Concurrency

EECS 4315

www.eecs.yorku.ca/course/4315/

Books

- Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes and Doug Lea. *Java Concurrency in Practice*. Addison-Wesley, 2006.
- Mary Campione, Kathy Walrath and Alison Huml. *The Java Tutorial. Lesson: Threads: Doing Two or More Tasks At Once*.
- James Gosling, Bill Joy, Guy L. Steele Jr., Gilad Bracha and Alex Buckley. *The Java Language Specification*. 2015.

Concurrency

Threads can exchange information by accessing and updating shared attributes.

Question

One thread executes

```
v = 1;
v = v + 1;
```

and another thread executes

```
v = 0;
```

What is the final value of v?

Concurrency

Threads can exchange information by accessing and updating shared attributes.

Question

One thread executes

```
v = 1;
v = v + 1;
```

and another thread executes

```
v = 0;
```

What is the final value of v?

Answer

0, 1 or 2. This example shows that concurrency gives rise to nondeterminism.

Concurrency

Question

One thread executes

```
v = v + 1;
```

and another thread executes

```
v = v + 1;
```

If the initial value of v is 0, then what is the final value of v?

Concurrency

Question

One thread executes

```
v = v + 1;
```

and another thread executes

```
v = v + 1;
```

If the initial value of v is 0, then what is the final value of v?

Answer

1 or 2.

Concurrency

Question

How can the final value of v be 1?

Concurrency

Question

How can the final value of v be 1?

Answer

The assignment `v = v + 1` is not atomic.

Concurrency

Question

How can the final value of v be 1?

Answer

The assignment `v = v + 1` is not atomic.

```
0: getstatic
3: iconst_1
4: iadd
5: putstatic
```

Concurrency

Question

One thread executes

```
v = 0;
```

and another thread executes

```
v = Long.MAX_VALUE;
```

How many different final values can v have?

Concurrency

Question

One thread executes

```
v = 0;
```

and another thread executes

```
v = Long.MAX_VALUE;
```

How many different final values can v have?

Answer

4 (on 32-bit machines).

Concurrency

Question

How can v have 4 different final values?

Concurrency

Question

How can v have 4 different final values?

Answer

The assignments `v = 0` and `v = Long.MAX_VALUE` may not be atomic (on 32 bit machines).

Thread creation

In Java, threads are created dynamically:

```
// create and initialize Thread object
Thread thread = new Thread();
// execute run method of Thread object concurrently
thread.start();
```

The class `Thread` is part of package `java.lang` (and, hence, does not need to be imported). Its API can be found at the URL

https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html .

Thread API

- `public Thread(String name)`
  Initializes a new `Thread` object with the specified name as its name.
- `public void start()`
  Causes this thread to begin execution; the Java virtual machine calls the run method of this thread.
- `public void run()`
  This method does nothing and returns.

Printer

Question

Develop a Java class called `Printer` that is a `Thread` and prints its name 1000 times.

Printer

```java
public class Printer extends Thread {
  public Printer(String name) {
    super(name);
  }

  public void run() {
    final int NUMBER = 1000;
    for (int i = 0; i < NUMBER; i++) {
      System.out.print(this.getName());
    }
  }
}
```

Two concurrent printers

Question

Develop an app that creates two `Printer`s with names 1 and 2 and run them concurrently.

Two concurrent printers

```
public class TwoPrinters {
  public static void main(String[] args) {
    Printer one = new Printer("1");
    Printer two = new Printer("2");
    one.start();
    two.start();
  }
}
```

Two concurrent printers

Question

What is the output of the app?

Two concurrent printers

Question

What is the output of the app?

Answer

A sequence of 1000 1's and 2's (arbitrarily interleaved). This example shows that concurrency gives rise to nondeterminism.

Two concurrent printers

Question

What happens if we replace `start` with `run` in the app?

Two concurrent printers

Question

What happens if we replace `start` with `run` in the app?

Answer

Let's try it.

Two concurrent printers

Question

What happens if we replace `start` with `run` in the app?

Answer

Let's try it.

