Nachos Assignment 1

Hadi Salimi, Sadeq Nobari Computer Engineering Department, Iran University of Science and Technology, Due date: July 23, 2008, 12:00 PM.

Note: This assignment has been adopted from the homepage of professor <u>Sang. H. Song</u> at Virginia University.

Note: All the problems should be implemented under NachOS.

Note: The project should be delivered to Mr. Nobari in person. So do not e-mail it.

Part 1:

1. Implement producer/consumer communication through a bounded buffer with size 4, using lock and condition variables. The producer places characters from the string "Hello world" into the buffer one character at a time; it must wait if the buffer is full. The consumer pulls characters out of the buffer one at a time and prints them to the screen; it must wait if the buffer is empty. Test your solution with a multi-character buffer and with multiple producers and consumers. Of course, with multiple producers or consumers, the output display will be gobbledygook; the point is to illustrate.

Part 2:

1. You have been hired by the traffic control center to synchronize traffic over a narrow lightduty bridge on a public highway. Traffic may only cross the bridge in one direction at a time, and if there are ever more than 10 vehicles on the bridge at one time, it will collapse under their weight. In this system, each car is represented by one thread, which executes the procedure *OneVehicle* when it arrives at the bridge:

```
OneVehicle(int direc)
{
    ArriveBridge(direc);
    CrossBridge(direc);
    ExitBridge(direc);
}
```

In the code above, direc is either 0 or 1; it gives the direction in which the vehicle will cross the bridge. Write the procedures *ArriveBridge* and *ExitBridge* (the *CrossBridge* procedure should just print out a debug message), using lock and condition variables. *ArriveBridge* must not return until it is safe for the car to cross the bridge in the given direction (it must guarantee that there will be no head-on collisions or bridge collapses). *ExitBridge* is called to indicate that the caller has finished crossing the bridge; *ExitBridge* should take steps to let additional cars cross the bridge. This is a lightly-traveled rural bridge, so you do not need to guarantee fairness or freedom from starvation.