

Sistemas de Tempo-Real

Escalonamento: Tarefas Dependentes

- O problema da inversão de prioridades
- Protocolo de Herança de prioridades
- Protocolo “Priority Ceiling”

Sistemas de Tempo-Real

Escalonamento: Tarefas Dependentes

Compartilhamento de Recurso

Mecanismos de exclusão mútua

O compartilhamento de recursos e as relações de exclusão decorrentes do mesmo, levam a bloqueios em tarefas mais prioritárias, conhecidos como *inversão de prioridades*

Sistemas de Tempo-Real

Escalonamento: Tarefas Dependentes

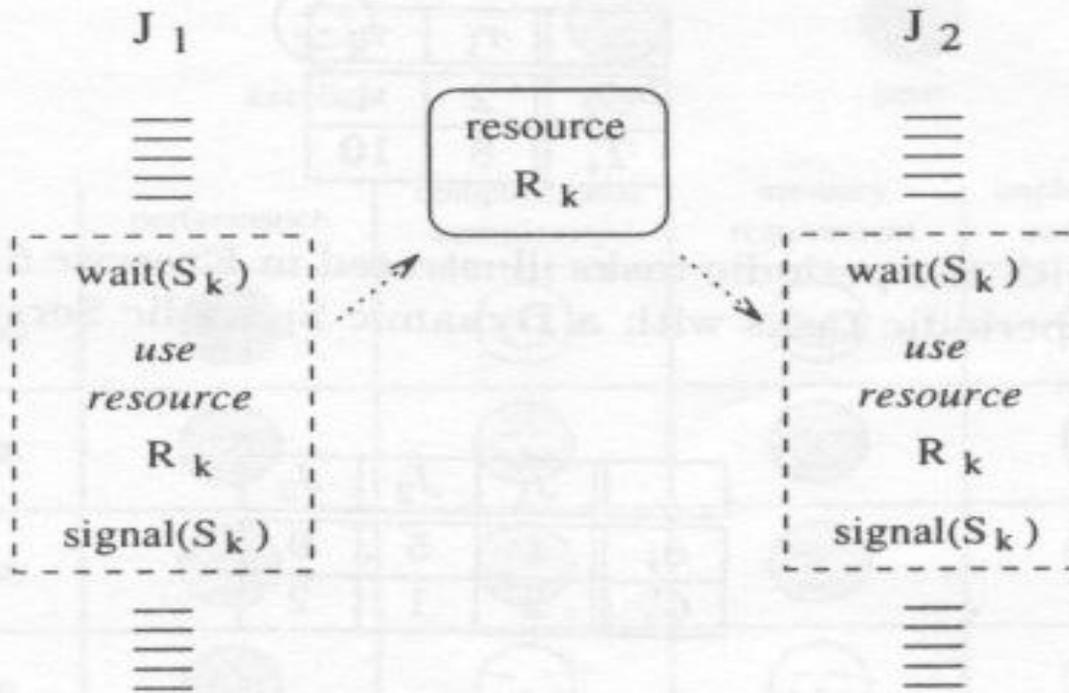
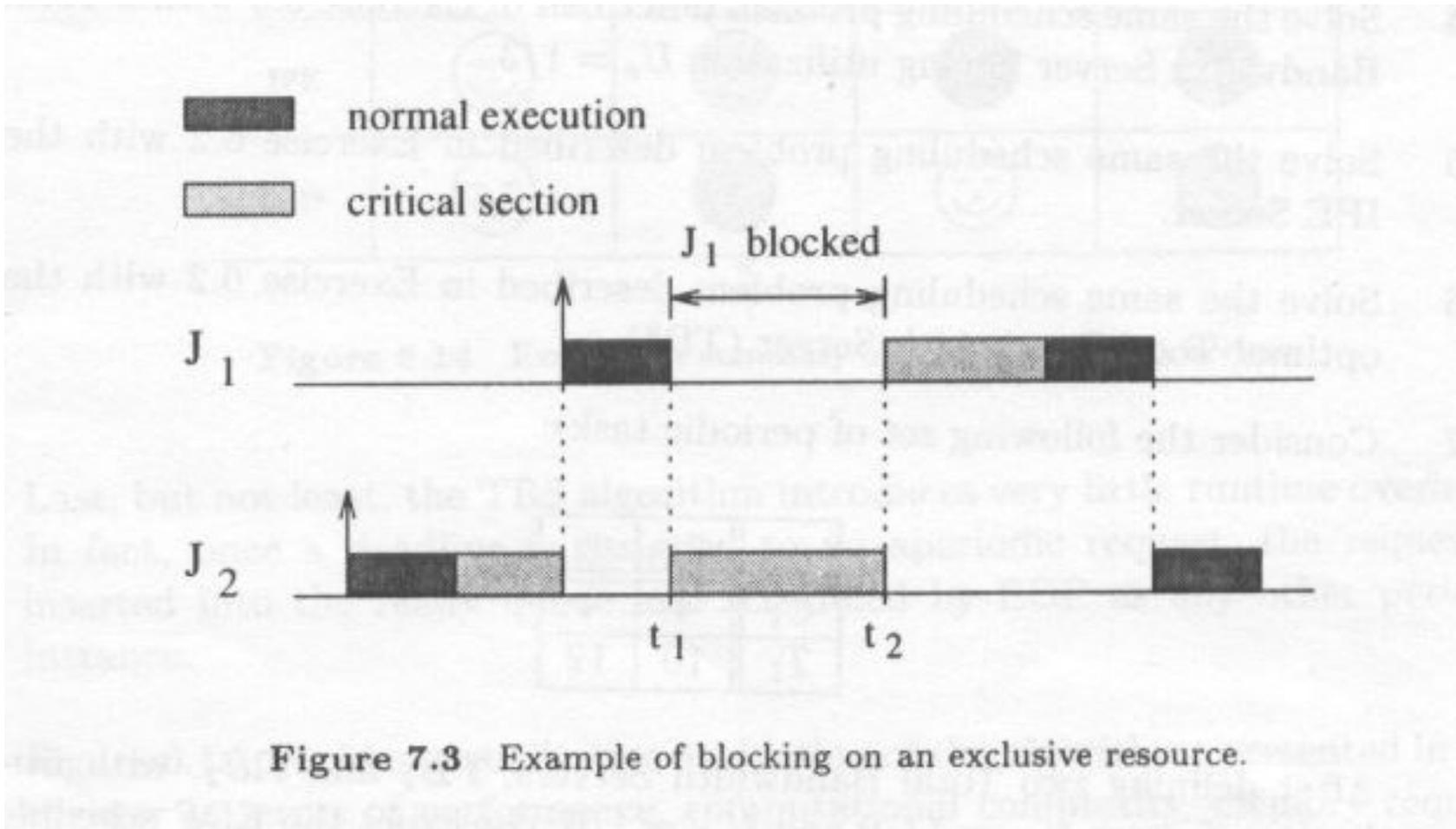


