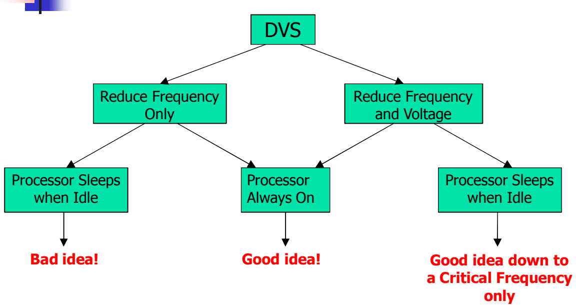
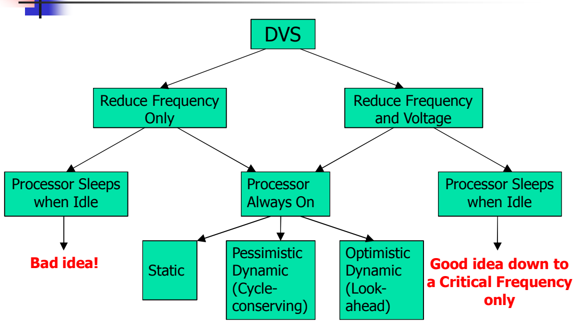
Energy

To Turn-off or not to Turn-off

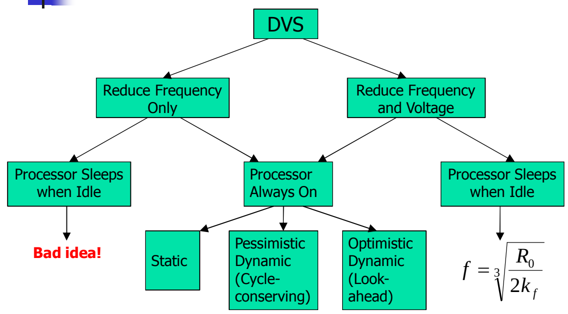
Recap



Recap



Recap



Turning Processors Off
The Cost of Wakeup

- Energy expended on wakeup, E_{wake}
- To sleep or not to sleep?

Turning Processors Off
The Cost of Wakeup

- Energy expended on wakeup, E_{wake}
- To sleep or not to sleep?
  - Not to sleep (for time t):

    $E_{no\text{-}sleep} = (k_v V^2 f + R_0)\, t$
  - To sleep (for time t) then wake up:

    $E_{sleep} = P_{sleep}\, t + E_{wake}$

Turning Processors Off
The Cost of Wakeup

- Energy expended on wakeup, E_{wake}
- To sleep or not to sleep?
  - Not to sleep (for time t):
    $E_{no\text{-}sleep} = (k_v V^2 f + R_0)\, t$
  - To sleep (for time t) then wake up:
    $E_{sleep} = P_{sleep}\, t + E_{wake}$
  - To save energy by sleeping: $E_{sleep} < E_{no\text{-}sleep}$

$$t > \frac{E_{wake}}{k_v V^2 f + R_0 - P_{sleep}}$$

Turning Processors Off
The Cost of Wakeup

- Energy expended on wakeup, E_{wake}
- To sleep or not to sleep?
  - Not to sleep (for time t):
    $E_{no\text{-}sleep} = (k_v V^2 f + R_0)\, t$
  - To sleep (for time t) then wake up:
    $E_{sleep} = P_{sleep}\, t + E_{wake}$
  - To save energy by sleeping: $E_{sleep} < E_{no\text{-}sleep}$

$$t > \frac{E_{wake}}{k_v V^2 f + R_0 - P_{sleep}}$$

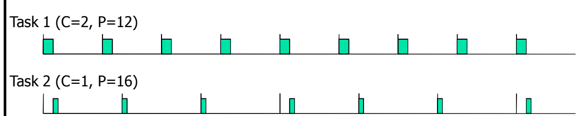
Minimum sleep interval

Dynamic Power Management

- DPM refers to turning devices off (or putting then in deep sleep modes)
- Device wakeup has a cost that imposes a minimum sleep interval (a breakeven time)
- DPM must maximize power savings due to sleep while maintaining schedulability

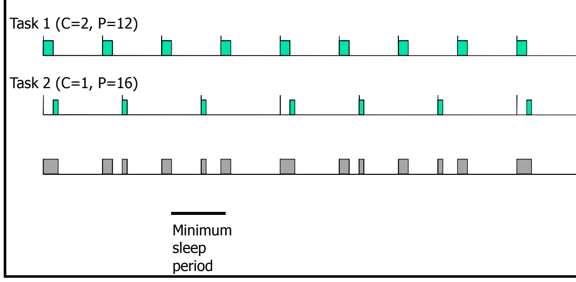
DPM and the Problem with Work-conserving Scheduling

- Example:



