Outline

• Final class
• Advanced topics
  – Practical java
  – Threads
  – Networks
  – Unsolvable problems
• Review and overview
  – What we covered and how it fits together
  – Requested topics

Announcements

• Will also post sample exams etc
• Open notes/book exam
• Extra credit assignment posted tonight, will be due at final

CPU

• The “Brains” of the operation
• Can only do one thing at a time
• One would like to be able to run many things at once
• Solution ??
**Algorithms**

- Round robin
- Priority
- Longest wait
- Shortest run

**Traditional program**

1. Start
2. Compute
3. End

**Real programs**

- Always on
- Multiple users
- Multiple states
- Shared resources
- Flexibility
- Cost involvement

**Parallel programming**

- How to program parts?
- How to be able to run on multiple cpu’s?
- Issues:
  - Sharing
  - Communicating
  - Saving
threads

• What are they?
• Cost of switching context
  – Threads under single process
  – Cpu support for threads (mini processes)
• How implemented on computer side
• How supported in language

Code examples

• Example of thread class
• Example of runnable interface

Java.util.*

• Stack class
• Timer
• Vector
• Random

Java.lang.*

• Math
• Enum
• Number
• StringBuffer
  – Usage
  – Advantage for optimization
**Misc topics**

- Sometimes due to native support java performs better/worse at certain tasks
  - Can profile the running program to see what is being called frequently
- Pipelines in CPU
  - Run things at the same time
  - Exceptions, have to undo work
  - Out of order execution of code
  - Multiple CPU’s

**Final**

- Will hold extra office hours next week
- Please contact early for help