Figure 7.2 Structure of two tasks that share an exclusive resource.

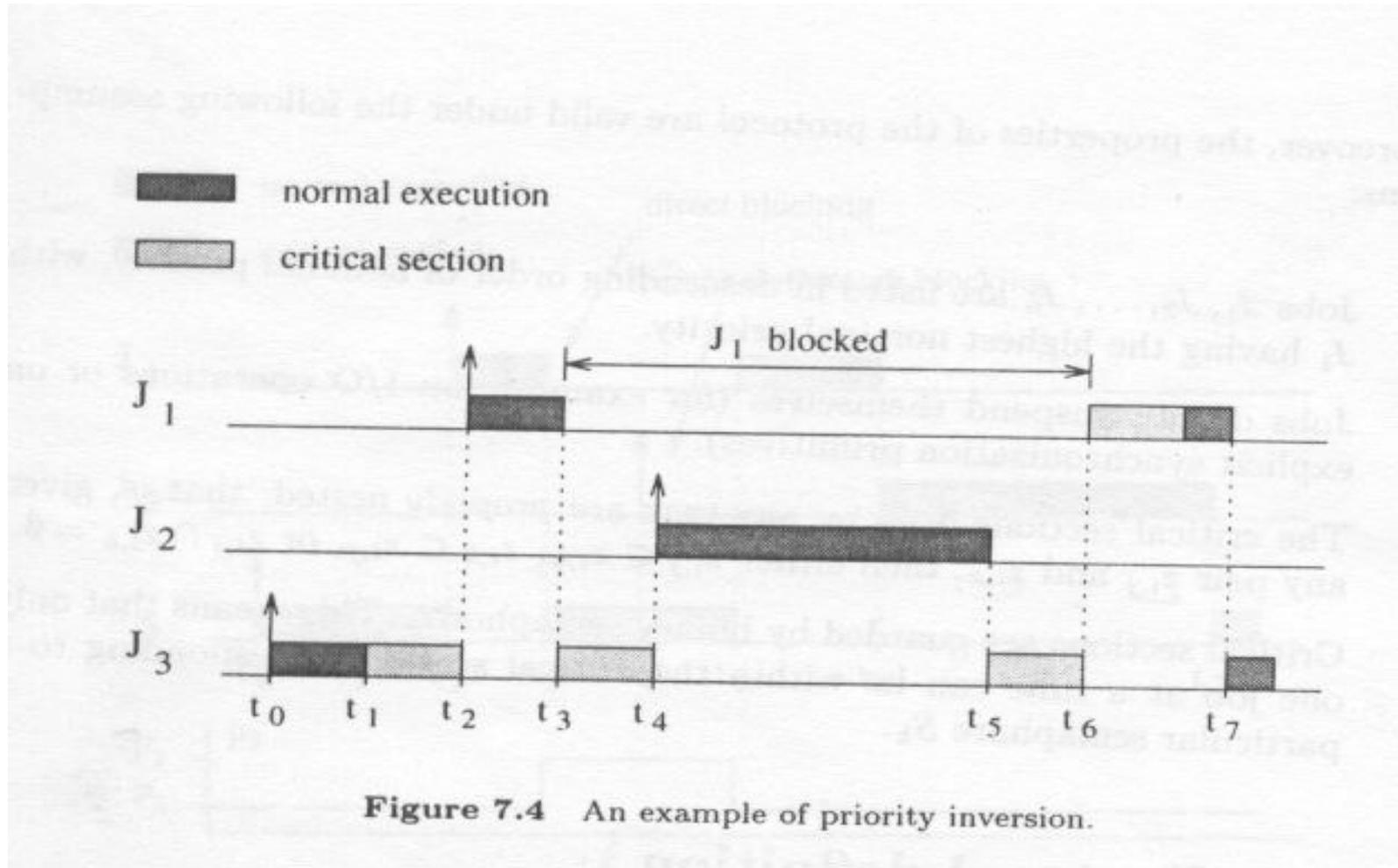
Sistemas de Tempo-Real

Escalonamento: Tarefas Dependentes



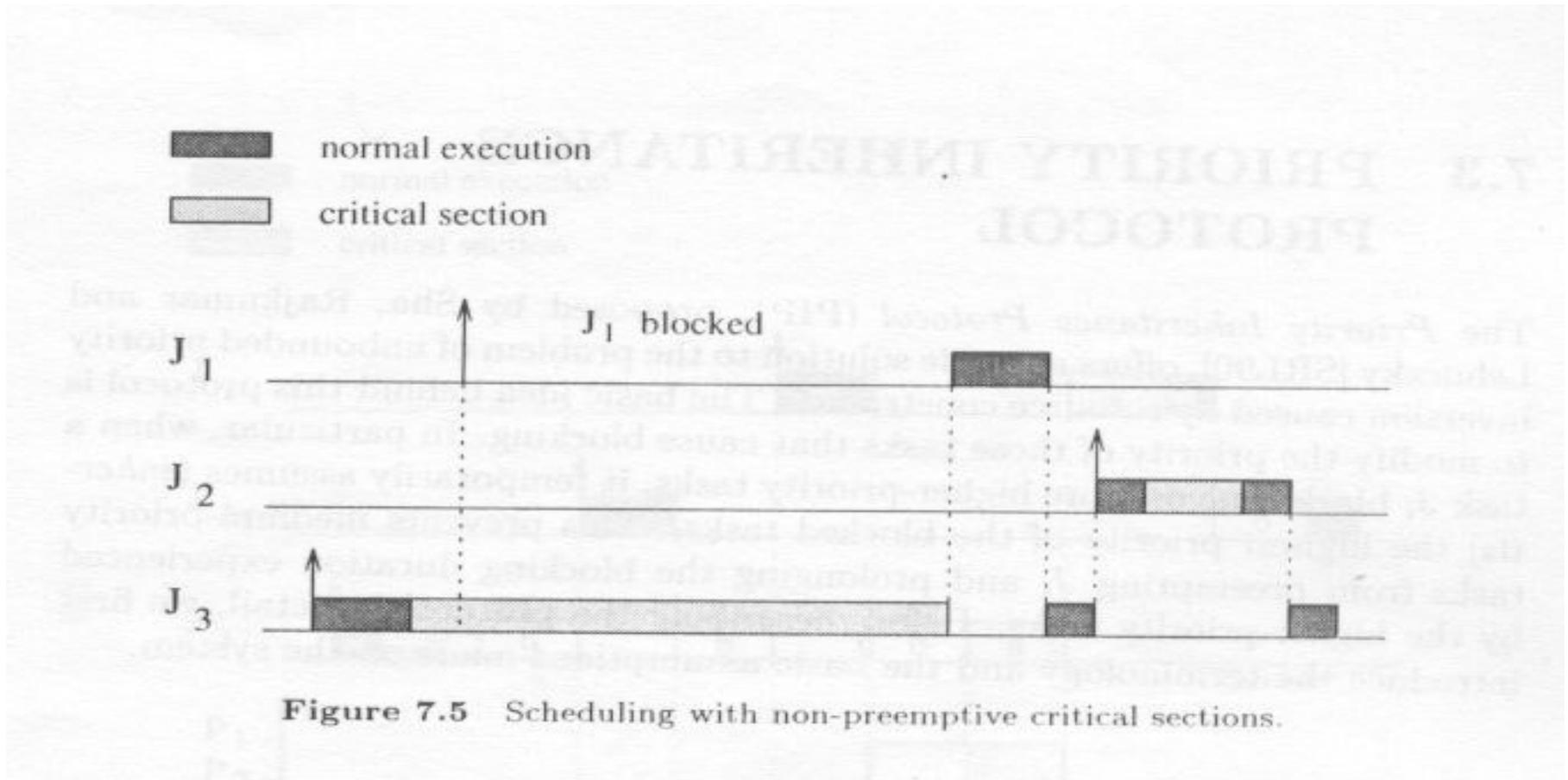
Sistemas de Tempo-Real

Escalonamento: Inversão de prioridades



Sistemas de Tempo-Real

Escalonamento: Inversão de prioridades



Desabilitar preempção

Sistemas de Tempo-Real

Escalonamento: Protocolos de controle

O Protocolo de Herança de Prioridade (PHP)

Este protocolo tem como idéia principal a modificação da prioridade daquelas tarefas que causam o bloqueio.

Particularmente, quando uma tarefa T_i bloqueia uma ou mais tarefas de maior prioridade, ela temporariamente assume (herda) a prioridade mais alta das tarefas, prevenindo a preempção de T_i por tarefas de prioridade média, o que prolongaria a duração do bloqueio das tarefas de mais alta prioridade.

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

Terminologia

Considere um conjunto de n tarefas, T_1, T_2, \dots, T_n , que cooperam através de m recursos, R_1, R_2, \dots, R_m . Cada tarefa é caracterizada por um período P_i e um tempo de computação C_i . A *deadline* de qualquer tarefa periódica é o final do período. Cada recurso R_K é guardado por um semáforo S_K . A seguinte notação é adotada:

Pr_i prioridade *nominal*

$SC_{i,j}$ j -ésima SC de T_i

p_i prioridade *ativa*

