National University of Computer & Emerging Sciences, Karachi
Fall -2021 School of Computing
Mid-2 Examination
26th November 2021, 08:30 am – 09:30 am

Course Code: CS2006	**Course Name: Operating Systems**
Instructor Names: Dr. Ghufran Ahmed, Anaum Hamid and Safia Baloch	
Student Roll No:	**Section No:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **3 questions** on **2 pages.**
- All the answers must be solved according to the sequence given in the question paper.

Time: 60 minutes. **Max Marks: 40**

Question1: (Estimated time = 15 min) **[Marks:5+5+4=14]**

a) Determine if the following problems exhibit task or data parallelism:

I. Search of a text file for a string **S**.
Data parallelism

II. Find the product of the integers from 5 to 500 and the average of the integers from 5 to 500. Task parallelism

III. Find the dot product of two different matrices.
Task parallelism

IV. Convert the first character of each string in an array to uppercase. Data parallelism

V. Sum of integers from 1 to n.
Data parallelism

b) Consider a program where ONLY 20% of its execution is serial. Given that the performance of the serial portion is directly proportional to memory access latencies. How much speedup we get if we upgrade the computing unit from **4 cores to twice**.

4 Cores: 2.5x.

8 Cores: 3.2x

c) Suppose two threads execute the following C code concurrently, accessing shared variables a, b, and c.

Initialization

```
int a = 4;
int b = 0;
int c = 0;
```

Thread 1	**Thread 2**
`if (a < 0) {`	`b = 10;`
`  c = b - a;`	`a = -3;`
`} else {`	
`  c = b + a;`	
`}`	

What are the possible values for 'c' after both threads complete? You can assume that reads and writes of the given variables are atomic, and that the order of statements within each thread is preserved in the code generated by the C compiler.

Answer: 4, 7, 13, 14, -3

Question2: (Estimated time = 20 min) **[Marks: 5+5=10]**

a. Ping and pong are two separate threads executing their respective procedures. The code below is intended to cause them to forever take turns, alternately printing “ping” and “pong” to the screen. The minithread_stop() blocks the calling thread, and the minithread_start () makes a specific thread runnable if that thread has previously been stopped, otherwise its behavior is unpredictable.

```
void ping()
{
      while(true) {
            minithread_stop();
            Printf(“ping is here\n”);
            minithread_start(pongthread);
      }
}
void pong()
{
      while(true) {
            Printf(“pong is here\n”);
            minithread_start(pingthread);
            minithread_stop();
      }
}
```

The code shown above exhibits a wellknown synchronization flaw. Show how to fix the problem by replacing the minithread_stop and start calls with semaphore P and V operations.

```
Sema sping, spong = sema_init(0);
void ping()
{
      while(true) {
            P(sping);
            Printf("ping is here\n");
            V(spong);
      }
}
void pong()
{
      while(true) {
            Printf("pong is here\n");
            V(sping);
            P(spong);
      }
}
```

b. Agents from four terminals of Pakistan Stock Exchange create processes concurrently. These concurrent processes make changes in the value of dollars by modifying the global shared variable: **worth**. Processes created by agents AX and BX read **worth** from memory, increment the value by 20K, store it back to memory and then terminate. Processes created by agents CX and DX read **worth** from memory, decrement the value by 40K, store it back to memory and then terminate.

Synchronize and design the strategy to get the minimum possible value of **worth** after all processes complete their execution?

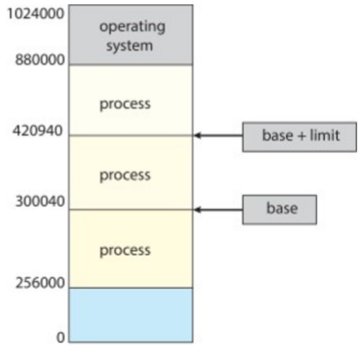
/*Answer Same as Shared Class Example*/

Question 3: (Estimated time = 20 min) **[Marks:5+5+6=16]**

a) Considering the memory protection mechanism, how an operating system classifies an instruction as legal or trap. Describe in detail about the security mechanism of a process loaded in memory. Illustrate your answer with diagram.

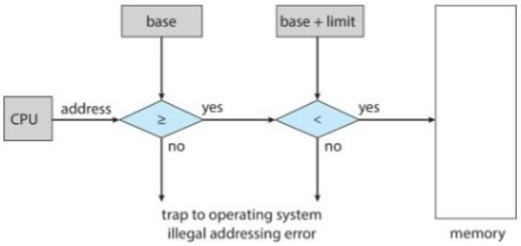
Base and Limit Registers

- A pair of **base** and **limit registers** define the logical address space
- CPU must check every memory access generated in user mode to be sure it is between base and limit for that user.
- Limit register = 120900



Hardware Address Protection

- CPU must check every memory access generated in user mode to be sure it is between base and limit for that user



- the instructions to loading the base and limit registers are privileged

b) Suppose a user process of 2048KB is rolled-out from main memory to hard disk with a standard transfer rate of 1Mbps. Calculate how long will it take if another user process of 1024KB rolls-in to the main memory.

User process size is 2048Kb

Data transfer rate is 1Mbps = 1024 kbps

Roll out Time = process size / transfer rate

= 2048 / 1024

= 2 seconds

= 2000 milliseconds

Roll in time = process size / transfer rate

= 1024 / 1024

= 1 seconds

= 1000 milliseconds

Total time = 2000 + 1000 = 3000 milliseconds / 3 sec.

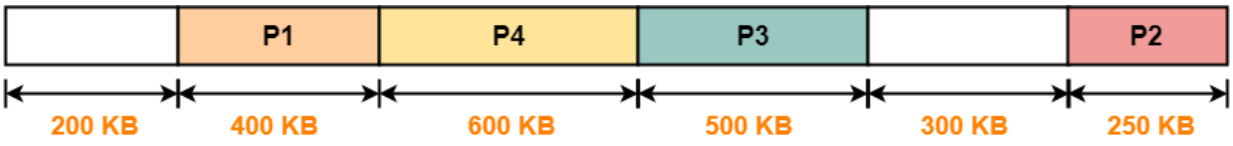
c) Consider six contiguous memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB and 250 KB. These partitions need to be allocated to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order. Suggest which placement algorithm will work best for the above scenario.

First Fit

Main Memory

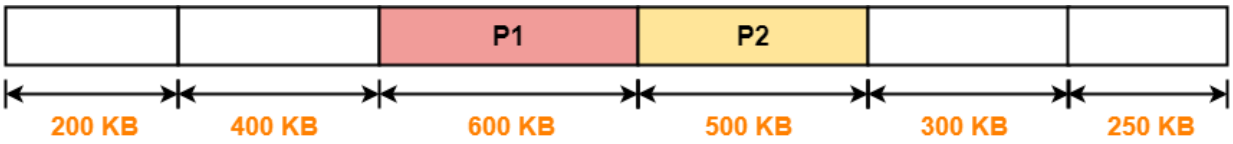
- Process P4 can not be allocated the memory.

Best Fit



Main Memory

Worst Fit



Main Memory

- Process P3 and Process P4 cannot be allocated the memory.

**********Best of Luck**********