications will be provided during the examination. You can write all your assumptions in your answers, if any.

Q.No	Short descriptive answers: Part A (10 Questions * 1 mark) – 10 Marks	Marks
1	Briefly explain why there needs to be at least two modes (user and kernel) of operations to be supported by a CPU.	**1**
2	Give one possible reason for a process which is currently running in the CPU to get moved into the Blocked Queue.	**1**
3	Assume that all the below five processes have become ready at time zero. If they are scheduled as shown below, compute the average waiting time.  	**1**
4	From the above Gantt chart, can you guess the scheduler that was used to pick these processes for scheduling? **Note**: You can assume lower process IDs are added first into the Ready queue when two processes arrive at the same time. How many processes would have been in the Ready Queue at time unit 9?	**1**
5	Assume you have a preemptive scheduler running with a quantum time of 4 units. If all the processes added to the system are having execution times lower than four time units, the number of context switches will be same as the number of processes that have come into the system. Is the above statement true or false? Justify your answer.	**1**

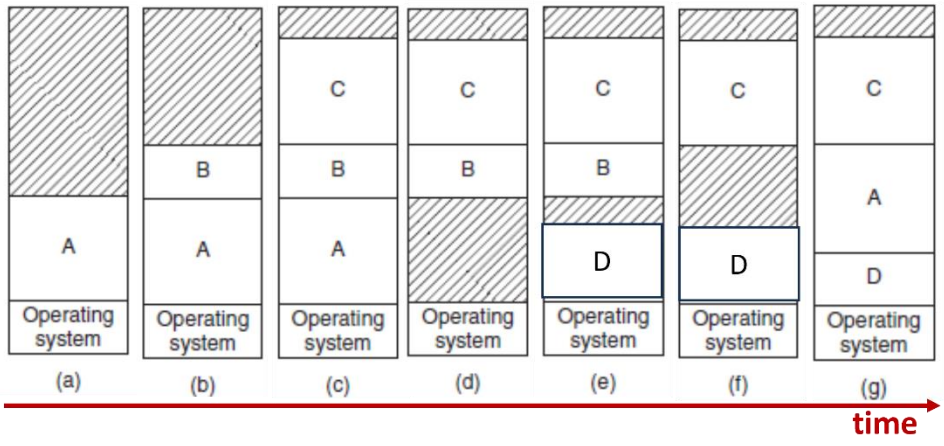
6	Assuming all the processes have completed their execution at time unit 19. Give the longest and the shortest execution times and their process IDs. 	**1**
7	What is the major difference between **pthread_exit()** and **exit()** system calls?	**1**

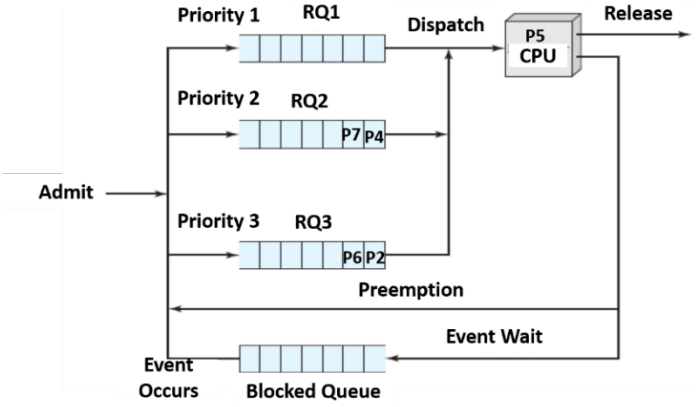
8	Give the messages printed by the parent and child processes when the below code executes. You can assume that it works correctly.
	if(fork() == 0)
	  printf("Message1\n");
	else
	  printf("Message2\n");
	
	printf("Message3\n");
		1

9	Assume that the thread with the **id** has executed **pthread_exit("Done\n")** What happens when another thread executes **pthread_join(id, &status)**? **Note**: The **id** here refers to the thread which has executed **pthread_exit().**	**1**
10	Explain in words what does this **datatype** declaration mean. **float *(*(*myFptr)(int, int))(float, float);** **Note**: Bonus 2 marks will be given if you can write a code which uses the above declaration in a program, that compiles correctly and runs.	**1**

Part B is on the next Page

RV UNIVERSITY Go, change the world an initiative of RV EDUCATIONAL INSTITUTIONS **School of Computer Science & Engineering** **B.Tech(H) Program**	**MID-TERM: ANSWER SCHEME** ✅ **CIE** **Academic Year: 2023-24** **Term: II** **Semester: Sem 3** **Section: A, B and C** **Date: 18 Oct 23 (9:15 am to 11:15 am)** **Course Code: CS2100** **Course Name: Operating Systems and Systems Software** **Max Marks: 30 (Weightage: 20 Marks)**

