National University of Computer and Emerging Sciences, Lahore Campus

Course:	Operating Systems	Course Code:	CS 2006
Program:	BS(Computer Science)	Semester:	Spring 2022
Date:	08-June-2022	Total Marks:	70
Exam Type:	Final	Page(s):	9
Time:	180 minutes	Weightage:	45
Sections:	ALL		

Roll Number:_____ **Section:**_____

Instructions:

1. Attempt all questions
2. You may use extra sheets for working, but do not attach them.
3. Write the final answer in the space provided for it.
4. Exchange of calculators or other items is not allowed.

Question	Marks	Marks Obtained	CLO
1	10		CLO1, CLO2, CLO3, CLO4
2	15		CLO1, CLO2, CLO3, CLO4
3	10		CLO3
4	10		CLO4
5	10		CLO2
6	10		CLO3
7	5		CLO4, CLO5

Question 1: Choose the correct option. [10 marks]

1. Which of the following is true for many-to-one thread mapping?
   a. It requires fewer resources in the kernel space.
   b. It does not block other threads of the process if one thread in the same process makes a blocking call.
   c. It can provide multithreading in an operating system that has no native support for multithreading.
   d. All above statements are true
   e. Only a and c are true

2. Which of the following is false about threads?
   a. Threads share instructions.
   b. Threads share stack.
   c. Threads share global variables.
   d. Threads share process id.

3. On a system that has a single CPU, which of the following scenarios may cause the ready queue to be empty?
   a. All processes are I/O bound.
   b. All processes are CPU bound.
   c. 50% of the processes are CPU bound. The remaining processes are I/O bound.
   d. Both a and b

4. Which of the following statements is true?
   a. Keeping time quantum large in round robin scheduling can make the scheduling more like first-come-first-served algorithm.
   b. Keeping the time quantum small will decrease the number of context switches.
   c. Keeping the time quantum large will decrease the number of context switches.
   d. Both a and b
   e. Both a and c

5. Which of the following statements is true?
   a. The size of Logical address space is always equal to the size of physical address space.
   b. The size of logical address space may be larger than the size of physical address space.
   c. The size of logical address space is always less than the size of physical address space.
   d. All statements are false.

6. The time it takes for a process to begin executing for the first time is known as:
   a. turnaround time
   b. waiting time
   c. dispatch latency
   d. response time

7. Which of the following system calls is a blocking call?
   a. read
   b. write
   c. pipe

d. none of these

8. Copy-on-write is a technique in which:
   a. parent and child processes share all pages. Any changes in a page by one process are visible to other process.
   b. parent and child processes don't share any page.
   c. parent and child processes initially share pages. However, any changes in a page cause the page to be copied and placed in separate memory, i.e., the modified page is no longer shared.
   d. parent and child share all pages. Any changes in a page are not allowed.
   e. none of the above

9. A zombie process is a process that has terminated and:
   a. its parent process has also terminated without applying the wait system call.
   b. the parent process is currently executing without applying the wait system call.
   c. It was created without any parent.
   d. none of these.

10. In ordinary pipes, communication between parent and child is possible because:
   a. the pipe's buffer is part of address space of both child and parent
   b. child inherits parent's file descriptor table
   c. the data is read from and written to a file present on hard disk.
   d. none of these

Question 2: Give short and precise answers to following questions: [15 marks]

1. Give two benefits of dynamically linked libraries. [2 + 2= 4 marks]

   a. It reduces the size of executable.
   b. It saves memory space since multiple processes can use the same memory frame(s) where the library code is placed.
   c. It can allow the process to use the latest version of the library.

2. What is the difference between short-term scheduler (CPU-scheduler) and long-term scheduler (job scheduler)? Which one of these two schedulers runs more often? [2 + 1= 3 marks]

   a. Short term scheduler selects one of the processes from the ready queue for execution.
   b. Long term scheduler decides which process to remove from the memory (and hence from the ready queue) to limit the number of processes that are executing concurrently.

   Short term scheduler runs more often.

3. What is meant by bounded wait in process synchronization? [2 marks]

Bounded wait means that a process must not wait infinitely before entering the critical section.

4. What is task parallelism? What is data parallelism? Give one example of both. [1+1+1+1=4 marks]

Task parallelism involves distributing task (threads) across multiple computing cores. Each thread is performing a unique operation. Different threads may be operating on the same data, or they may be operating on different data.

Data parallelism focuses on distributing subsets of the same data across multiple computing cores and performing the same operation (but on a different subset of data) on each core.