$d_{i,j}$ duração de $SC_{i,j}$

$S_{i,j}$ semáforo de $SC_{i,j}$

$R_{i,j}$ recurso de $SC_{i,j}$

$SC_{i,j} \subset SC_{i,k}$ indica que $SC_{i,j}$ está contido em $SC_{i,k}$

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

Abordagem

Tarefas são escalonadas com base nas suas prioridades ativas. Tarefas com a mesma prioridade são executadas conforme FCFS.

Quando uma tarefa T_i tenta entrar em uma seção crítica $SC_{i,j}$ e o recurso $R_{i,j}$ já está sendo usado por uma tarefa de mais baixa prioridade, T_i será bloqueado pela tarefa que detém o recurso, senão T_i entra na SC.

Quando uma tarefa T_i é bloqueada em um semáforo, ela cede sua prioridade ativa para a tarefa, T_K , que tem o semáforo. Desta forma, T_K continua e executa o resto da SC com prioridade $p_K = p_i$, ou seja, T_K herda a prioridade de T_i .

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

Abordagem

Quando T_K sai da SC ele libera o semáforo e a tarefa de prioridade mais alta, é acordada. Além disto, a prioridade da tarefa T_K será atualizada seguindo o seguinte: se nenhuma outra tarefa está bloqueada por T_K , p_K será igual ao seu valor nominal, senão será igual a prioridade mais alta das tarefas bloqueadas em T_K .

A herança de prioridades é transitiva, ou seja, se T3 bloqueia T2 e T2 bloqueia T1, então T3 herda a prioridade de T1 através de T2.

