

---

## Sistemas de Tempo Real: Multiprocessadores: Exclusão Mútua em Sistemas Particionados

Rômulo Silva de Oliveira  
Departamento de Automação e Sistemas – DAS – UFSC

romulo@das.ufsc.br  
http://www.das.ufsc.br/~romulo  
Março/2011

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 1

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 4

---

### Referências

- MPCP (prioridade fixa)
  - R. Rajkumar, L. Sha, J. P. Lehoczky, "Real-time synchronization protocols for multiprocessors". In proceedings of the Real Time Systems Symposium, pp. 259-269, 1988.
  - R. Rajkumar, "Real-Time Synchronization Protocols for Shared Memory Multiprocessors". In proc. of the Tenth International Conference on Distributed Computing Systems, 1990.
  - K. Lakshmanan I, D. de Niz, R. Rajkumar, "Coordinated Task Scheduling, Allocation and Synchronization on Multiprocessors". In Proc. of the Real Time Systems Symposium, pp. 469-478, 2009.
- MDPCP (MPCP para EDF)
  - C-M. Chen, S.K. Tripathi, A. Blackmore, "A Resource Synchronization Protocol for Multiprocessor Real-Time Systems". In Proceedings of the International Conference on Parallel Processing, pp. 159-162, 1994.
- MSRP (prioridade fixa e EDF)
  - P. Gai, G. Lipari, M. Di Natale, "Minimizing memory utilization of real-time task sets in single and multi-processor systems-on-a-chip." In Proceedings of the Real-Time Systems Symposium, pp. 73-83, 2001.
- Comparação entre MPCP e MSRP
  - P. Gai, M.D. Natale, G. Lipari, A. Ferrari, C. Gabellini, P. Marceca, "A comparison of MPCP and MSRP when sharing resources in the Janus multiple-processor on a chip platform". In Proceedings of the Real-Time and Embedded Technology and Applications Symposium 2003.

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 2

---

### MPCP – Multiprocessor Priority Ceiling Protocol

- Em (Rajkumar, Sha and Lehoczky, 1988) é apresentado um protocolo para compartilhamento de recursos em multiprocessadores onde:
  - cada recurso compartilhado global é controlado a partir de um processador específico designado para ele.
- Múltiplos processadores possam controlar diferentes recursos globais
- Apenas um processador controla todos os acessos a um dado recurso global
- Se uma tarefa requer o uso de um recurso global, ela deve
  - enviar um pedido para o processador controlador daquele recurso
  - e obter os resultados
- Este protocolo é particularmente apropriado para arquiteturas baseadas em mensagem
- Mas este protocolo não aproveita a disponibilidade de memória compartilhada globalmente acessível (arquiteturas multicore)

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 5

- 
- Recent studies [14, 15] have investigated the performance differences between spin-based and suspension-based synchronization protocols
  - [14] B.B.Brandenburg, J.M.Calandrino, A.Block, H.Leontyev, and J.H.Anderson, "Real-time synchronization on multiprocessors: To block or not to block, to suspend or spin?" RTAS, pp. 342–353, 2008
  - [15] B.B.Brandenburg and J.H.Anderson, "A comparison of the mpcp, d-ppcp, and fmlp on litmusRT," Proceedings of the 12th International Conference on Principles of Distributed Systems, pp. 105–124, 2008
  - Coordinated task scheduling critica os dois

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 3

---

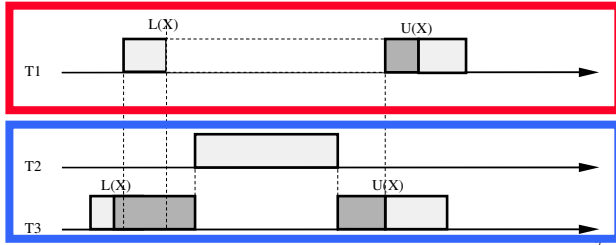
### MPCP – Multiprocessor Priority Ceiling Protocol

- Em (Rajkumar, 1990) é apresentado um protocolo de sincronização baseado em prioridades fixas que usa explicitamente memória compartilhada
  - Extensão do protocolo apresentado em (Rajkumar, Sha and Lehoczky, 1988)
- Em cada processador, uma tarefa pode ser
  - preemptada por uma tarefa de mais alta prioridade executando localmente
  - bloqueada por tarefas de mais baixa prioridade executando localmente
  - atrasada por tarefas de qualquer prioridade que executam remotamente mas compartilham recursos globais com a tarefa em questão
- Interferência local
- Bloqueio local
- Bloqueio remoto
- O objetivo de um protocolo de sincronização é minimizar e limitar o bloqueio remoto

