Scheduling Algorithm and Analysis

RT Synchronization Protocol
(Module 35)

Yann-Hang Lee
Arizona State University
yhlee@asu.edu
(480) 727-7507

Summer 2014
A task $T_H$ can be blocked by a lower-priority task $T_L$ in three ways:

- directly, i.e.

```
T_H ----> X ----> T_L  
```

request for
allocated to

- when $T_L$ inherits a priority higher than the priority $\pi_H$ of $T_H$.

```
T \neq T_H
```

$\pi > \pi_H$

- When $T_H$ requests a resource the priority ceiling of resources held by $T_L$ is equal to or higher than $\pi_H$.

```
T_H ----> Y ----> X ----> T_L  
```

$\pi_H \leq \Pi_X$
Blocked At Most Once (PCP)

$\tau_1:\{\ldots P(S1)\ldots P(S2)\ldots V(S2)\ldots V(S1)\ldots \}$

$\tau_2:\{\ldots P(S1)\ldots V(S1)\ldots \}$

$\tau_3:\{\ldots P(S2)\ldots V(S2)\ldots \}$

S1 locked

S1 unlocked

S2 locked

S2 unlocked

attempts to lock S1(blocked)

attempts to lock S1(blocked)

S1 locked

S1 unlocked

S2 locked

S2 unlocked
Deadlock Avoidance: Using PCP

\[ \tau_1 : \{ ...P(S1) ...P(S2) ...V(S2) ...V(S1) ... \} \]

\[ \tau_2 : \{ ...P(S2) ...P(S1) ...V(S1) ...V(S2) ... \} \]
Stack Sharing

Sharing of the stack among tasks eliminates stack space fragmentation and so allows for memory savings:

However:
- Once job is preempted, it can only resume when it returns to be on top of stack.
- Otherwise, it may cause a deadlock.
- Stack becomes a resource that allows for “one-way preemption”.

(no stack sharing)  (stack sharing)
Stack-Based Priority Ceiling Protocol

- **To avoid deadlocks**: Once execution begins, make sure that job is not blocked due to resource access
  - allow preemption only if the priority is higher than the ceiling of the resources in use

- **Update Current Ceiling in the usual manner**
  - If no resource allocated, $\Pi_S(t) = \Omega$

- **Scheduling Rule:**
  - $J_i$ released and blocked until $\pi_i(t) > \Pi_S(t)$
  - When not blocked jobs are scheduled in the usual manner.

- **Allocation Rule:**
  - Allocate when requested
Stack-Based PCP (cont)

- The Stack-Based Priority-Ceiling Protocol is deadlock-free:
  - When a job begins to execute, all the resources it will ever need are free.
  - Otherwise, $\Pi_S(t)$ would be higher or equal to the priority of the job.
  - Whenever a job is preempted, all the resources needed by the preemption are free.
  - The preempting job can complete, and the preempted job can resume.

- Worst-case blocking time of Stack-Based Protocol is the same as for Basic Priority Ceiling Protocol.

- Stack-Based Protocol smaller context-switch overhead
  - 2 context switches since once execution starts, job cannot be blocked (may be preempted)
  - 4 context switches for PCP since a job may be blocked at most once
Supplementary Slides