Announcements

• Project 0 is due tonight, 11:59pm
• Curator instructions have been posted on website
  – You should have received a password to submit – if not, contact vijayms@vt.edu right away!
• Please submit a correct tarball, following instructions
  – Avoids karma point deductions.

Overview for today

• Finish
  – Process States
• Process/Thread API Examples
• Fork/join model
• Time Slices

Process States

- Only 1 process (per CPU) can be in RUNNING state

Reasons for Preemption

- Generally two: quantum expired or change in priorities
- Reason #1:
  – A process of higher importance than the one that’s currently running has just become ready
- Reason #2:
  – Time Slice (or Quantum) expired
  – Default time slice in Linux is 100ms
- Question: what’s good about long vs. short time slices?

I/O Bound vs CPU Bound Procs

- Processes that usually exhaust their quanta are said to be CPU bound
- Processes that frequently block for I/O are said to be I/O bound
- Q.: what are examples of each?
  
- What policy should a scheduler use to juggle the needs of both?
Priority Based Scheduling

- Done in Linux, Windows, Pintos (after you complete Project 1), …

Only threads with the highest priority run
If more than one, round-robin

Higher Priority

6
3
2
MIN

Priority Based Scheduling (2)

- Advantage:
  - Dead simple: the highest-priority process runs
  - Q.: what is the complexity of finding which process that is?

- Disadvantage:
  - Not fair: lower-priority processes will never run
  - Hence, must adjust priorities somehow

- All schedulers used in today’s general purpose OS work like this
  - Only difference is how priorities are adjusted

Windows XP

- Thread state diagram in a industrial kernel

- Source: Dave Probert, Windows Internals – Copyright Microsoft 2003

Windows XP

- Source: Microsoft® Windows® Internals, Fourth Edition: Microsoft Windows Server™