

Multi-Threading Summary

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

1

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

2

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

3

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

4

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

5

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

6

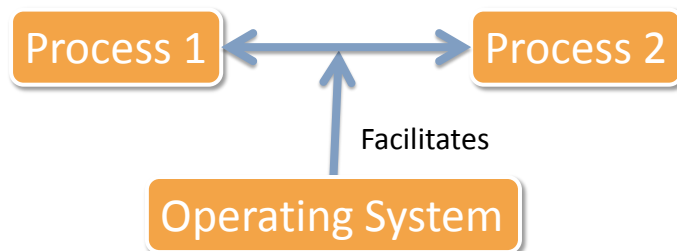
Question: If you wanted to include a picture you received in an email in a presentation you were creating in Keynote, how would you share this information?

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

7

Multi-Tasking



April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

8

What is the overhead?

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

9

Threads

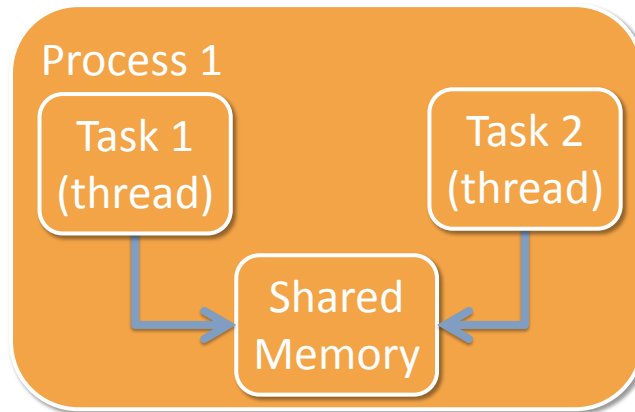
- A single program can “simultaneously” execute different **threads** of code
- All of a processes threads **share** the same memory (assigned by the operating system).