Q. No.	Problems and long descriptive answers: Part B Answer any 10 out of 12 (10 Questions * 2 marks) – 20 Marks	Marks
11	Draw the Gantt chart if the SRTF scheduler is running in the system. **Process Id** / **Arrival time** / **Burst time** P1 / 0 / 9 P2 / 1 / 4 P3 / 2 / 9	**2**
12	In a memory hierarchy, if the access time of a memory which is closer to the CPU is designed to be lower than the memory which is farther away from the CPU, how does it improve the performance of the system?	**2**
13	If the below snapshots have not missed out any movements of processes in and out of main memory, answer the following questions: a) Did process A move out of MM at any time and why? b) Give one valid reason for the process D's size going down. 	**2**
14	Explain any two responsibilities of memory management system.	**2**

15	Explain address translation done by the MMU using Relocation register with an example.	**2**
16	Ready queues of three priority levels with processes in them are shown below. Within the same priority level, LIFO is the scheduling used. Assume that the currently running process P5 creates two new processes P8 followed by P9 both with Priority 1. Assume, a non-preemptive scheduler is running in the system. Assume Priority 1 is the highest, with Priority 1 > Priority 2 > Priority 3 Which is the next process that would run on the CPU after P5 terminates? 	**2**
17	a) Give all the commands that you would give to make the below shell script run. Assume that this file is in the directory **/home/pi/chi/script** b) When it is run, what will be the output on the terminal? **Contents of the file: Q17.sh** (assume that it does not have any execute permissions now) **#!/bin/bash** **echo "This is a shell script, in Q17 of midterm exam!!!"** **cd ../..** **pwd**	**2**
18	Explain briefly about stdin, stdout and stderr. What are their file descriptor values?	**2**

19	Explain what happens when the below function is successfully executed. **pthread_create(&thread_id, NULL, thread_function, (void *)message);** **Note:** Explain about the purpose of each parameter.	**2**

20	When the following code executes, what will be printed?
	Note: You can assume that always the parent process gets a chance to run before the child processes, after any **fork()** call.
	printf("Start of Q20\n");
	fork();
	printf("After the first fork()\n");
	fork();
	printf("After the second fork()\n");
	fork();
	printf("After the third fork()\n");
		2

21	Explain along with msgs output by this program.
	//All essential include files are assumed to be here.
	void *threadFn(void *arg);
	int main(){
	  void *thread_result;
	  pthread_t id;
	  pthread_create(&id, NULL, threadFn, NULL);
	  pthread_join(id, &thread_result);
	  printf("I am here ...\n");
	  while(1);
	}
	void * threadFn(void *arg){
	  printf("Thread created.\n");
	  sleep(2);
	  pthread_exit(NULL);
	}
		2

22	Explain what happens when the below Q22.sh is executed. #!/bin/bash echo "Add two numbers passed to the script" num1=$1 num2=$2 echo "Values in num1 and num2 are $num1 and $num2" let sum=($num1 + $num2) echo "The sum of the numbers passed is $sum"	**2**

******

RV UNIVERSITY Go, change the world an initiative of RV EDUCATIONAL INSTITUTIONS School of Computer Science & Engineering B.Tech(H) Program	**MID-TERM: ANSWER SCHEME** ✅CIE Academic Year: 2023-24 Term: II Semester: Sem 3 Section: A, B and C Date: 18 Oct 23 (9:15 am to 11:15 am) Course Code: CS2100 Course Name: Operating Systems and Systems Software Max Marks: 30 (Weightage: 20 Marks)

Answer Keys:

Q1. User mode is for running user processes and more secure kernel mode to run the OS or kernel code.

Q2. Running State to Blocked Q

While a process is running on the CPU, the process might invoke a system call, to open a file or to receive inputs from the User, which will make the OS to move that process to Blocked state and adding it to the Blocked Queue.

Q3. Avg waiting time = (0 + 2 + 8+ 10 + 12) / 5 = 32/5 = 6.4 time units

Q4. FIFO (first in first out), because the processes are running in the order of lower process IDs to higher IDs. – 0.5 marks

At time unit 9, there would have been two (P4 and P5) processes in the Ready queue.

Q5. True: Since all the processes entering the system are of lower execution times than 4 time units, before the quantum time kicks in, the processes would have terminated and the new process would have got scheduled. So, the number of context switches would be same as the total number of processes in the system.