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 6

### Bloqueio Remoto 1/3

- T1 sofre bloqueio remoto de T3 por compartilhar mutex X
- T2 preempta T3 e aumenta o tempo de bloqueio remoto de T1
- Com herança de prioridade isto não acontece
  - T3 herda a prioridade mais elevada de T1



Rômulo Silva de Oliveira, DAS-UFSC, março/2011 7

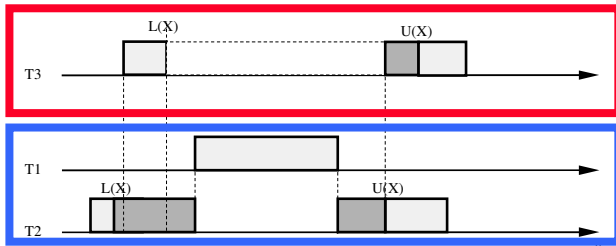
### MPCP – Multiprocessor Priority Ceiling Protocol

- Alocação estática de tarefas em processadores
- Acesso uniforme a memória global
- Prioridade fixa preemptiva
- Instruções do tipo Read-Modify-Write estão disponíveis para a implementação de mutexes (semáforos binários) residindo na memória compartilhada
  - “test-and-set”
  - “swap”
- Mutex global é acessado por tarefas em diferentes processadores
  - Protege uma seção crítica global (gcs)
- Mutex local é acessado apenas por tarefas alocadas a um mesmo processador
  - Protege uma seção crítica local (lcs)
- Uma seção crítica global
  - não pode aninhar nenhuma outra seção crítica
  - nem ser aninhada por qualquer outra

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 10

### Bloqueio Remoto 2/3

- T3 sofre bloqueio remoto de T2 por compartilhar mutex X
- T1 preempta T2 e aumenta o tempo de bloqueio remoto de T3
- Herança de prioridade não altera nada neste exemplo (PCP também não)
  - T1 tem mesmo prioridade mais elevada



Rômulo Silva de Oliveira, DAS-UFSC, março/2011 8

### MPCP – Multiprocessor Priority Ceiling Protocol

- A mais alta prioridade atribuída a qualquer tarefa no sistema inteiro é denotada por PH
- A mais alta prioridade entre as tarefas que acessam um dado semáforo  $S_i$  é denotada por  $PS_i$
- O priority ceiling de um semáforo local  $S$  é definido como a prioridade da tarefa de mais alta prioridade entre aquelas que acessam o semáforo  $S$ 
  - Como no caso do monoprocessador

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 11

### Bloqueio Remoto 3/3

- Um protocolo de sincronização para multiprocessadores busca limitar o bloqueio remoto
- Seções críticas são em geral curtas em relação ao tempo de execução total de uma tarefa
- Logo, a duração do bloqueio remoto deve estar associada com a duração das seções críticas das outras tarefas
- Execução fora de seção crítica não deveria somar no bloqueio remoto
  - De outra forma, os bloqueios remotos ficarão muito grandes

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 9

### MPCP – Multiprocessor Priority Ceiling Protocol

- O priority ceiling de um semáforo global  $SG_i$  deve ser definido de tal forma que
  - o priority ceiling de  $SG_i$  é estritamente mais alto que PH
  - Se  $SG_i$  e  $SG_j$  são semáforos globais e  $PS_{G_i} > PS_{G_j}$  então o priority ceiling de  $SG_i$  é mais alto que o priority ceiling de  $SG_j$
- Por exemplo,
  - Define-se um valor PG de base para os priority ceilings dos semáforos globais tal que  $PG > PH$
  - O priority ceiling  $PGS_i$  do semáforo global  $SG_i$  é definido como  $PG + PS_{G_i}$

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 12

## MPCP – Multiprocessor Priority Ceiling Protocol

- Recursos protegidos por semáforo local são acessados usando o PCP para monoprocessador em cada processador
- Recursos compartilhados globais são acessados usando instruções read-modify-write para obter semáforos globais e manipulação de filas
  - Busy-waiting limitado pela duração dessas operações
- Todas as seções críticas globais são executadas nos seus respectivos níveis de prioridade teto pré-definidos