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

10

Multi-Threading



April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

11

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

12

Example: Web Servers

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

13

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

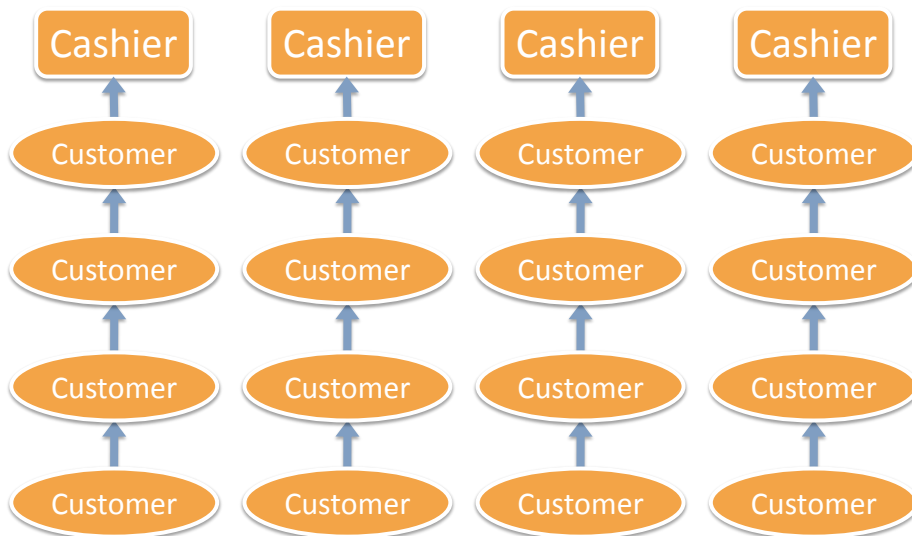
14

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

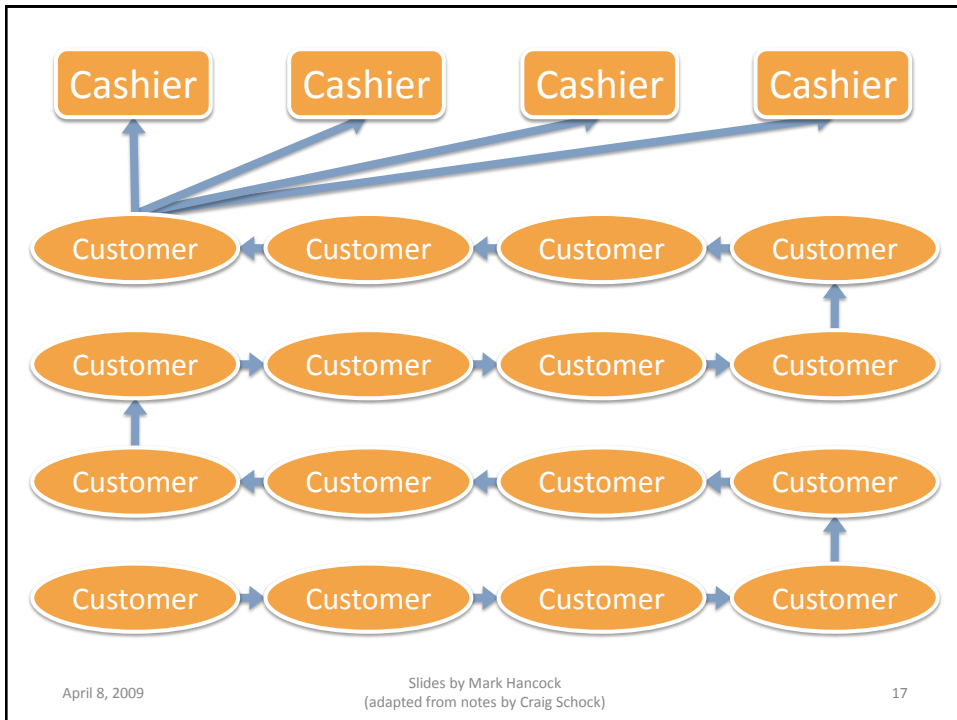
15



April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

16



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

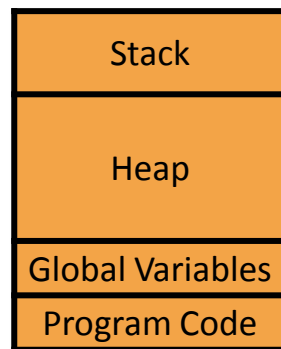
Review

- What are the components of a process?

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

19

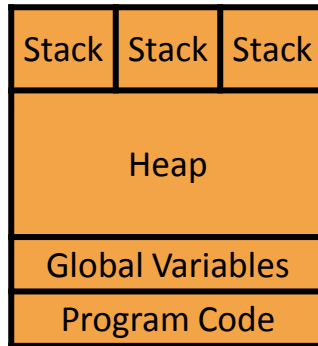


What needs to change?

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

20



April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

21

Threads in Java

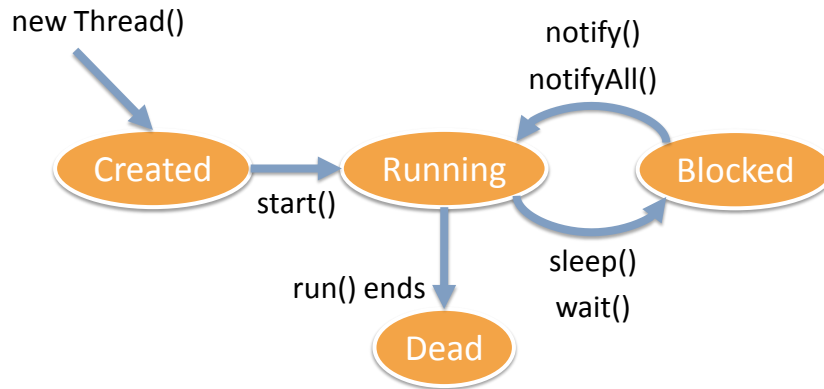
- Java supports multiple threads
- When a Java program is started, the **virtual machine** starts up a main thread
- It **also** sets up other threads to maintain the virtual machine
- Example: garbage collection

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

22

Thread States



More about **operating systems**: CPSC 457

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

23

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

24

Starting a new thread

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

25

Example

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

26

Runnable Interface

- The **Runnable** interface has one method: **run**

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

27

MyTask.java

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

Why use an infinite loop?

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

28

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

29

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

30

Multi-Threading Summary

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

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

31

Next Class

- Review for Final Exam

April 8, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

32