Real Time Scheduling

Marie Roch
Tanenbaum 7.5
contains slides from: Tanenbaum 3rd ed. © 2008

Real time

• Two types:
  – hard real time
  – soft real time

• Scheduled processes
  – periodic
  – aperiodic
Hard Real Time

• Can we determine when a program or subroutine will stop in general?

• Sources of timing variability
  – virtual memory
  – disk access
  – scheduling

• Many hard time OS’s sacrifice some of these services that general purpose OS’s take for granted.

Scheduling periodic homogeneous processes

• Sample application: video server
  – fixed frame rate
  – multiple channels
  – stream data before next frame deadline

• FCFS sufficient provided that:
  – $t_{\text{frame}} \times \text{streams} < \text{frame period}$
  – scheduled at beginning of each period
Periodic processes

Is a set of periodic processes schedulable?

- \( C_i \): CPU time required
- \( P_i \): time between deadlines

\[
\sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1
\]

\[
\sum_{i=1}^{3} \frac{C_i}{P_i} = .808 \leq 1
\]

Schedulable?

We know that

\[
\sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1 \rightarrow \exists \text{ an algorithm that can schedule successfully}
\]

This does not necessarily imply that we know what which algorithm does this.
Static vs. dynamic scheduling

• Static
  – priority of each task is fixed
  – example: rate monotonic scheduling

• Dynamic
  – priorities of tasks may change
  – example: earliest deadline first

Rate monotonic scheduling (RMS)

• Assumptions
  – Task must complete within period
  – No dependencies between tasks
  – CPU burst requirements are static
  – Only periodic tasks may have deadlines
  – Preemption is without cost*

• Shown to be optimal for static schedulers
  [Liu and Layland, 1973]

* Clearly false
Rate monotonic scheduling

- Priority based on frequency
  e.g. period 40 ms \(\rightarrow \frac{1000}{40} = 25\) Hz

Note preemption of B in its 3\textsuperscript{rd} dispatch
Earliest deadline first (EDF) scheduling

- Dynamic priorities
- Aperiodic or periodic
- CPU bursts may vary, but must still be known

Earliest deadline first

- Deadlines
  - A: 30, 60, 90, 120
  - B: 40, 80, 120
  - C: 50, 100

- Queue sorted by deadline

---

Tanenbaum, p. 491
Earliest deadline first

Ready queue (process/deadline):
- At time 0: A₁/30, B₁/40, C₁/50, schedule A
- At time 25: C₁/50, schedule B
- At time 30: A₂/60, A₂ arrives, later deadline than C₁
- At time 40: A₂/60, B₂/80, B₂ arrives, later deadline than C₁

So far, same as rate monotonic

Tanenbaum, p. 491

Earliest deadline first

Ready queue (process/deadline):
- At time 90: empty
- A₄/120 arrives, B₃/120 is running

RMS would preempt, but
- deadlines tied
- EDF continues B to avoid context switch penalty

Tanenbaum, p. 491
A second example

- A’s CPU burst ↑ to 15ms every 30 ms
- Other processes as before:
  - B: 15 ms every 40 ms
  - C: 5 ms every 50 ms
- Are these schedulable?

\[ \sum_{i=1}^{3} \frac{C_i}{P_i} = 0.975 \leq 1 \]
Analysis of RMS failure

• RMS is a little simpler but,
  – failures can occur when CPU heavily loaded
  – It can be shown that RMS is guaranteed when:

\[
\sum_{i=1}^{m} \frac{C_i}{P_i} \leq m \left(2^{\frac{1}{m}} - 1\right)
\]

• EDF always works