Answer

The output is a sequence of 1000 1's followed by 1000 2's

Java only supports single inheritance

The following is not allowed in Java.

```
public class Printer extends Applet, Thread
```

Thread creation

```
// create and initialize Runnable object
Runnable runnable = new ...();
// create and initialize Thread object
Thread thread = new Thread(runnable);
// execute run method of Runnable object concurrently
thread.start();
```

The interface `Runnable` is part of package `java.lang` (and, hence, does not need to be imported). Its API can be found at the URL

`https://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html`

Runnable is an interface

In Java, you cannot create instances of an interface.

```
public class Printer implements Runnable {
  ...
}
```

The assignment

```
Runnable printer = new Printer();
```

is valid since the class `Printer` implements the interface `Runnable`.

Printer

Question

Develop a Java class called `Printer` that implements `Runnable` and prints the thread's name 1000 times.

Printer

```java
public class Printer implements Runnable {
  public void run() {
    final int NUMBER = 1000;
    for (int i = 0; i < NUMBER; i++) {
      System.out.print(Thread.currentThread().getName());
    }
  }
}
```

Two concurrent printers

Question

Develop an app that creates two `Printer`s with names 1 and 2 and run them concurrently.

Printer

```
public class TwoPrinters {
  public static void main(String[] args)    {
    Printer printer = new Printer();
    Thread one = new Thread(printer, "1");
    Thread two = new Thread(printer, "2");
    one.start();
    two.start();
  }
}
```

Printer

In particular when the `run` method is small, one might use an anonymous class.

An introduction to anonymous classes can be found here.

Printer

```
public static void main(String[] args) {
  for (int i = 1; i <= 2; i++) {
    String name = "" + i;
    (new Thread () {
      @Override
      public void run() {
        final int NUMBER = 1000;
        for (int i = 0; i < NUMBER; i++) {
          System.out.print(name);
        }
      }
    }).start();
  }
}
```

Incrementer

Question

Develop a Java class called `Incrementer` that is a `Thread` and increments a shared static attribute named `value`.

Incrementer

```
public class Incrementer extends Thread {
  public static int value = 0;

  public void run () {
    Incrementer.value++;
  }
}
```

Incrementer

Question

Develop an app that creates two `Incrementer`s and run them concurrently. Assert that the final value of `value` is two.

Two incrementers

```
public class TwoIncrementers {
  public static void main(String[] args) {
    try {
      Incrementer one = new Incrementer();
      Incrementer two = new Incrementer();
      one.start();
      two.start();
      one.join();
      two.join();
      assert Incrementer.value == 2;
    } catch (InterruptedException e) {}
  }
}
```

Using JPF

We can use JPF to check whether the assertion hold for each execution.

```
target=TwoIncrementers
classpath=<path to TwoIncrementers.class>
```

Using JPF

```
JavaPathfinder core system v8.0 (rev d772dfa80ea692f916aa67

====================================== syst
TwoIncrementers.main()

====================================== sear

====================================== erro
gov.nasa.jpf.vm.NoUncaughtExceptionsProperty
java.lang.AssertionError
       at TwoIncrementers.main(TwoIncrementers.java:7)
...
```

Using jpf-visual

Install jpf-shell and jpf-visual.

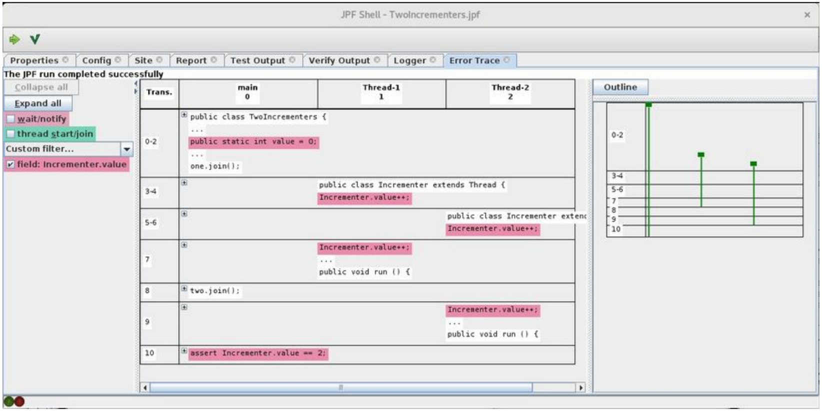
```
target=TwoIncrementers
classpath=<path to TwoIncrementers.class>
sourcepath=<path to TwoIncrementers.java>

@using jpf-visual
report.errorTracePrinter.property_violation=trace
report.publisher+=,errorTracePrinter
report.errorTracePrinter.class=ErrorTracePrinter
shell=gov.nasa.jpf.shell.basicshell.BasicShell
shell.panels+=,errorTrace
shell.panels.errorTrace=ErrorTracePanel
```

