

CS162 Fall 2003
 Prof. Alan Jay Smith
 Midterm 1

1. Describe the differences between processes, programs, and threads. (As part of your answer, you should define each of these.) (12)
2. For the following table of jobs, arrival times (NOT interarrival times), and service times, compute the completion times of each job and the average flow time under FCFS, SJF, and RR. For RR, $Q = 1.0$. Show your work. (15)

Job	Arrival Time	Service Time
A	0	10
B	2.1	8
C	5.2	3
D	11.3	12

3. a. Explain why for scheduling CPU processes, SET is expected to have a lower mean flow time than RR which should have a lower mean flow time than FCFS. (8)
 - b. Now consider the situation of a computer technician building PCs on a workbench. The components of each PC are specified in a parts list. (E.g. CPU board, memory SIMMS, power supply, 40GB disk, 80 GB disk , JAZ drive, ...). The tech receives orders to build computers over time. He ships each computer when he finished it. He could schedule the assembly using SET, RR, FCFS. Which of these algorithms would give the best performance (minimum mean flow time)? Which would give the worst? Why? Explain. (8)
 - c. Would it be possible to schedule the computer assembly (as in part (b)) using SRPT? Why or Why Not? If it is possible, how would SRPT perform relative to SET, RR and FCFS? (4)
4. Suppose process A will attempt to acquire resources R1, R2, R3 in that order, and process B will attempt to acquire resources R3 then R1. Draw a resource request graph illustrating the possible deadlock. What characteristic of this graph shows deadlock? Give an approach to prevent this from happening. (12)
 5. In this problem we attempt to deal with the “Too Much Milk” problem by providing two methods. Any thread calling `buyMilkIfWeNeedIt()` at any time should buy mil if and only if there is no milk. Any thread calling `drinkMilkIfWeHaveIt()` at any time should drink milk if and only if there is milk. It should never occur that `buy_milk()` is called twice without an intervening call to `drink_milk`, and vice versa. Determine what, if anything, is wrong with this solution. (please list and explain everything that you think is wrong) (12)

```
boolean gotmilk = false;
```

```
Lock mutex = new Lock() ;
```

```
public buyMilkIfWeNeedIt()
{
    if (gotmilk)
        return;
    mutex.acquire();
    buy_milk( );
    gotmilk = true;
    mutex.release() ;
}
```

```

public drinkMilkIfWeHaveIt ()
{
    if (!gotmilk)
        return;
    mutex.acquire();
    drink_milk();
    gotmilk = false;
    mutex.release();
}

```

6. A computer system has a 36-bit virtual address space with a page size of 64 kilobytes. There are 4 bytes per page table entry. Each page table entry uses 1 bit for valid, 1 bit for “reference”, three bits for protection, 1 bit for dirty, and the rest for the real page address.
- How many pages are in the virtual address space? Explain. (3)
 - What is the maximum size of addressable physical memory on this system? Explain. (3)
 - How large is the page table, assuming that it is one level and all entries are valid? Explain. (3)
 - Suppose the currently running process makes a call to load data from 36-bit virtual address x12345BEEF. Draw a picture of how this address is translated using paging and show how the actual data is found in physical memory. (8)
7. For the following two cases, please either show a complete safe sequence or prove that there isn't one (12):

Process	has-X	has-Y	max-needs-X	max-needs-Y
A	30	10	60	40
B	50	80	100	220
C	30	20	105	50
D	0	70	50	90

- (a) available : X : 40 Y: 40
 (b) available: X: 40 Y: 35