False: Only if the explanation says that it is false because the first process loading is not considered to be a context switch and the number of context switches are one less than the number of processes in the system – give full marks.

Q6. Shortest process P1: 1 time unit and the longest process P3: 11 time units.

Q7. pthread_exit() will allow other threads belonging to the family of this thread, including its parent to continue to run, while the thread calling this will be terminated. Whereas, exit() will not only terminate the thread calling it, but also the parent and the other threads belonging to the same family.

Q8. Parent process prints the following messages:

Message2

Message3

Child process prints the following messages:

Message1

Message3

Note: If Message3 is not mentioned, and other two messages are given correctly, give 0.5 marks

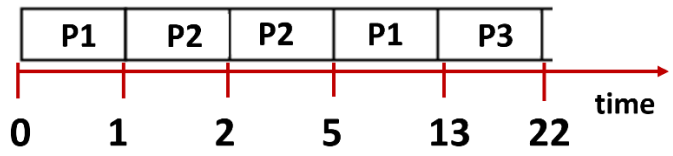
Q9. When another thread executes pthread_join(), if the thread with id has already terminated it will not be blocked, it will be given the pointer to "Done\n" passed by the pthread_exit(), if not, the thread will be blocked until pthread_exit() is done by the thread with the result.

Q10. myFptr is function pointer to a function which accepts two integers and returns another function pointer which accepts two float values and returns a float ptr.

Sample code will be provided.

Part B

Q11. If the SRTF scheduler is running the Gantt chart is as below:



Q12. Since the CPU will be accessing the memory which is closer to the CPU more often than the one which is farther away from the CPU, the average time taken for the overall memory access will be much lower because most of the time CPU will be accessing the memory which is closer to it, thus improving the performance of the system.

Q13.

a) Did process A move out of MM at any time and why?

Yes, it went out to create space for the Process D which is in Ready state and the Process A would have been in a blocked state or of lower priority.

b) Give one valid reason for the process D's size going down.

May be because of its stack or heap usage going down. Stack size may go down because of depth of function invocations getting reduced because of control returning from the function calls.
Note: If the reason is given as to bring in process A, reduce 0.5 marks, because another process area cannot be arbitrarily cut down to accommodate another process.

Q14. Ref Session 6B that explains about the responsibilities or solutions provided by the Memory management system of the OS.

Q15. Ref. Session 6C.

Q16. ANS P9:
Short explanation is required justifying your answer.
The next Process that would run after P5 terminates is: **P9** (this would be the process at the head of the highest priority ready queue (Priority 1) Since P5 is creating P8 with Priority 1 first, it will be in the head of Priority 1 Queue and later P9 will be added ahead of P8 in the same Priority 1 queue because of LIFO) So, P9 will be picked up. **Note**: Since it is a non-preemptive scheduler, no time-slice need to be assumed.

Q17. a) To make the script file run, the following commands need to be given: - 1 mark
chmod +744 Q17.sh or chmod +x Q17.sh (or any other valid octal inputs making it executable)
./Q17.sh

b) When this shell script runs, the output will be: - 1 mark
This is a shell script, in Q17 of midterm exam!!!
/home/pi

Q18. The stdin , stdout , and stderr are standard streams for input, output, and error output. By default, standard input is read from the keyboard, while standard output and standard error are printed to the screen.

Every program running in Linux, is associated with stdin to receive a stream of inputs.

stdout: Is the standard output where the printf messages go to. stderr, is the error messages from the program is routed to. The file descriptors of each are 0, 1 and 2.

Q19. pthread_create(&thread_id, NULL, thread_function, (void *)message);

It creates a thread by giving the entry function as thread_function that accepts a void * as the input parameter. It writes the thread id of the newly created thread, into the pointer passed as the first parameter. The second parameter is the attribute of the thread created, if it is passed as NULL, a thread is created with the default parameters.

Q20. Order can be arbitrary, the number of messages need to be correct. Relatively reduce marks based on the correctness.

Start of Q20 – Only once
After the first fork() – twice
After the second fork()\n – four times
After the third fork() – eight times

These two questions may be explained in the class, if required.

Q21. Main thread creates one child thread which prints a message and sleeps for 2 seconds. The main thread after creating the child thread, gets blocked for the child thread to call pthread_exit(). When the child thread wakes up after 2 seconds and calls pthread_exit(), the child thread will get terminated. The main thread will be woken up and it prints the message and runs for ever.

Note: The child thread does not pass any valid message to the main thread on exit.

Q22. It is a shell script file which receives two numbers as parameters and it adds both of them and prints the result.

****