Example of Task parallelism: performing addition, subtraction and multiplication (on same data or on different data).

Example of Data Parallelism: Search on first half of the array and on the second half of the array using two threads.

5. Suppose we have an operating system that has two versions of mutex implementation. One implementation allows busy wait whilst the other implementation does not allow busy wait, i.e., the process that could not acquire the lock is put to sleep until the other process releases the lock. Now Under what circumstance, is it desirable to use the version of mutex that allows busy wait? [2 marks]

Mutex with busy wait is desirable in the case if the other process will use the critical section for a very short time.

Question 3: Consider the following processes, their arrival and CPU burst times. [5 + 5= 10 marks]

Process	Arrival Time	CPU Burst Time
P1	0	9
P2	1	7
P3	2	4

a. Draw Gantt chart of the above set of processes by applying shortest-remaining-time-first schedule.

b. What is the waiting time of each process?

P1=0+11=11
P2=0+4=4
P3=0

Question 4: Suppose that a machine has 48-bit virtual address and 32-bit physical address, and the memory is byte-addressable, i.e., each byte can be accessed individually using its physical address. The size of a page is 4KB, i.e. 4*1024 bytes. Now calculate the following: [10 marks]

1. Number of bits required for the page offset. [2 marks]

$log_2(4 * 1024) = 12$

2. Number of bits required for the physical page (also called frame) offset. [2 marks]

Same as page offset.

3. Number of bits required for the page number. [2 marks]

48-12=36

4. Number of bits required for the frame number. [2 marks]

32-12=20

5. Total number of pages that a process can have. [1 mark]

2^36=68719476736

6. The size of physical memory in MBs. Assume 1KB=1024B and 1MB=1024KB. [1 mark]

2^32= 4294967296B =4194304KB= 4096MB

Question 5: There is an executable program named "sort.out" that, whenever executed, asks a list of numbers from the user via the keyboard. The executable program then sorts the list and prints the numbers in sorted order. Your task is to write another program that calls the executable program sort.out in such a way that sort.out gets the list of numbers from a file nums.txt rather than from the user via keyboard. Also, after sorting, the sort.out program will print the numbers to another file sorted_nums.txt rather than on screen. After sort.out has finished execution, your program will print "Task Completed!". Write the program using C/C++ syntax. There is no need to write code to include the required libraries. [10 marks]

```
int main()
{
  int pid = fork();
  if (pid == 0)
  {
    int readFd = open("nums.txt", O_RDONLY, 0);
    int writeFd = open("sorted_nums.txt", O_WRONLY | O_CREAT, 0666);
    dup2(readFd, 0);
    dup2(writeFd, 1);
    close(readFd);
    close(writeFd);
    execlp("sort.out", "sort.out", NULL);
    cout << "execlp error" << endl;
  }
  else if (pid > 0)
  {
     wait(NULL);
     cout << "Task completed!" << endl;
  }
  else
    cout << "fork error" << endl;
}
```

Question 6: Suppose we have a gym in Lahore that allows men as well women. However, to enforce gender separation, the gym has the following policy: When a woman is in the gym, other women may enter the gym, but no man can enter the gym. Similarly, when a man is in the gym, other men may enter the gym, but no woman can enter the gym. Your task is to enforce this rule using semaphores. There must not be any busy wait in your solution. Give your solution in the form of pseudocode.[10 marks]

```
//Create any shared variables here.
semaphore w=1, m=1,entry=1; //binary semaphores
int noOfMen=0, noOfWomen=0;
```

Woman (process)	Man (process)

wait(w);
noOfWomen++;
if (noOfWomen==1)
  wait(entry);
signal(w);
 	
Use Gym() [Critical Section]
 	
wait(w);
noOfWomen--;
if (noOfWomen==0)
  signal(entry);
signal(w);
	wait(m);
	noOfMen++;
	if (noOfMen==1)
	  wait(entry);
	signal(m);
	
	Use Gym() [Critical Section]
	
	wait(m);
	noOfMen--;
	if (noOfMen==0)
	  signal(entry);
	signal(m);

Question 7: Show execution of the optimal page replacement algorithm on the following page reference string: [5 marks]

2 3 1 4 3 5 1 2 1 4

Assume there are only three frames in the RAM. Show contents of memory frames after each page access from the reference string. (Please note that the number of boxes below maybe less or more depending upon the question. It certainly does not mean you have to utilize exactly the given number of boxes.)

2	2	2	4	4	4	4	4	4	4														
	3	3	3	3	5	5	2	2	2														
		1	1	1	1	1	1	1	1														