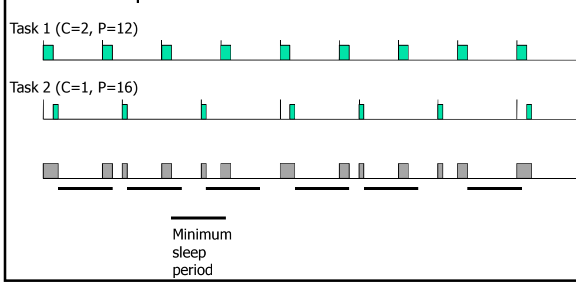
DPM and the Problem with Work-conserving Scheduling

- Example:



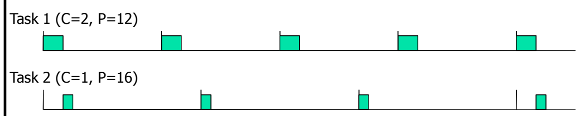
DPM and the Problem with Work-conserving Scheduling

- Example:



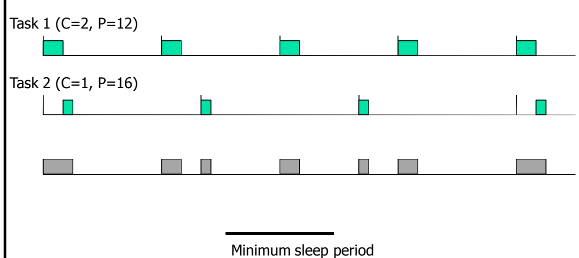
DPM and the Problem with Work-conserving Scheduling

- Example:



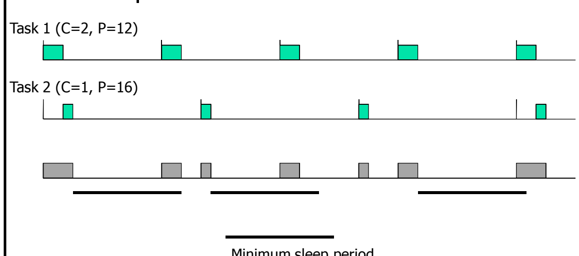
DPM and the Problem with Work-conserving Scheduling

- Example:



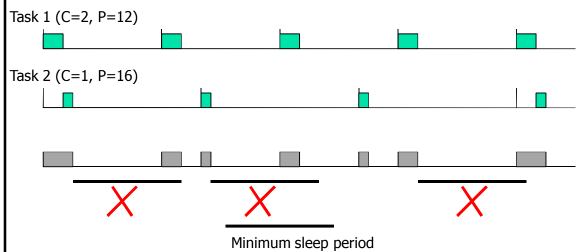
DPM and the Problem with Work-conserving Scheduling

- Example:



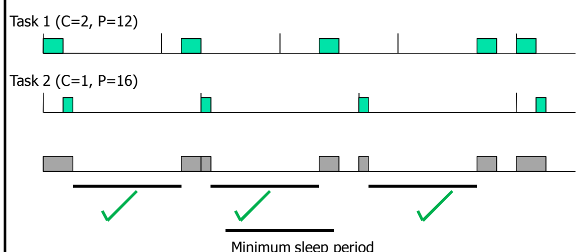
DPM and the Problem with Work-conserving Scheduling

- No opportunity to sleep ☹



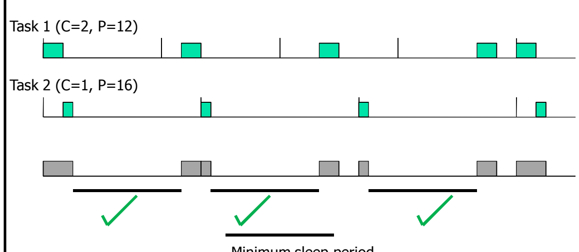
DPM and the Problem with Work-conserving Scheduling

- Must batch! ☺



A Schedulability Question:

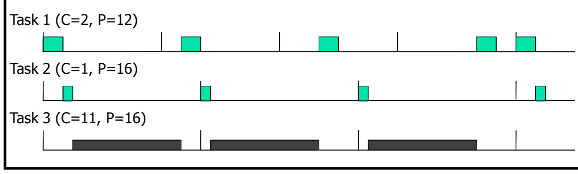
How to Analyze Schedules with Sleep Periods?



A Schedulability Question:

How to Analyze Schedules with Sleep Periods?

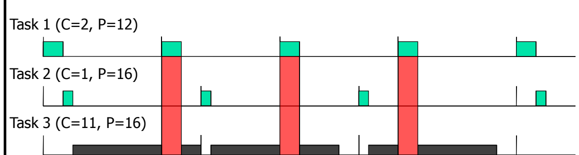
- Option 1: Treat sleep periods like a sporadic task. Use the Liu and Layland utilization bound for schedulability. Problems?