jpf-visual

`https://www.youtube.com/watch?v=mrgsFxUI88I`

jpf-visual


JPF Shell - TwoIncrementers.jpf

The JPF run completed successfully

Trans.	main 0	Thread-1 1	Thread-2 2
0-2	public class TwoIncrementers { ... public static int value = 0; ... one.join();		
3-4		public class Incrementer extends Thread { Incrementer.value++;	
5-6			public class Incrementer extend Incrementer.value++;
7		Incrementer.value++; ... public void run () {	
8	two.join();		
9			Incrementer.value++; ... public void run () {
10	assert Incrementer.value == 2;		

How many different executions?

Question

One thread prints 1 one. Another thread prints 1 two. How many different executions are there?

How many different executions?

Question

One thread prints 1 one. Another thread prints 1 two. How many different executions are there?

Answer

2.

How many different executions?

Question

One thread prints 2 ones. Another thread prints 2 twos. How many different executions are there?

How many different executions?

Question

One thread prints 2 ones. Another thread prints 2 twos. How many different executions are there?

Answer

6.

How many different executions?

Question

One thread prints 3 ones. Another thread prints 3 twos. How many different executions are there?

How many different executions?

Question

One thread prints 3 ones. Another thread prints 3 twos. How many different executions are there?

Answer

20.

How many different executions?

Question

One thread prints 1000 ones. Another thread prints 1000 twos.
How many different executions are there?

How many different executions?

Question

One thread prints 1000 ones. Another thread prints 1000 twos. How many different executions are there?

Answer

2048151626989489714335162502980825044396424887981397033820382637671748186202083755828932994182610206201464766319998023692415481798004524792018047549769261578563012896634320647148511523952516512277685886115395462561479073786684641544445336176137700738556738145896300713065104559595144798887462063687185145518285511731662762536637730846829322553890497438594814317550307837964443708100851637248274627914170166198837648408435414308177859470377465651884755146807496946749238030331018187232980096685674585602525499101181135253534658887941966653674904511306110096311906270342502293155911108976733963991149120.

How many executions?

Question

One thread prints 1000 ones. Another thread prints 1000 twos. How many different executions are there?

How many executions?

Question

One thread prints 1000 ones. Another thread prints 1000 twos. How many different executions are there?

Answer

$\binom{2000}{1000} = \frac{2000!}{1000!1000!}$.

How many executions?

Question

One thread executes n instructions. Another thread executes n instructions. How many different executions are there?

How many executions?

Question

One thread executes n instructions. Another thread executes n instructions. How many different executions are there?

Answer

At most $\binom{2n}{n}$.

How many executions?

Question

One thread executes n instructions. Another thread executes n instructions. How many different executions are there?

Answer

At most $\binom{2n}{n}$.

Question

Can there be fewer?

How many executions?

Question

One thread executes n instructions. Another thread executes n instructions. How many different executions are there?

Answer

At most $\binom{2n}{n}$.

Question

Can there be fewer?

Answer

Yes. For example, if each instruction is `x = 1` then there is only one execution.

How many executions?

Question

There are k threads. Each thread executes n instructions. How many different executions are there?

How many executions?

Answer

$$\binom{kn}{n}\binom{(k-1)n}{n}\cdots\binom{2n}{n}$$

How many executions?

