

# MC378 Midterm Exam

Serial #:

This exam is closed-book and mostly closed-notes. You may, however, use a single 8 1/2 by 11 sheet of paper with *hand-written* notes for reference. (Both sides of the sheet are OK.)

Please write your name only on this page. Do not turn the page until instructed, in order that everyone may have the same time. Then, be sure to look at all problems before deciding which one to do first. Some problems are easier than others, so plan your time accordingly. You have 50 minutes to work.

Write the answer to each problem on the page on which that problem appears. You may also request additional paper, which should be labeled with your test number and the problem number.

**If you are stuck, ask for help. At worst, I'll offer to sell you a hint for some points.**

Name: \_\_\_\_\_

Problem	Page	Possible	Score
1	2	12	
2	3	12	
3	4	12	
4	5	12	
<b>Total</b>		48	

1. [ **12 Points** ] Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time. In answering the questions, use nonpreemptive scheduling and base all decisions on the information that you have at the time the decision must be made.

<b>Process</b>	<b>Arrival Time</b>	<b>Burst Time</b>
$P_1$	0	10
$P_2$	2	5
$P_3$	5	1

- Draw a Gantt chart illustrating the scheduling of these processes using FCFS.
- What is the waiting time of each process using FCFS?
- Draw a Gantt chart illustrating the scheduling of these processes using SJF.
- What is the waiting time of each process using SJF?

2. [ 12 Points ] Suppose we want to model a dressing room that can be shared by any number of people at a time, but never both men and women at the same time. We'll use a Java class, with four synchronized methods: `manEnters()`, `manExits()`, `womanEnters()`, and `womanExits()`. Each of the `Enters` methods is supposed to wait until the entry is rule-abiding. Fill in the blanks. You may use simple Java facilities like arithmetic, and also synchronization features like `wait()`, `notify()`, and `notifyAll()`. You should not make people wait more than necessary.

```
public class DressingRoom{

    private ----- = -----;

    private ----- = -----;

    public synchronized void manEnters() throws InterruptedException {

        while(-----)

            -----;

        -----;

    }

    public synchronized void manExits(){

        -----;

        if(-----)

            -----;

    }

    public synchronized void womanEnters() throws InterruptedException {

        while(-----)

            -----;

        -----;

    }

    public synchronized void womanExits(){

        -----;

        if(-----)

            -----;

    }

}
```

3. [ **12 Points** ] Suppose a system has only two page frames, which start empty. Given the following page reference string, show whether each reference is a hit or miss, and if a miss, what the new contents of the two frames are. Write an H for a hit, and a pair of page numbers for a miss, e.g., “0, 1.” (If one page frame remains empty, you can write just a single page number.) Do this for FIFO and LRU replacement, in the respective columns.

	FIFO	LRU
0		
1		
0		
2		
0		
1		

4. [ **12 Points** ] The two most recent papers we read, on Hummingbird and SIS, both involve taking advantage of the unusual characteristics of how specific applications use the file system. Write one to three paragraphs on this theme. Be sure to mention what the specific application is in each case, what is unusual about that application's file system usage, and (in general terms) how those unusual characteristics are exploited.