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

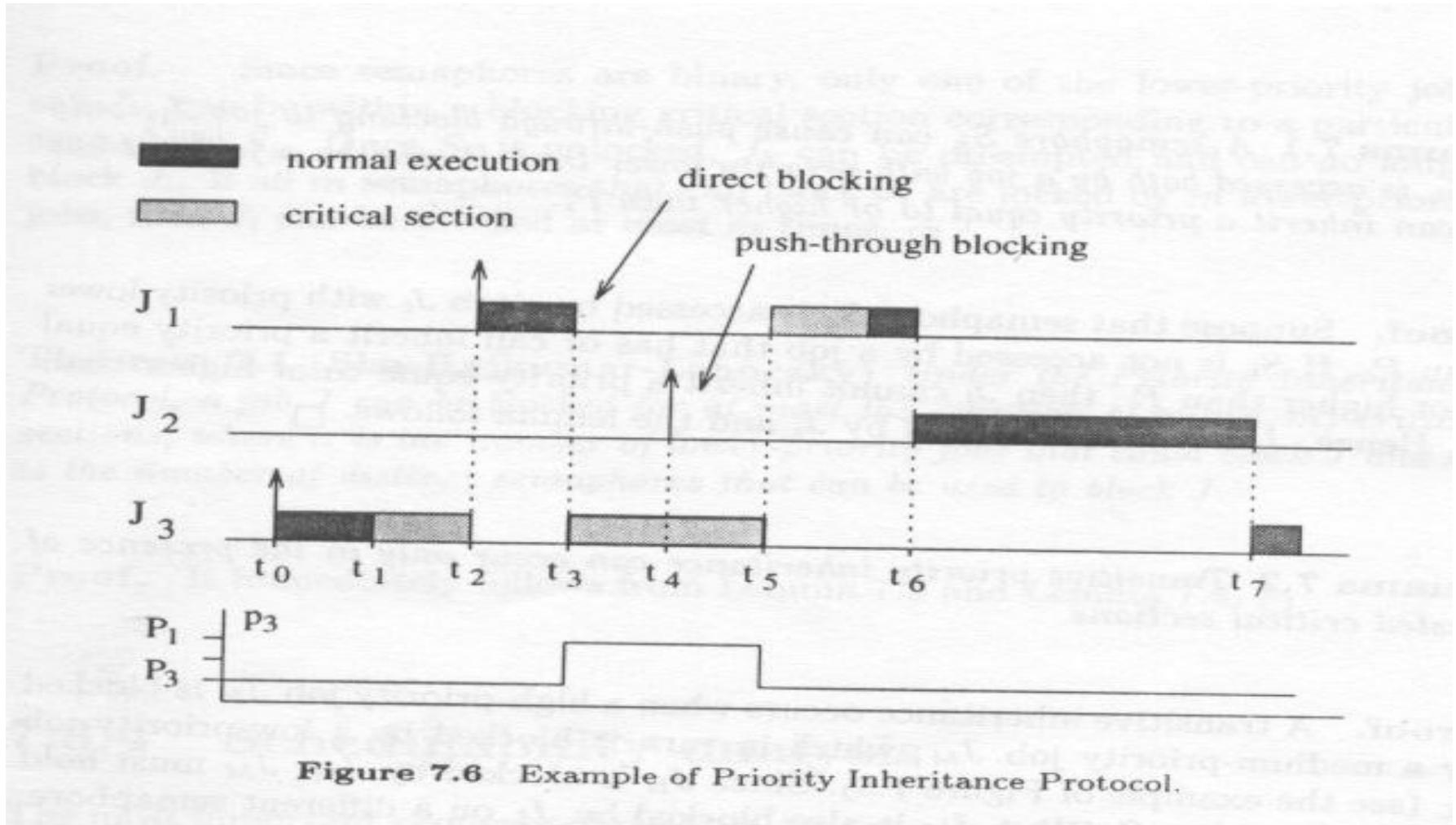


Figure 7.6 Example of Priority Inheritance Protocol.

