

Name _____

All questions are worth 10 points. Maximum score: 70

1. Suppose the operating system does a context switch between process A and process B. Explain why this might occur, and give the steps that must occur during the context switch.

Interrupt occurs

Save Registers

Change Status of Process A

Service Interrupt

Schedule Process

Change Status of Process B

Load Registers

2. What is memory protection, and why is it needed?

Certain areas of the memory are inaccessible to ordinary processes. This is primarily to protect the operating system from attacks (deliberate or not) from user processes. It also protects processes from each other.

3. Analyze the read and write sets of the following two processes. Is there a potential problem? If so, what is it, and how might I solve the problem?

```
shared int Count=0;
shared Thing Head =NULL, Tail=NULL;
```

<pre>Manager Thing *NewItem = new Thing; cin>>NewItem->Price; if (Head == NULL) { Head = NewItem; } else { Tail->Next = NewItem; } Tail=NewItem; Count++;</pre>	<pre>User Thing *Item = Head; for (int i=0 ; i<Count ; i++) { Total = Item->Price; Item = Item->Next; }</pre>
---	--

R: Head, Tail, Count

W: Head, Tail, Count

R:Head, Count

W: NULL

There is a potential for the User process to see an inconsistent state because of the intersection between the manager write set and the user read set. To fix this create a shared semaphore mutex, and add `mutex.wait()` at the beginning of each process, and `mutex.signal()` at the end.

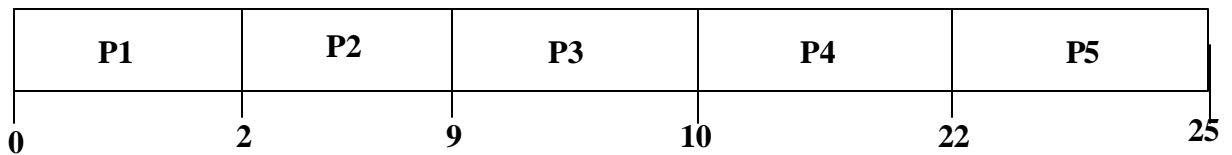
4. Why does shortest job first give a smaller average wait time than first come first served?

Processes that are serviced first have their burst times added to the wait times of all succeeding processes. Putting the short processes first causes the other processes to accumulate wait time at a slower rate.

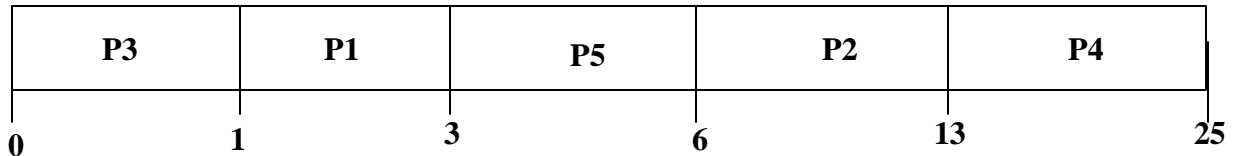
5. The following five processes arrive at time zero in the order given. For these processes, compute a) the average wait time, and b) the average turn-around time, for the two algorithms Shortest Job First, First Come First Served. Show a Gantt chart for each algorithm.

Process	Burst Time
P1	2
P2	7
P3	1
P4	12
P5	3

FCFS



SJF



	FCFS		SJF	
	Wait	Turnaround	Wait	Turnaround
P1	0	2	1	3
P2	2	9	6	13
P3	9	10	0	1
P4	10	22	13	25
P5	22	25	3	6
Total	43	68	23	48
Average	8.6	13.6	4.6	9.6

6. We have two processes operating on a shared stack. One pushes and one pops. Do the shared variables need protection? Why? If so, use TestAndSet to protect the shared variables.

Process 1(Push)

```
while(TestAndSet(X));  
PushItem->Next=Head;  
Head = PushItem;  
X=false;
```

Process 2 (Pop)

```
while(TestAndSet(X));  
if (Head != NULL)  
{  
    PopItem = Head;  
    Head = Head->Next;  
}  
X=false;
```

This is needed because Head is in the write-set of both processes.

7. Show how to solve the following problem using semaphores. We want to model a line at the bank. There is one teller process and multiple customer processes. (Show only one customer process.) A customer process will enter the bank. If the line already has 10 people, the customer will leave. Otherwise, the customer will wait in line until the teller calls for him/her. If the line contains no customers, the teller will go to sleep until a customer arrives. **Don't** make this problem more complicated than necessary. **Do** use actual C++ code and realistic-looking function calls for your semaphores.

```
Teller:
while(true)
{
    Teller.Wait();
    Line.Signal();
    Service Customer
}
```

```
Customer:
while (true)
{
    Mutex.Wait();
    if (CC>=10)
    {
        Mutex.Signal();
        Exit();
    }
    CC++;
    Mutex.Signal();
    Teller.Signal();
    Line.Wait();
    Be Serviced
    Mutex.Wait();
    CC--;
    Mutex.Signal();
}
```