COMP2511

Object-Oriented Design and Programming

Concurrency

Wayne Wobcke

w.wobcke@unsw.edu.au

Producers and Consumers

```
public class Producer implements Runnable;
  public void run() {
    try {
      if (!queue.isFull()) queue.add(i);
    }
    catch (InterruptedException exception) {};
  }

public class Consumer implements Runnable;
  public void run() {
    try {
      if (!queue.isEmpty()) queue.remove();
    }
    catch (InterruptedException exception) {};
  }
```

Today’s Lecture

- Threads
- Race Conditions
- Object Locks
- Reentrant Locks

Threads

```
public class ThreadTester {

  BoundedQueue<String> = new BoundedQueue<String>(10);

  Runnable run1 = new Producer(“Hello”, queue);
  Runnable run2 = new Producer(“Hello”, queue);
  Runnable run3 = new Consumer(“Goodbye”, queue);

  Thread t1 = new Thread(run1);
  Thread t2 = new Thread(run2);
  Thread t3 = new Thread(run3);

  t1.start();
  t2.start();
  t3.start();
}
```

BoundedQueue

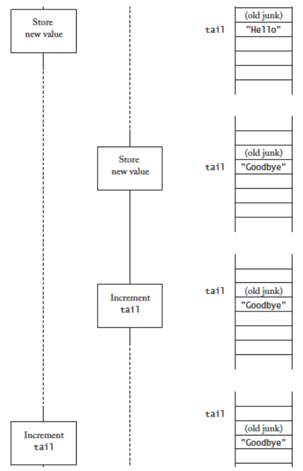
```
public void add(E newValue) {
   elements[tail] = newValue;
   tail++;
   size++;
   if (tail == elements.length)
      tail = 0;
}
```

```
public E remove() {
   E r = elements[head];
   head++;
   size--;
   if (head == elements.length)
      head = 0;
   return r;
}
```

```
public boolean isFull() {
   return size == elements.length;
}
```

```
public boolean isEmpty() {
   return size == 0;
}
```

Race Condition



Exercise

- How could size be 11?

Object Locks

```
public synchronized void add(E newValue)
{
  …
}
```

```
public synchronized E remove()
{
  …
}
```

- Why doesn’t this work?

Simple Locks

```
public void add(E newValue) {
  queueLock.lock();
  while (isFull()) sleep();
  elements[tail] = newValue;
  tail++;
  size++;
  if (tail == elements.length)
    tail = 0;
  queueLock.unlock();
}
```

```
public E remove() {
  queueLock.lock();
  while (isEmpty()) sleep();
  E r = elements[head];
  head++;
  size--;
  if (head == elements.length)
    head = 0;
  queueLock.unlock();
  return r;
}
```

- Why doesn’t this work?

Object Locks

```
public synchronized void
    add(E newValue) {
  while (isFull()) wait();
  elements[tail] = newValue;
  tail++;
  size++;
  if (tail == elements.length)
    tail = 0;
  notifyAll();
}
```

```
public synchronized E
    remove() {
  while (isEmpty()) wait();
  E r = elements[head];
  head++;
  size--;
  if (head == elements.length)
    head = 0;
  notifyAll();
  return r;
}
```

- This does work, but can it be better?

Reentrant Locks

```
public void add(E newValue) {
  queueLock.lock();
  try {
    while (isFull()) space.await();
    elements[tail] = newValue;
    tail++;
    size++;
    if (tail == elements.length)
      tail = 0;
    value.signalAll();
  }
  finally {queueLock.unlock();}
}
```

```
public E remove() {
  queueLock.lock();
  try {
    while (isEmpty()) value.await();
    E r = elements[head];
    head++;
    size--;
    if (head == elements.length)
      head = 0;
    space.signalAll();
    return r;
  }
  finally {queueLock.unlock();}
}
```

Next Week

- Project Assessment
- Review and Sample Exam
  - E-mail or post questions