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

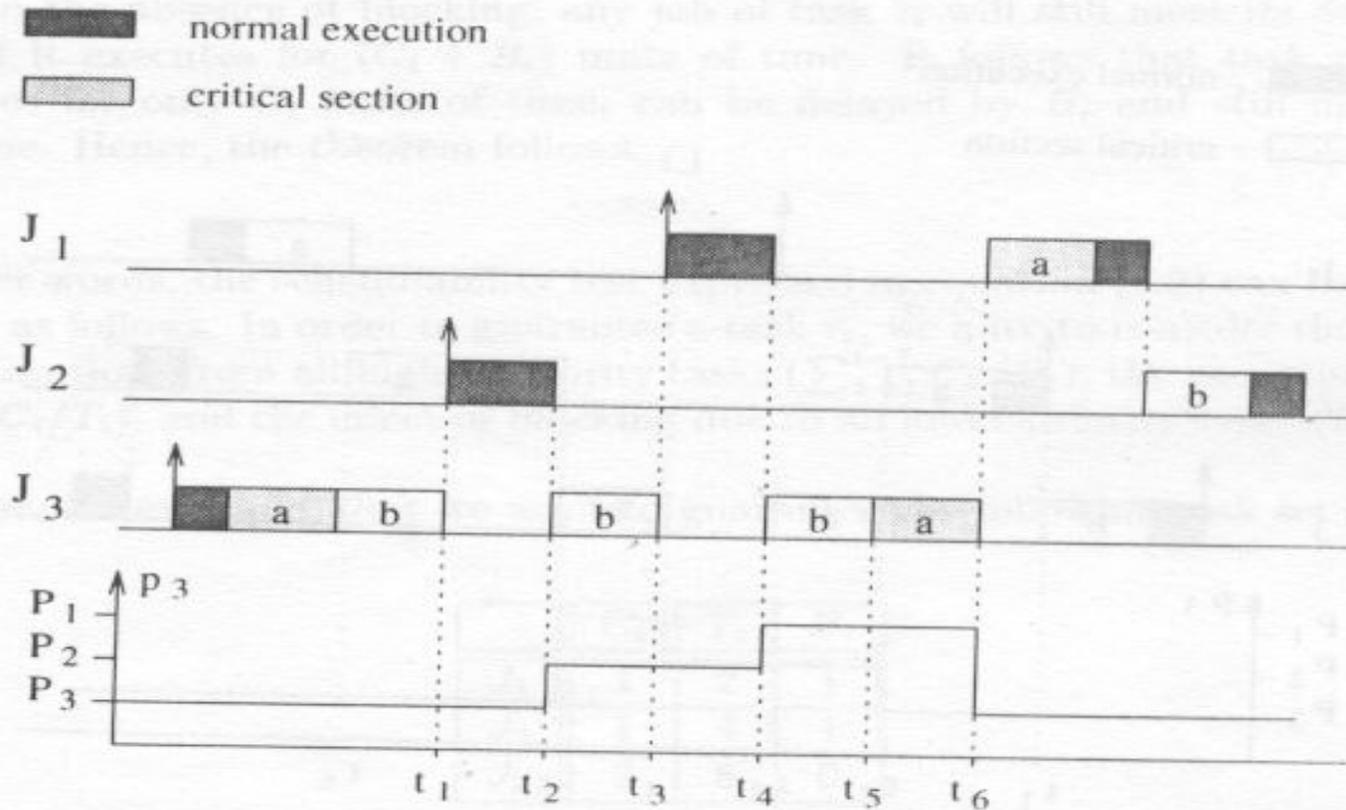


Figure 7.7 Priority inheritance with nested critical sections.

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

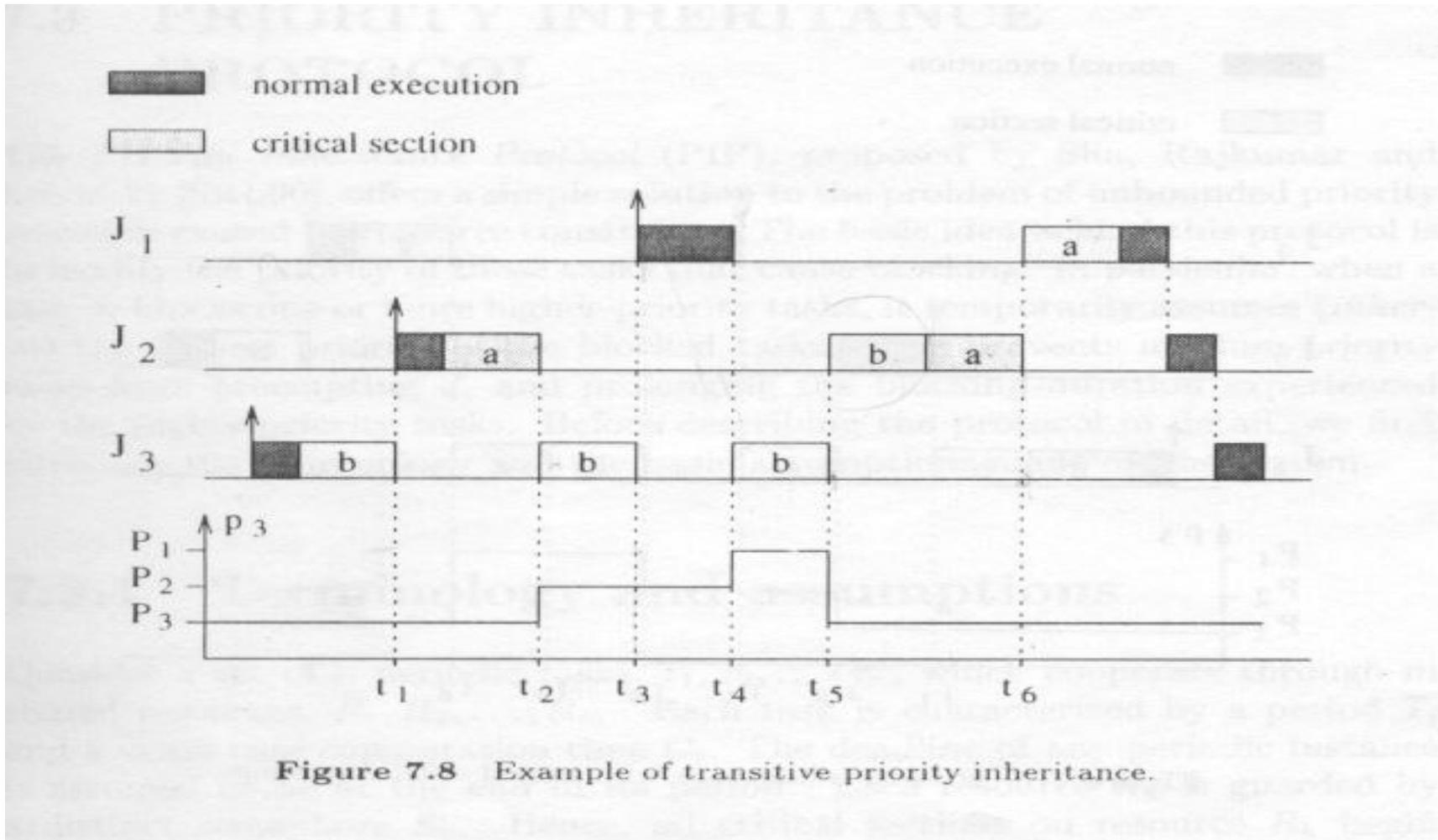


Figure 7.8 Example of transitive priority inheritance.

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

Detalhes de Implementação

A implementação do PHP implica em algumas modificações nas estruturas de dados do kernel associadas as tarefas e semáforos.