Answer

$$\binom{kn}{n}\binom{(k-1)n}{n}\cdots\binom{2n}{n}$$
$$= \frac{(kn)!}{n!((k-1)n)!}\frac{((k-1)n)!}{n!((k-2)n)!}\cdots\frac{(2n)!}{n!n!}$$

How many executions?

Answer

$$
\begin{aligned}
&\binom{kn}{n}\binom{(k-1)n}{n}\cdots\binom{2n}{n} \\
&= \frac{(kn)!}{n!((k-1)n)!}\frac{((k-1)n)!}{n!((k-2)n)!}\cdots\frac{(2n)!}{n!n!} \\
&= \frac{(kn)!}{(n!)^k}
\end{aligned}
$$

How many executions?

Answer

$$
\begin{aligned}
&\binom{kn}{n}\binom{(k-1)n}{n}\cdots\binom{2n}{n} \\
&= \frac{(kn)!}{n!((k-1)n)!}\,\frac{((k-1)n)!}{n!((k-2)n)!}\cdots\frac{(2n)!}{n!n!} \\
&= \frac{(kn)!}{(n!)^k} \\
&= \frac{(kn)(kn-1)\cdots(kn-n+1)}{n!}\cdots\frac{2n(2n-1)\cdot(n+1)}{n!}\,\frac{n!}{n!}
\end{aligned}
$$

How many executions?

Answer

$$\begin{aligned}
&\binom{kn}{n}\binom{(k-1)n}{n}\cdots\binom{2n}{n}\\
&= \frac{(kn)!}{n!((k-1)n)!}\frac{((k-1)n)!}{n!((k-2)n)!}\cdots\frac{(2n)!}{n!n!}\\
&= \frac{(kn)!}{(n!)^k}\\
&= \frac{(kn)(kn-1)\cdots(kn-n+1)}{n!}\cdots\frac{2n(2n-1)\cdot(n+1)}{n!}\frac{n!}{n!}\\
&\geq \left(\frac{2n(2n-1)\cdot(n+1)}{n!}\right)^{k-1}
\end{aligned}$$

How many executions?

Answer

$$
\begin{aligned}
&\binom{kn}{n}\binom{(k-1)n}{n}\cdots\binom{2n}{n} \\
&= \frac{(kn)!}{n!((k-1)n)!}\frac{((k-1)n)!}{n!((k-2)n)!}\cdots\frac{(2n)!}{n!n!} \\
&= \frac{(kn)!}{(n!)^k} \\
&= \frac{(kn)(kn-1)\cdots(kn-n+1)}{n!}\cdots\frac{2n(2n-1)\cdot(n+1)}{n!}\frac{n!}{n!} \\
&\geq \left(\frac{2n(2n-1)\cdot(n+1)}{n!}\right)^{k-1} \\
&= \left(\frac{2n(2n-1)\cdot(n+1)}{n(n-1)\cdots 2}\right)^{k-1}
\end{aligned}
$$

How many executions?

Answer

$$
\begin{aligned}
& \binom{kn}{n}\binom{(k-1)n}{n}\cdots\binom{2n}{n} \\
= \quad & \frac{(kn)!}{n!((k-1)n)!}\frac{((k-1)n)!}{n!((k-2)n)!}\cdots\frac{(2n)!}{n!n!} \\
= \quad & \frac{(kn)!}{(n!)^k} \\
= \quad & \frac{(kn)(kn-1)\cdots(kn-n+1)}{n!}\cdots\frac{2n(2n-1)\cdot(n+1)}{n!}\frac{n!}{n!} \\
\geq \quad & \left(\frac{2n(2n-1)\cdot(n+1)}{n!}\right)^{k-1} \\
= \quad & \left(\frac{2n(2n-1)\cdot(n+1)}{n(n-1)\cdots 2}\right)^{k-1} \\
\geq \quad & n^{k-1}
\end{aligned}
$$

How many executions?

Question

There are k threads. Each thread executes n instructions. How many different executions are there?

Answer

In the worst case, more than n^{k-1}.

Conclusion

The number of different executions may grow exponential in the number of threads.