Rômulo Silva de Oliveira, DAS-UFGO, março/2011 13

## Exemplo 3/3

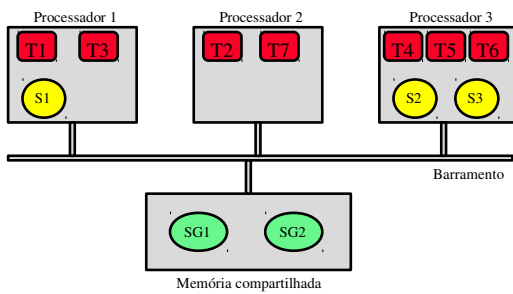
- Define o valor de base  $PG = P(T1) + 1$
- Semáforo S1 (local) priority ceiling  $P(T1)$
- Semáforo S2 (local) priority ceiling  $P(T4)$
- Semáforo S3 (local) priority ceiling  $P(T5)$
- Semáforo SG1 (global) priority ceiling  $PG + P(T1)$
- Semáforo SG2 (global) priority ceiling  $PG + P(T2)$

- $T1 = \{ \dots, P(S1), \dots, V(S1), \dots, P(SG1), \dots, V(SG1), \dots \}$
- $T2 = \{ \dots, P(SG1), \dots, V(SG1), \dots, P(SG2), \dots, V(SG2), \dots \}$
- $T3 = \{ \dots, P(SG1), \dots, V(SG1), \dots, P(S1), \dots, V(S1), \dots \}$
- $T4 = \{ \dots, P(S2), \dots, V(S2), \dots, P(SG1), \dots, V(SG1), \dots, P(S2), \dots, V(S2), \dots \}$
- $T5 = \{ \dots, P(S3), \dots, V(S3), \dots, P(SG2), \dots, V(SG2), \dots \}$
- $T6 = \{ \dots, P(S2), \dots, V(S2), \dots, P(S3), \dots, V(S3), \dots \}$
- $T7 = \{ \dots, P(SG2), \dots, V(SG2), \dots \}$

Rômulo Silva de Oliveira, DAS-UFGO, março/2011 16

## Exemplo 1/3

- Exemplo com 3 processadores, 7 tarefas
  - 2 semáforos globais, 3 semáforos locais



Rômulo Silva de Oliveira, DAS-UFGO, março/2011 14

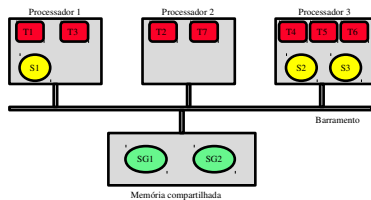
## MPCP – Descrição do Protocolo 1/4

- Cada tarefa usa sua prioridade original a não ser que ela esteja dentro de uma seção crítica

Rômulo Silva de Oliveira, DAS-UFGO, março/2011 17

## Exemplo 2/3

- $T1 = \{ \dots, P(S1), \dots, V(S1), \dots, P(SG1), \dots, V(SG1), \dots \}$
- $T2 = \{ \dots, P(SG1), \dots, V(SG1), \dots, P(SG2), \dots, V(SG2), \dots \}$
- $T3 = \{ \dots, P(SG1), \dots, V(SG1), \dots, P(S1), \dots, V(S1), \dots \}$
- $T4 = \{ \dots, P(S2), \dots, V(S2), \dots, P(SG1), \dots, V(SG1), \dots, P(S2), \dots, V(S2), \dots \}$
- $T5 = \{ \dots, P(S3), \dots, V(S3), \dots, P(SG2), \dots, V(SG2), \dots \}$
- $T6 = \{ \dots, P(S2), \dots, V(S2), \dots, P(S3), \dots, V(S3), \dots \}$
- $T7 = \{ \dots, P(SG2), \dots, V(SG2), \dots \}$



Rômulo Silva de Oliveira, DAS-UFGO, março/2011 15

## MPCP – Descrição do Protocolo 2/4

- PCP é usado para todos os acessos a semáforos locais
- Quando uma tarefa T solicita o semáforo local S no processador P
  - seja  $S^*$  o semáforo com o mais alto ceiling entre todos os semáforos locais bloqueados por tarefas que não T executando em P
  - Seja  $T^*$  a tarefa que detem  $S^*$
- Uma tarefa T no processador P pode obter o semáforo local S se sua prioridade é mais alta que o ceiling de  $S^*$
- Caso contrário, T é dita estar bloqueada por  $T^*$ 
  - A tarefa  $T^*$  herda a prioridade de T até que libere  $S^*$
- Uma tarefa T pode preemptar outra tarefa  $T'$  se sua prioridade for mais alta que a prioridade atual (nominal ou herdada) de  $T'$