1. toda tarefa deve ter uma prioridade nominal e uma ativa
2. cada semáforo tem um campo do tipo *bloqueado por*
3. na tarefa, campo no TCB, *bloqueado em* , simplifica o caso de bloqueio transitivo

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

Detalhes de Implementação

Ph_wait(s)

- Se o semáforo esta livre, torna o mesmo bloqueado e o nome da tarefa executando fica armazenada no campo *bloqueado por* do descritor do semáforo.
- Se o semáforo esta bloqueado, a tarefa executando é bloqueada na fila do semáforo, o ID do mesmo é colocado no campo *bloqueado em* da tarefa (TCB) e a sua prioridade é herdada pela tarefa que bloqueia o semáforo. Se esta tarefa esta bloqueada em outro semáforo, a regra da transitividade deve ser aplicada. A tarefa pronta com prioridade mais alta volta executar.

Sistemas de Tempo-Real

Escalonamento: Protocolo PHP

Detalhes de Implementação

Ph_signal(s)

- Se a fila do semáforo esta vazia, o mesmo é desbloqueado.
- Se a fila do semáforo não esta vazia, a tarefa de mais alta prioridade que esta na fila é acordada, seu ID colocado no campo *bloqueado por* de *s*, a prioridade ativa da tarefa executando é atualizada e a tarefa pronta com prioridade mais alta volta executar.

Sistemas de Tempo-Real

Escalonamento: Tarefas Dependentes

Problemas

Embora o PHP limite o problema da inversão de prioridade, a duração de um bloqueio de uma tarefa pode ainda ser significativo devido a formação de cadeias de bloqueio. Outro problema do PHP é que ele não previne contra deadlocks.

Sistemas de Tempo-Real

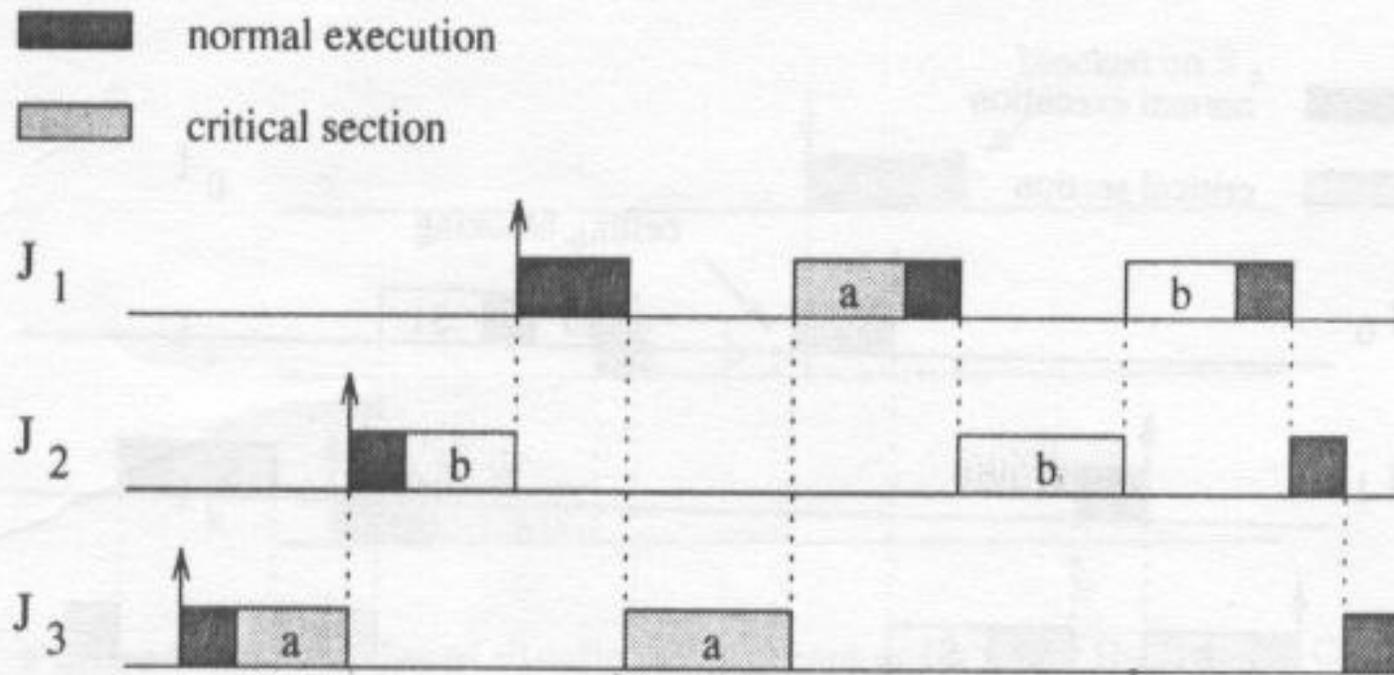


Figure 7.10 Example of chained blocking.

