

GEORGE MASON UNIVERSITY
Computer Science Department

CS 571 – Operating Systems

Fall 2002

Review Questions

Sanjeev Setia

1. Describe the steps taken by an operating system kernel to context-switch between (i) processes (ii) kernel-level threads.
2. What are the advantages of using user-level threads as opposed to kernel-level threads for an application? What are the disadvantages? Explain using examples.
3. When is busy-waiting preferable to using locks while trying to acquire a resource? Explain through an example.
4. Why does UNIX favor I/O bound processes over CPU bound processes? How can the operating system determine if a process is CPU-bound or I/O-bound? How does the implementation of the UNIX scheduling policy end up favoring interactive processes while at the same time not starving CPU bound processes?
5. List at least 4 hardware features provided by current computer architectures in order to support time-sharing operating systems such as UNIX.
6. Suppose that a machine's instruction set included an instruction named **swap** that operates as follows (as an indivisible instruction):

```
swap(a,b)
boolean a,b;
{
    boolean t;
    t := a;
    a := b;
    b := t;
}
```

Show how **swap** can be used to implement the P (wait) semaphore operation.

7. Suppose your computer does not have a **swap** or **Test-and-Set-Lock** instruction. Explain in not more than 2 or 3 sentences how the P (wait) operation would be implemented if the computer is (i) a uniprocessor (ii) a multiprocessor.
8. Recall the dining philosopher problem. Consider the following code executed by philosopher *i*.

```

repeat
    wait(chopstick[i]);
    wait(chopstick[(i+1) mod 5]);
    :
    EAT
    :
    signal(chopstick[i]);
    signal(chopstick[(i+1) mod 5]);
    :
    THINK
    :
until false;

```

Discuss what is wrong with this code.

9. a. Consider the following set of processes:

<u>Process</u>	<u>Arrival Time</u>	<u>Execution Time</u>
P1	0	3
P2	0	3
P3	3	3
P4	6	3
P5	10	3

What is the *average turnaround time* for this set of processes under a FCFS policy? Under a Round Robin policy? (For the Round Robin policy, assume that the quantum length is 1 and that a newly arrived process joins the tail of the ready queue).

- b. Now consider the following set of processes:

<u>Process</u>	<u>Arrival Time</u>	<u>Execution Time</u>
P1	0	5
P2	0	1
P3	3	1
P4	6	4
P5	10	4

What is the *average turnaround time* for this set of processes under a FCFS policy? Under a Round Robin policy?

- c. Based on the answers to (a) and (b), can you draw any conclusions about the relative performance of FCFS and RR in general? Note that in (a) and (b), the *sum of the execution times* of the processes is identical. Similarly, the arrival times of processes in (a) and (b) are identical. What is the reason for the difference in the relative performance of FCFS and RR, if any?

NOTE: It is a good idea to attempt the questions at the end of each chapter of the textbooks in preparation for the midterm.