CS 4/53201 **Project #2, Question #4** Operating Systems

Due via email by 11:59pm on Friday 6 November 1999

The Problems

4. (25 points) It's a little known fact, but Bill Gates has a younger brother—Billy Bob Gates. Much less famous than his older brother, Billy Bob works in a dark room deep within the Microsoft campus, where Bill makes him write Nachos code for a living (ack!). Even worse, Billy Bob isn't a very good programmer. For example, he recently wrote the following code, which purports to allocate technical support people to customers who call in with questions on Microsoft projects:

```
int allocate()  /* Returns index (ID number) of available technician.*/
{
     int i;

     P(nfree);      /* wait until a technician is available */
     for (i=0; i < NTECHS; i++)
          if (available[i] != 0) {    /* tech is available */
               available[i] = 0;     /* tech is now busy */
               return i;
          }
     /* should never get here */
}
```

NTECHS is a constant that indicates the number of available technicians, and *nfree* is a counting semaphore that is initialized to the *NTECHS*. Array available keeps track of the technicians; a value of 0 indicates a busy technician, and a non-zero value indicates an available technician. As each call to tech support comes in, a master control program uses this *allocate* function to get the ID number of an available technician, and then assigns the call to that technician.

Similarly, Billy Bob wrote this code for the master control program to use to "release" the technician when he or she finishes the call:

```
release(int technician)         /* releases technician */
{
     available[technician] = 1;
     V(nfree);
}
```

In the file **threads/threadtest.cc**, write a function *TechSupportTest* with the functionality described below. Modify **threads/main.cc** so that if Nachos is called with the "*–ts*" command line option, *TechSupportTest* is called. Your function *TechSupportTest* should generate calls to *allocate* to simulate tech support calls coming in, and calls to *release* to simulate the custormer hanging up.

After writing this code, in a file called **proj2.prob4**, do the following:

a. Include a sample run to demonstrate that everything you implemented works.

b. One problem that's been noticed with Billy Bob's code is that occasionally two people call at the "same" time, and both get assigned to the same technical support person. Explain how this can occur. Does this have anything to do with the line in the code that says "should never get here"?

c. Modify your *TechSupportTest* code as necessary, and add calls to *Thread::Yield* where appropriate, to demonstrate the problem mentioned in part b. Iinclude a sample run to demonstrate that this problem occurs. (Note — after you add these *Yield*s, comment them out when you email the code to the TA.)

d. Use semaphores in the appropriate manner to correct this problem, and include a sample run to demonstrate that you have fixed the problem.

e. If you did not complete this problem, clearly describe what you have done, what is not working, and how you would go about finishing the problem if you had more time.