Sistemas de Tempo-Real

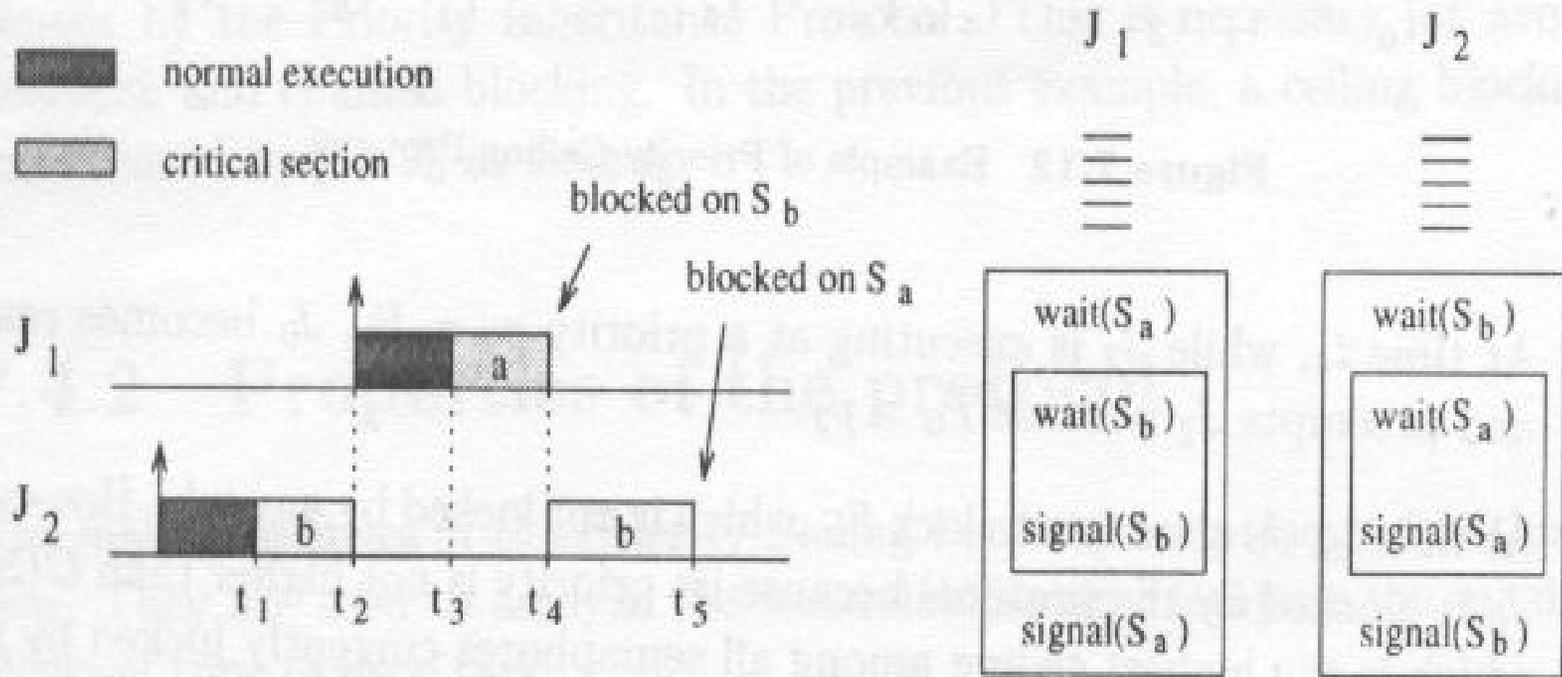


Figure 7.11 Example of deadlock.

Sistemas de Tempo-Real

Escalonamento: Protocolo de Acesso

O Protocolo de Prioridade Teto (Priority Ceiling Protocol)

Na abordagem PCP a idéia central é limitar o problema de inversão de prioridade e evitar a formação de deadlocks e cadeias de bloqueios. O PCP é dirigido para escalonamento de prioridade fixa, com RM.

Ele é uma extensão do PHP no sentido de que ele adiciona uma regra de controle sobre os pedidos de entrada em exclusão mútua.

Sistemas de Tempo-Real

Escalonamento: Protocolo Prioridade Teto

O Protocolo de Prioridade Teto (Priority Ceiling Protocol)

Para evitar o bloqueio múltiplo, a regra não permite que uma tarefa entre em uma SC se existem semáforos bloqueados que podem bloquear a tarefa, ou seja, uma vez que a tarefa entra na sua 1ª. SC, ela não pode ser bloqueada por tarefas de prioridade mais baixa até terminar. Para por esta idéia em pratica, cada semáforo tem uma *prioridade teto* associada que é igual a prioridade da tarefa de maior prioridade que pode pegar o semáforo. Assim, uma tarefa T tem permissão de entrar em uma SC somente se sua prioridade é maior que todas as prioridades teto dos semáforos pertencentes a outras tarefas diferentes de T.

Sistemas de Tempo-Real

Escalonamento: Protocolo Prioridade Teto

O Protocolo de Prioridade Teto – Definição

Cada semáforo S_K tem uma prioridade teto (“ceiling”) $C(S_K)$ = prioridade da tarefa de mais alta prioridade que pode pegar o semáforo. $C(S_K)$ é um valor estático calculado em off-line.

Se T_i é a tarefa de mais alta prioridade (pronta) então T_i recebe o processador.

S^* é o semáforo com a mais alta prioridade teto entre todos os semáforos alocados por outras tarefas diferentes de T_i e $C(S^*)$ é o seu teto.

Para entrar em uma SC guardada por um semáforo S_K , T_i deve ter uma prioridade mais alta que $C(S^*)$. Se $P_i \leq C(S^*)$, $\text{wait}(S_K)$ é negado e T_i é considerado como sendo bloqueado no semáforo S^* pela tarefa que detém S^* .

Sistemas de Tempo-Real

Escalonamento: Protocolo Prioridade Teto

O Protocolo de Prioridade Teto – Definição

- Quando uma tarefa T_i é bloqueada em um semáforo, ela transmite sua prioridade para a tarefa, T_K , que tem o semáforo. Desta forma, T_K continua e executa o resto da SC com a prioridade de T_i , ou seja, T_K herda a prioridade de T_i .

Quando T_K sai da SC ela libera o semáforo e a tarefa de maior prioridade bloqueada no semáforo é acordada. Além disto, a prioridade da tarefa T_K será atualizada seguindo o seguinte: se nenhuma outra tarefa está bloqueada por T_K , P_K será igual ao seu valor nominal, senão será igual a prioridade mais alta das tarefas bloqueadas em T_K .

A herança de prioridades é transitiva, ou seja, se T_3 bloqueia T_2 e T_2 bloqueia T_1 , então T_3 herda a prioridade de T_1 através de T_2 .

