Concurrent Programming Project

1. Introduction

This project is designed to be submitted before **midnight of November 19th, 2004**. You will obtain experience with concurrent programming through the BACI concurrent interpreter. Some of the learning objectives and goals are the following:

(a) to learn and implement the Producer Consumer Problem with a shared buffer using semaphores,

(b) to implement the varied rate Producer Consumer program to observe the outcome for different rates.

2. Project Description

You will implement the basic producer consumer problem using BACI by maintaining a shared, bounded buffer. You will need to use semaphores to implement the exclusion operation to enforce the appropriate access of the producer and consumer. The following functions need to be implemented:

void producer(int NullLoop, int NumberOfItems)

The function simulates the producer. **NumberOfItems** is the total data items that the producer should produce. The producer generates a random integer and then puts it into the shared buffer. You will then do the following to display output to the screen:

```
cout << "producer enter a new item " << i << " , value: " << value << endl;
```

i is the index of the new item in the total NumberOfItems items, not the index of its position in the shared buffer. value is the value of the new item.

The **NullLoop** is used to control the speed of the producer. Since you will need a way to change the speed of the producer relative to the consumer, you need to create an empty ‘for’ loop that will iterate NullLoop times. This will allow you to simulate varied rate producer. The bigger NullLoop, the slower the producer runs.

When the producer has produced all the items, it will need to set up another shared variable to indicate that the producer is finished so that the consumer will terminate after the producer terminates and it consumes all the items.

void consumer(int NullLoop, int ConsumerID)

The function simulates the consumer. The **ConsumerID** is the ID for the consumer procedure (using 1, 2 if there are 2 consumers). While you will only have one producer, you may have several consumers, and this variable provides a way to differentiate between multiple consumers.

The consumer will add up all the values that the produced put in the buffer. The consumer will first check whether the producer terminates and whether there are available items in the buffer. If there are available items, it will pick up the items from the buffer (you could use the First In First Out or First In Last Out to pick up the items). The consumer will add the items and then output to the screen using:

```
cout << "consumer " << ConsumerID << " remove an item" << endl;
```

NullLoop is used to control the speed of the consumer just like for the producer() function. You just put the NullLoop numbers of none-operation loops in the iteration. This will allow you to simulate the varied rate for the consumer.

After you finish the implementation coding of the above two functions. Do the following 3 simulation (one producer and one consumer):

(a) producer runs the same speed as the consumer. For example producer(1, 20); consumer(1, 1);
(b) producer runs faster than the consumer.
(c) producer runs slower than the consumer

Observe the different output of your program, and write your observation report. You are also encouraged to observe the scheme of one producer and two consumers in your report.

3. Project Direction

You will write the program based on the BACI interpreter. Its homepage is located at http://www.mines.edu/fs_home/tcamp/baci/.

The BACI support C-- programming, which is a subset of C++ syntaxes. After you write your C-- code, use **bacc** to turn it into PCODE object code executable by the interpreter. Then use **bainterp** to see the result. Here is what your should do to get started:

(a) Download the BACI toolkit from BACI homepage and choose a suitable platform version to develop your program. (BACI has various versions for different platforms)
(b) Download and read through **cmimi.pdf**, the user guide located here: http://www.mines.edu/fs_home/tcamp/baci/bacidocpdf.tar.gz
(c) Download the **obj5.cm** (a framework to help you begin coding) http://www.cs.ucf.edu/courses/cop4600/fall2004/item/obj5.cm.
(d) Write your code.

4. Project Submission

Please email your source code (just the obj5.cm) and the report describing your implementation and observations to haocheng@cs.ucf.edu before the deadline. I will send back the email confirmation if I receive your submission. Thank you!

5. Hints

The C-- is a restricted subset of the C++. Its programming is straightforward. You should first read the user guide and run the example code to get idea of the BACI interpreter before your coding.