CMSC 433
Programming Language Paradigms and Technologies
William Pugh
March 4th
Project 4

• We'll spend some time going over it, show example solutions
Some people used the idea of CountUpDownLatch, but just added the two fields to their ParallelExplorer class

Bad idea

Concurrent software is nasty, subtle tricky stuff

Pull out whatever abstraction you can. Easier to understand and test
Waiting inside a task

- Inside a task
- Create a Runnable or a FutureTask
- Ask the executor to execute it
- wait for the Runnable or future task to finish
- Why is this yuck?
What if we are backed up?

• Assume we have 20 tasks to execute, and only 10 threads to execute them with.

• What if each task requests another task be executed, and then waits for that task to complete?
When should you do it in parallel?

- If you don't need the result (but someone else will), or
- if you can fire all a bunch of things, and then wait for them all to complete
- but we worried about what happens when you get backed up. Better to spawn another task that will wait for them to finish
Project 5 - CompoundLock

• Due Tuesday, March 9th

• You have a compound lock, and can create any number of associated component lock

• If any thread holds the compound lock, no thread may hold any of the associated component locks
Notes

• You only need to support 3 Lock methods
  • lock, tryLock and unlock
• The locks don't have to be reentrant
• The code is simple, although perhaps subtle
• Hint: make sure you look at the Javadoc for ReadWriteLock