C7

Threads

Concurrency

Concurrency, processes and threads

- Concurrency
  - Multiple activities *(appear to)* occur simultaneously.
  - ‘Time slicing’ allows a single execution unit to give the appearance of concurrent execution.
- Process
  - Distinct execution context that (by default) shares nothing.
- Thread
  - Intra-process execution context.
  - Multiple threads can (and do) execute the same methods on the same objects.

Why threads?

- ‘Concurrency’
  - Separate concerns (e.g. rendering vs. logic)
  - Good for: distinct tasks that naturally occur concurrently
- ‘Parallelism’ (a special case of concurrency)
  - Break task into pieces, exploit parallel hardware
  - Good for: computationally intensive problems that can be readily partitioned


Australian
National
University
J16
Java Threads
Thread and Runnable
start(), join() and sleep()
Races and synchronized
Structured Programming 1110/1140/6710

`Thread` and `Runnable`

- The `Thread` class is used to create threads and interact with them.
- Two ways to create a thread:
  1. Subclass `Thread`, overriding its `run()`method.
     - Correspondence between instances of the class and threads.
     - Disadvantages: can't subclass anything else.
  2. Use the `Runnable` interface and implement its `run()` method.
     - Use `Thread.currentThread()` to access the thread that is executing the `run()` method.

`start()`, `join()` and `sleep()`

- Calling `t.start()` will start execution of the `run()` method within the thread `t` (then continue execution of the current thread).
- Calling `t.join()` will cause the current thread to wait until thread `t` terminates.
- Calling `Thread.sleep(ms)` will cause the current thread to go to sleep for `ms` milliseconds.

Races and the `synchronized` keyword

- Too many cooks…
  - Coordination is the big challenge of concurrency
  - How do we avoid conflicts?
  - How do we impose some level of coherence and order?
- A ‘race condition’ is a situation where one or more threads race non-deterministically to be the first to read or write a variable
- The `synchronized` keyword
  - Qualify a method, ensures only one thread executes that method at any time