Rômulo Silva de Oliveira, DAS-UFGO, março/2011 18

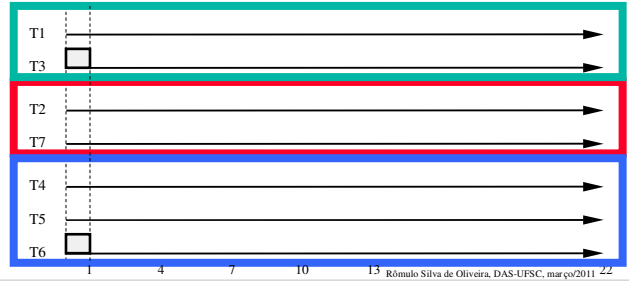
### MPCP – Descrição do Protocolo 3/4

- Quando uma tarefa T solicita um semáforo global SG, SG pode ser alocada para T através de uma transação atômica na memória compartilhada
  - Se SG não está alocada no momento para outra tarefa
- Uma tarefa T dentro de uma seção crítica global protegida por um semáforo global SG recebe a prioridade teto associada com SG imediatamente ao obter SG
- Uma tarefa T dentro de uma seção crítica global pode preemptar outra tarefa T' dentro de outra seção crítica global
  - Se a prioridade corrente de T for mais alta que a prioridade corrente de T'

Rômulo Silva de Oliveira, DAS-UFGO, março/2011 19

### Exemplo 2/20

- Em  $t=0$  as tarefas T3 e T6 são as únicas tarefas aptas e iniciam suas execuções nos seus respectivos processadores P1 e P3



Rômulo Silva de Oliveira, DAS-UFGO, março/2011 22

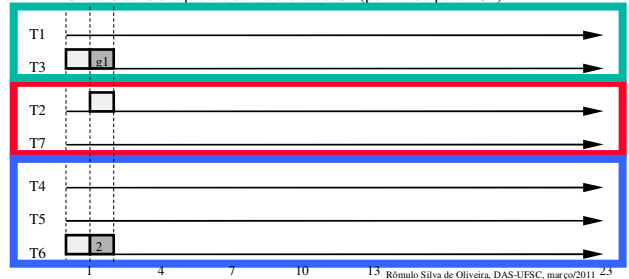
### MPCP – Descrição do Protocolo 4/4

- Se uma solicitação do semáforo global SG não pode ser atendida
  - A tarefa T solicitante é adicionada a uma fila ordenada associada com SG
  - A operação de inserção na fila é atômica (disable interrupt & spin-lock)
  - A prioridade usada para ordenação da fila é a prioridade original de T
- Quando uma tarefa T libera um semáforo global SG
  - A tarefa com mais alta prioridade T' esperando por SG é sinalizada
  - T' torna-se elegível para execução em seu processador na prioridade teto associada com SG
  - Se não existe tarefa suspensa esperando por SG, o semáforo SG fica livre

Rômulo Silva de Oliveira, DAS-UFGO, março/2011 20

### Exemplo 3/20

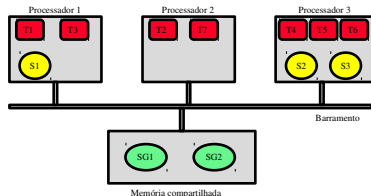
- Em  $t=1$  T3 bloqueia o semáforo global SG1 em P1
  - inicia a execução no nível de prioridade PG+P2
- A tarefa T2 inicia sua execução em P2
- Em P3 a tarefa T6 bloqueia o semáforo local S2 (permitido pelo PCP)



Rômulo Silva de Oliveira, DAS-UFGO, março/2011 23

### Exemplo 1/20

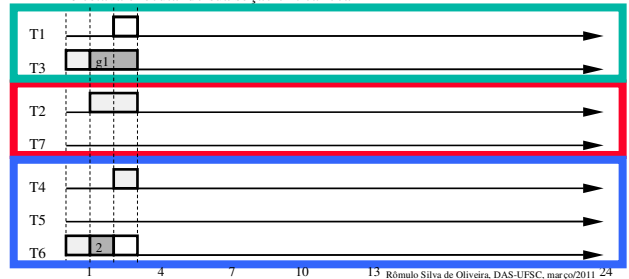
- $T1 = \{ \dots, P(S1), \dots, V(S1), \dots, P(SG1), \dots, V(SG1), \dots \}$
- $T2 = \{ \dots, P(SG1), \dots, V(SG1), \dots, P(SG2), \dots, V(SG2), \dots \}$
- $T3 = \{ \dots, P(SG1), \dots, V(SG1), \dots, P(S1), \dots, V(S1), \dots \}$
- $T4 = \{ \dots, P(S2), \dots, V(S2), \dots, P(SG1), \dots, V(SG1), \dots, P(S2), \dots, V(S2), \dots \}$
- $T5 = \{ \dots, P(S3), \dots, V(S3), \dots, P(SG2), \dots, V(SG2), \dots \}$
- $T6 = \{ \dots, P(S2), \dots, V(S2), \dots, P(S3), \dots, V(S3), \dots \}$
- $T7 = \{ \dots, P(SG2), \dots, V(SG2), \dots \}$