Sistemas de Tempo-Real

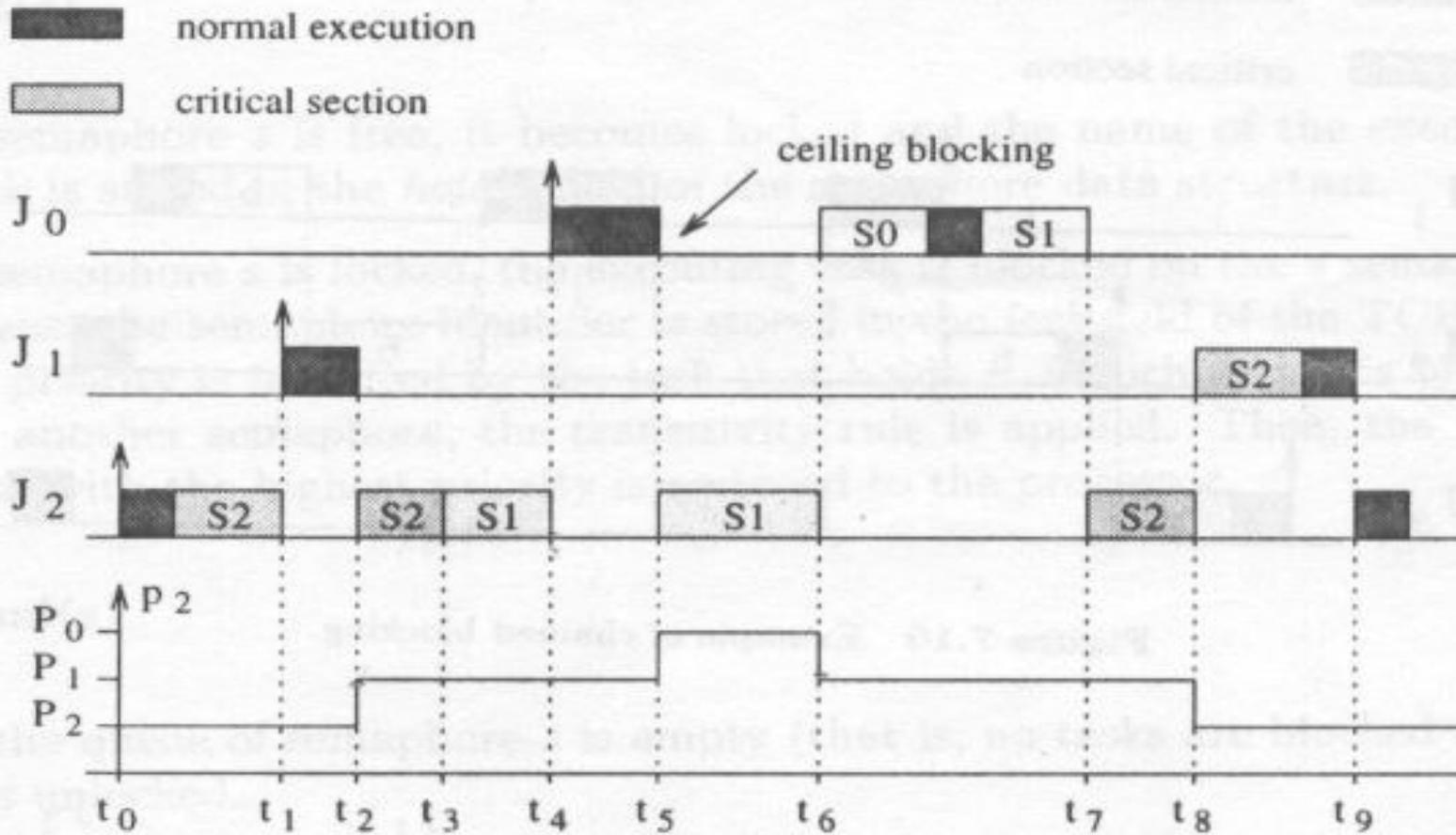


Figure 7.12 Example of Priority Ceiling Protocol.

Sistemas de Tempo-Real

Escalonamento: Protocolo Prioridade Teto

Detalhes de Implementação

Em relação a implicação do PCP nas estruturas de dados do kernel, verifica-se que as filas de semáforos não são necessárias visto que as tarefas bloqueadas (pelo protocolo) podem ser mantidas na fila de prontos. Sempre que uma tarefa é bloqueada em um semáforo, a tarefa que detém o semáforo herda a prioridade da tarefa bloqueada e recebe o processador, enquanto que a tarefa bloqueada volta para a fila de prontos. Assim que a tarefa libera o semáforo, sua prioridade é atualizada e se ela for menor que a prioridade da 1^a. tarefa na fila de prontos, ela é preemptada.

Sistemas de Tempo-Real

Escalonamento: Protocolo Prioridade Teto

Detalhes de Implementação

cada semáforo deve armazenar o identificador da tarefa que detém o mesmo e a prioridade teto do semáforo.

no descritor da tarefa, TCB, um campo deve armazenar a prioridade ativa da mesma.

prever um campo na TCB para guardar a identificação do semáforo no qual a tarefa esta bloqueada

Manter uma lista atual dos semáforos bloqueados, em ordem decrescente de prioridades teto. Útil na verificação da prioridade máxima que uma tarefa deve obedecer para entrar em uma SC e também na atualização das prioridades ativas das tarefas na saída das SCs.

Sistemas de Tempo-Real

Escalonamento: Protocolo Prioridade Teto

Detalhes de Implantação

$P_c_wait(s)$

Busca o semáforo S^* com o teto máximo C^* entre os que estão bloqueados por outras tarefas que não a executando T_{exe} .

Se $p_{exe} \leq C^*$, transfere P_{exe} para a tarefa que detém S^* , insere T_{exe} na fila de pontos e executa a tarefa com prioridade mais alta.

Se $p_{exe} > C^*$, ou sempre que s é liberado, bloqueia o semáforo s , insere s na lista dos bloqueados e coloca P_{exe} no campo identificador da tarefa do semáforo s .

Sistemas de Tempo-Real

Escalonamento: Protocolo Prioridade Teto

Detalhes de Implementação

Pc_signal(s)

Retira s da lista de bloqueados.

Se não tem outras tarefas bloqueadas por T_{exe} , faz $p_{exe} = P_{exe}$, senão faz $p_{exe} =$ prioridade mais alta das tarefas bloqueadas por T_{exe} .

Faz p^* a prioridade mais alta entre as tarefas prontas. Se $p_{exe} < p^*$, insere T_{exe} na fila de prontos e executa a tarefa mais prioritária da fila.