A Schedulability Question:

How to Analyze Schedules with Sleep Periods?

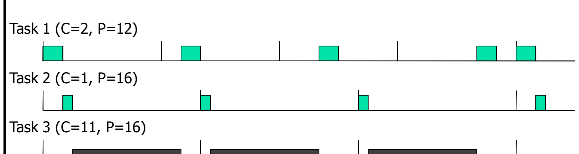
- Option 1: Treat sleep periods like a sporadic task. Use the Liu and Layland utilization bound for schedulability. Problems?
  - Does not work because the "sleep task" cannot be preempted, whereas the rest of the tasks are preemptible. The utilization bound works only for fully preemptive scheduling.



A Schedulability Question:

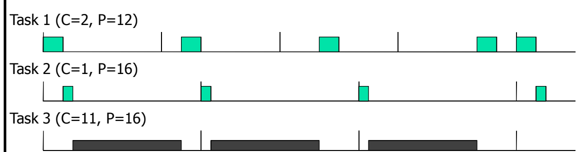
How to Analyze Schedules with Sleep Periods?

- Option 2: Treat sleep periods like the *highest-priority* sporadic task. Use the Liu and Layland utilization bound for schedulability. Problems?



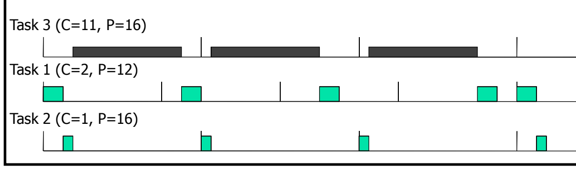
A Schedulability Question:
How to Analyze Schedules with Sleep Periods?

- Option 2: Treat sleep periods like the *highest-priority* sporadic task. Use the Liu and Layland utilization bound for schedulability. Problems?
  - Does not work because the "sleep task" may need to have a larger period than the actual top-priority task, which contradicts rate-monotonic scheduling. The bound does not work.



A Schedulability Question:
How to Analyze Schedules with Sleep Periods?

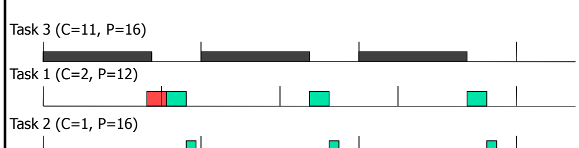
- Option 3: Treat sleep periods like the *highest-priority* sporadic task. Use *response time analysis* for schedulability. Problems?



Device Forbidden Regions

- Option 3: Treat sleep periods like the *highest-priority* sporadic task. Use *response time analysis* for schedulability. Problems?
  - A Valid solution, but pessimistic.

  (Called: Device Forbidden Regions. Published in RTAS 2008.)



DVS on Homogeneous Multiprocessors

- Consider a set of tasks, where task i has period P_i and total number of cycles C_i
  - Sort tasks from largest to smallest utilization C_i / P_i
  - Assign tasks one at a time (largest-first) to the least utilized processor
  - Apply one of the previous algorithms on each processor separately

Question

- From the perspective of minimizing energy, is it always a good idea to use up all processors?

How Many Processors to Use?

- Consider using one processor at frequency f versus two at frequency $f/2$
- Case 1: Total power for one processor
  - $k_f f^3 + R_0$
- Case 2: Total power for two processors
  - $2 \{k_f (f/2)^3 + R_0\} = k_f f^3 / 4 + 2 R_0$

How Many Processors to Use?

- Consider using one processor at frequency f versus two at frequency $f/2$
- Case 1: Total power for one processor
  - $k_f f^3 + R_0$
- Case 2: Total power for two processors
  - $2 \{k_f (f/2)^3 + R_0\} = k_f f^3 / 4 + 2 R_0$
- The general case: n processors
  - $n \{k_f (f/n)^3 + R_0\} = k_f f^3 / n^2 + n R_0$

How Many Processors to Use?

- The general case: n processors
  - $Power = n \{k_f (f/n)^3 + R_0\} = k_f f^3 / n^2 + n R_0$
  - $dPower/dn = -2 k_f f^3 / n^3 + R_0 = 0$

$$n = \sqrt[3]{\frac{2k_f f^3}{R_0}}$$

How Many Processors to Use?

- The general case: n processors
  - $Power = n \{k_f (f/n)^3 + R_0\} = k_f f^3 / n^2 + n R_0$
  - $dPower/dn = -2 k_f f^3 / n^3 + R_0 = 0$

$$n = \sqrt[3]{\frac{2k_f f^3}{R_0}}$$

- What if n is not an integer?

Advanced Configuration and Power Interface (ACPI Standard)

- C-states (idle states)
- P-states (dynamic voltage and frequency scaling)

Turning a Processor Off