Assignment Review

Producer-Consumer Problem

• Synchronization among multiple Java Threads
• One Producer, Multiple Consumers
• Bounded Circular Buffer
Bounded Buffer

- FIFO Buffer.
- Size is taken as command line input.
- One instance of Buffer used by both Producer and Consumer.
Producer

- Generates a random number (0-99)
- Insert elements to the Buffer.
- If the Buffer is full, wait for consumer to consume at-least one item.
- Print location of insertion(position in buffer), time stamp.
- Store the sum of all integers produced, display at the end.

- Example:
  - Producer: Inserted integer 27 at Location 2 at instant: 2017-11-02 15:46:50.20912
  - Producer: Unable to insert, buffer full, at instant: 2017-11-02 15:47:50.21012
Consumer

- Consumes elements from Buffer in the same order as inserted.
- If Buffer is empty, wait for Producer to produce at-least one element.
- One or Two consumer threads at the same time.
- Print location of removal, timestamp of element removal.
- Maintain sum of the elements consumed.

- Example:
  - Consumer 1: Removed 27 from location: 2 at Time: 2017-11-02 15:49:50.20924
  - Consumer 2: Unable to consume, buffer empty, at Time: 2016-10-29 15:57:50.31012
Producer and Consumer

Waiting

producer
V(supply)

full queue

consumer
P(supply)

Producer and Consumer

Waiting

producer
V(supply)

empty queue

consumer
P(supply)
Creating Threads

• Can use ‘extends Thread’ or ‘implements Runnable’
• Use ‘synchronized’ to make sure that Buffer is accessed concurrently without conflict.
• Use Thread.join() to specify driver to wait for completion of a Thread.

Sample Code:

```java
synchronized (sharedQueue) {
    sharedQueue.add(i);
    sharedQueue.notifyAll();
}
```

➢ [http://www.cs.colostate.edu/~cs370/Fall16/lecture-codes/DiningPhilosophers.java](http://www.cs.colostate.edu/~cs370/Fall16/lecture-codes/DiningPhilosophers.java)
Displaying Time

Sample Code:

```java
this.start = System.nanoTime();
DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss.SSSSS");
Calendar cal = Calendar.getInstance();
```

Packages that can be used:
- java.sql.Date
- java.text.SimpleDateFormat
- java.text.DateFormat
- java.util.Calendar
Other Specifications (1/2)

• 4 Command-line arguments:
  - Number of Consumers
  - Buffer Size
  - Number of items to be produced
  - Maximum wait time

• Arguments for Producer and Consumer Program:
  - Size of Buffer
  - Maximum wait time
  - Number of items to be produced (Only for Producer)
Other Specifications (2/2)

- 1 Producer Thread, 1 or 2 Consumer Threads
- Consumer Thread terminates if consumer is empty for 15 times the maximum waiting time.
- Program terminates when all threads are terminated.
- Timestamp should have one-hundredth millisecond resolution.
Correctness Verification

- Items consumed should match the items produced.
- Only one thread of Circular Buffer accessed by the Producer and Consumer
- No deadlocks
- Program should work for any number of items and maximum wait time.
Do’s and Don'ts

- Use only circular bounded buffer.
- Do not use Thread.sleep(). Instead, use wait() and notify().
- Do not use methods from "java.util.concurrent" package.
- Do not use Boolean flags to toggle among Producer and Consumer.
Requirements

- Single `.tar` file that contains all the Java files, Readme.txt file.
- You can use `tar -cvf <filename>.tar <list of files>` to tar your submission.
- Include only the `.java` files not the src folder or any other folder structures.
- Name the driver as HW6.java.
- Make sure you remove “package org.java……” from the java files.
- Program should run on CS Department machines.

Failure to follow the above instructions will lead to a penalty of 10 points.
Questions?