

**CS 4431 Advanced Operating Systems**

**Winter 99**

**Midterm (Feb 2, Noon to 1:25 PM)**

**Open Book and Notes (Max = 100 points)**

**Answer in the space provided**

**Bullets and not wordy sentences**

**Legible writing is a requirement not an option!**

**i.e.**

**Don't write like your Prof!**

1. (1 point 1 min) The term “do the dirty bird” signifies
  - (a) an obscene gesture done by motorists to fellow motorists to express displeasure
  - (b) a term along with 2-minute warning popularized by the Psyche researchers
  - (c) what cows are trained to do on TV commercials to promote chicken dinners
  - (d) an ancient medieval ritual to summon the evil spirits
  - (e) a victory dance practiced by a little-known NFL team
  
2. (4 points, 4 min) (*True or False with justification*) Psyche is a thread-aware operating system.

3. (5 points, 5 min) (*Answer True or False with Justification*) Context switching among threads within the same address space is cheaper than switching among processes that are bound to different address spaces.

4. (5 points, 5 min) Explain the intuition behind the 2-minute warning in Psyche.

5. (25 points, 20 min)

```
serializer ser (ep1, ep2)
{
    queue q1, q2;
    crowd c1, c2;
    bool written = false;
    bool busy = false;
    int X = 0;
    int readers = 0;

    int ep1()
    {
        int stat;

        enqueue(q1) until (written);
        readers++;
        join (c1) {
            stat = X + readers;
        }
        readers--;
        return(stat);
    }

    void ep2(int arg)
    {
        enqueue (q2) until (!busy);
        busy = true;
        written = false;
        join (c2) {
            X = arg;
        }
        written = true;
        busy = false;
    }
}
```

In time order the following events happen:

- Threads T1 and T2 call ep1();
- Thread T3 calls ep2(10);

Apart from the inherent properties of the serializer make no further assumptions about the scheduling of the threads. Clearly explain what happens in the execution of the above calls. What possible values can T1 and T2 get back from their respective calls to ep1(), and why?



6. (30 points, 30 min)

```
monitor mon (read_start, read_end, write_start, write_end)
{
    bool busy = false;
    queue q;

    read_start()
    {
        if (busy) wait(q);
    }

    read_end()
    {
        if (queue(q)) signal(q);
    }

    write_start()
    {
        if (busy) wait(q);
        busy = true;
    }

    write_end()
    {
        busy = false;
        if (queue(q)) signal(q);
    }
}
```

This monitor is supposed to allow either **exactly** one writer thread, or **any number of simultaneous** reader threads to access a resource that is outside the monitor. Answer the following questions with respect to this monitor:

- (a) Explain what is incorrect about the above monitor solution.
- (b) Fix it so that it satisfies the problem definition.





7. (30 points, 20 min) This question has to be done in the context of the **Solaris** operating system on a **uniprocessor**. Assume that a switch from one user level thread to another (without change of LWP) costs 1 microsecond; a switch from one LWP to another in the same process costs 20 microseconds, and a switch from one LWP to another in a different process costs 30 microseconds. The first two switches incur no further overheads, while the last switch has a hidden memory hierarchy related overhead of 500 microseconds for the newly scheduled thread.

Consider the following scenario: Process P1 has threads U1 and U2 bound to LWP1, and thread U3 bound to LWP2. Process P2 has thread U4 on LWP3 and thread U5 on LWP4. Assume that currently U3 is executing on the processor.

Consider the following sequence of scheduling events:

- At time  $t_0$ , switch to U2, which needs 1 millisecond of execution time
- Then switch to U1, which needs 3 milliseconds of execution time
- Then switch to U4, which needs 5 milliseconds of execution time
- Then switch to U5, which needs 5 milliseconds of execution time

What is the total elapsed time (from  $t_0$ ) until and including the completion of U5's execution? Include all the costs of scheduling and related overheads, and the execution times of the threads as scheduled above. Explain your answer clearly showing how you account for the costs. Without any explanation you cannot expect to get any credit.

