

CSI 4337

Exam #1
50 min.

Feb. 8, 2007

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.

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

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;
```

Manager

```
Thing *NewItem = new Thing;
cin>>NewItem->Price;
if (Head == NULL)
{
    Head = NewItem;
}
else
{
    Tail->Next = NewItem;
}
Tail=NewItem;
Count++;
```

User

```
Thing *Item = Head;
for (int i=0 ; i<Count ; i++)
{
    Total = Item->Price;
    Item = Item->Next;
}
```

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

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 |

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)

```
PushItem->Next=Head;  
Head = PushItem;
```

Process 2 (Pop)

```
if (Head != NULL)  
{  
    PopItem = Head;  
    Head = Head->Next;  
}
```

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.