Multi-Threading Summary

- Multi-tasking
- Multi-threading
- Threads in Java

By the end of this lecture, you will be able to describe what a thread is and understand how threads work in Java.

Question: In the programs we have created so far, how many statements are executed at the same time?

When you use your computer, how many programs do you run at the same time? Why?

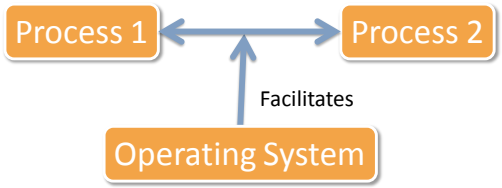
Fact: With only one CPU, only one machine instruction can be executed at a time.

Discussion: If you were to guess, how do you think it is possible that more than one program can run at the same time on your computer?

More about operating systems: CPSC 457

Question: If you wanted to include a picture you received in an email in a presentation you were creating in Keynote, how would you share this information?

Multi-Tasking

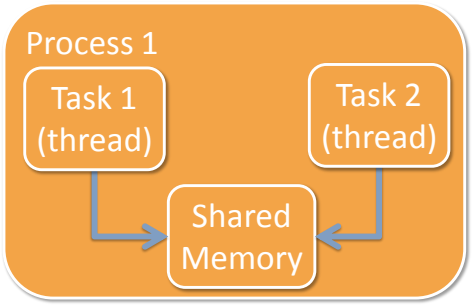


What is the overhead?

Threads

- A single program can "simultaneously" execute different threads of code
- All of a processes threads share the same memory (assigned by the operating system).

Multi-Threading



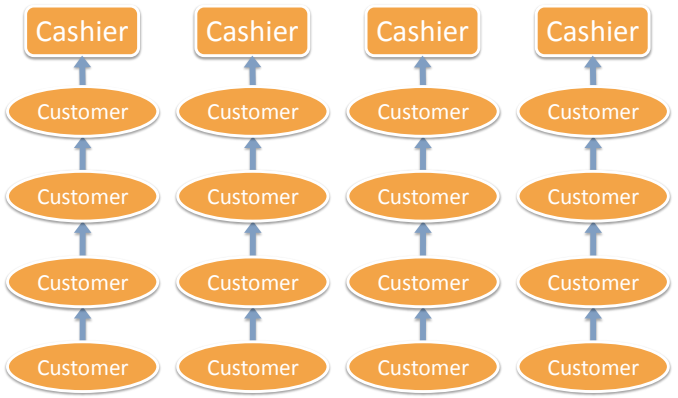
Discussion: What would be the benefit of multi-tasking over multi-threading?

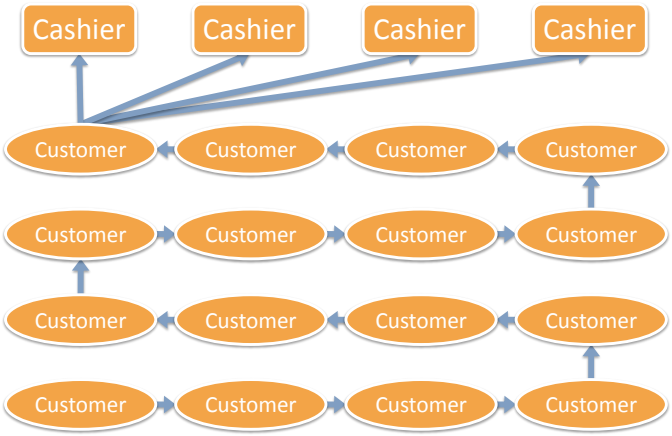
Example: Web Servers

- A web server takes requests from the Internet
- Each request must be handled in turn

How does the grocery store handle many requests from people wanting to pay for their groceries?

How does the bank handle many requests from people wanting to do transactions?

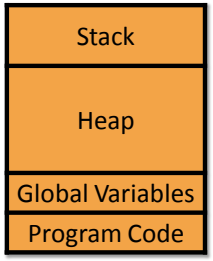




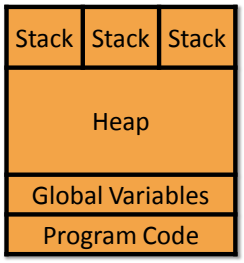
Discussion: What strategies do you use at the grocery store?

Review

- What are the components of a process?



What needs to change?

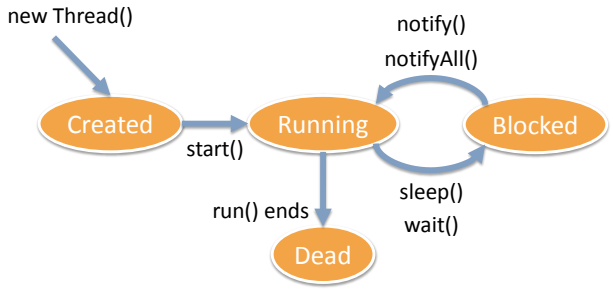


Threads in Java

- Java supports multiple threads
- When a Java program is started, the virtual machine starts up a main thread
- It also sets up other threads to maintain the virtual machine

- Example: garbage collection

Thread States



More about operating systems: CPSC 457

If you were writing a single program, what would be the benefit of using threads?

Starting a new thread

- Instantiate a new Thread object
- Invoke the start() method on that object

Example

```
MyTask aTask = new MyTask();
Thread theThread = null;
if (theThread == null)
{
    theThread = new Thread(aTask);
    theThread.start()
}
```


Runnable Interface

- The Runnable interface has one method: run

MyTask.java

```
public class MyTask implements Runnable
{
    public void run()
    {
        while (true)
        {
            // ....
        }
    }
}
```

Why use an infinite loop?

Discussion: Why are there separate run and start methods, and why don't we just call run directly?

Discussion: what issues might arise with the use of the same memory space in different threads?

Multi-Threading Summary

- Multi-tasking
- Multi-threading
- Threads in Java

Next Class

- Review for Final Exam