March 7, 1996

V2

cs450 - Operating Systems 1
Spring 1996

Midterm Exam
open books, open notes

Starts: **6:20 pm** Ends**: 8:00 pm**

Name:______________________________________(please print)

ID:__________________________

Problem	Max points	Your mark	Comments
1	40		20 + 5*4
2	20		10 + 10
3	25		5 + 10 + 10
4	15		10 + 5
	100		

1. Some operating system uses a preemptive SJF (Shortest Job First) algorithm for CPU scheduling. Processes with a lower ID number have priority in case of a tie. Several processes are submitted for execution: their arrival times and burst-intervals are summarized in the following table.

Process	Arrival	Burst Distribution
P1	0	C[a](1), D1[b](1.5), C(1), D2[c](2), C(2)
P2	0.5	C(1.5), D2(2.5), C(0.5), D2(1.5), C(1)
P3	0.5	C(0.5), D1(0.5), C(0.5), D2(0.5), C(0.5), D1(0.5), C(0.5), D2(0.5), C(0.5), D1(0.5), C(0.5)

a. C stands for the CPU and the number between parantheses for the length of the CPU burst
b. D1 stands for an I/O request on disk #1; the number between parantheses is the length of the I/O burst
c. D2 stands for an I/O request on disk #2; the number between parantheses is the length of the I/O burst

The length of the I/O burst time does not include the waiting time in the disk queue. Disk requests are processed on a FCFS basis.

a) Draw the Gantt chart illustrating the execution of these processes. Use the ruled paper attached to your exam.

b) What is the CPU utilization?

c) What is the throughput?

d) What is the average turnaround time?

e) What is the average waiting time? The waiting time is defined as the *total* time spent by a process in the waiting queue.

f) What is the average waiting time spent in disk waiting queues?

2. Fetch And Add (FAA), *atomically* increments a variable and returns the old value of the variable. `FAA(var, increment)` adds `increment` to the variable `var` and returns the initial value of `var`. We plan to use this for mutual exclusion:

```
shared int S;

// each process executes
while(TRUE){
      enter_section();
      ......                  // here is the critical section
      exit_section();
      ......                  // this is the remainder section
}

// enter_section() returns when it is safe to enter the critical
// section
void enter_section()
{
      while( FAA(S,-1) < 1 ) {
            FAA(S,1);
      }
}
```

a) What should be the initial value of S for this code to work? | *S* = |

b) Write the code for `exit_section()`.

3. Two processes, P1 and P2, execute at the same time the following code:

```
for (i=2, i>0; i--)
      S = S+1;
```

where S is a shared integer with the initial value of 0.

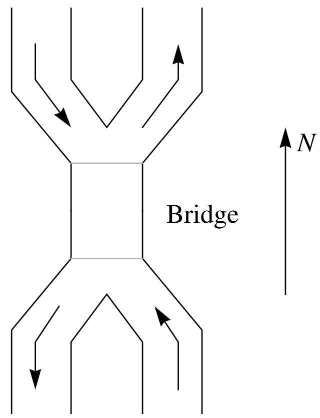
a) What would be the value of S if the two processes would execute to completion in the order P1, P2 (this means that P1 starts and runs to completion, then P2 executes)?

$S =$

b) What are possible values for S in reality? Hint: write the assembly code that implements the `for` loop.

c) Use semaphore(s) to fix the problem: in other words, modify the code in such a way as to always obtain the result described in part a.

4. At same place, the North-South Highway goes over a river. The bridge is in construction and the traffic on the bridge is on a single lane. At most MAX cars can be on the bridge at any time, all going the same direction (either North or South, but never in both directions at the same time).



a) Devise a solution with semaphores for this problem.

b) Is your solution starvation-free? Explain.