Rômulo Silva de Oliveira, DAS-UFGO, março/2011 21

### Exemplo 4/20

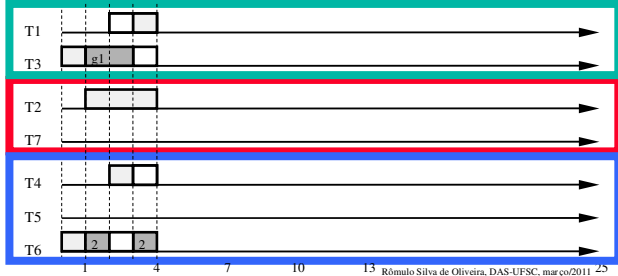
- Em  $t=2$  a tarefa T1 chega mas não é capaz de preemptar T3
- A tarefa T2 continua sua execução em P2
- Tarefa T4 chega em P3 e imediatamente preempta T6
  - T6 estava executando sua seção crítica local



Rômulo Silva de Oliveira, DAS-UFGO, março/2011 24

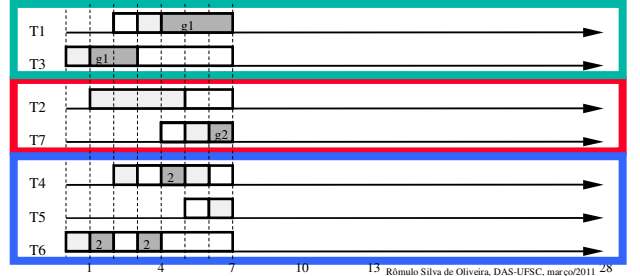
### Exemplo 5/20

- Em  $t=3$  a tarefa T3 libera o semáforo global SG1
  - Retorna para o seu nível de prioridade original, T1 preempta T3
- T4 tenta obter o semáforo local S2 mas ele está bloqueado por T6
  - T4 é bloqueada e T6 retoma sua execução na prioridade herdada de T4



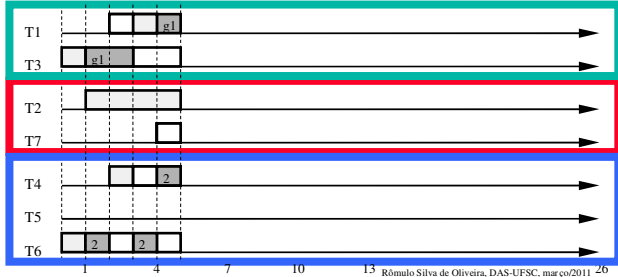
### Exemplo 8/20

- Em  $t=6$  a tarefa T2 ainda está bloqueada em P2
- A tarefa T7 obtém o semáforo global SG2, passa para prioridade PH+P1
- Em P3 a tarefa T4 tenta obter SG1 mas é bloqueada, T5 inicia execução
- T2 e T4 estão bloqueadas em SG1 em fila ordenada pela prioridade



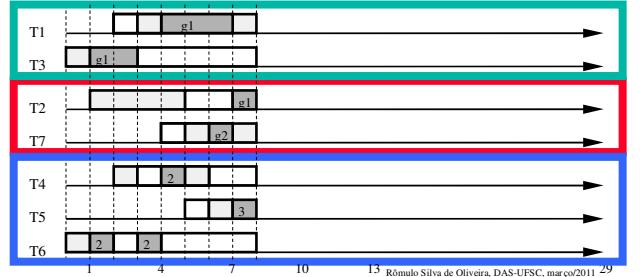
### Exemplo 6/20

- Em  $t=4$  a tarefa T1 aloca o semáforo global SG1
- A tarefa T7 chega mas não é capaz de preemptar T2
- A tarefa T6 libera S2 e retorna para sua prioridade original
  - A tarefa T4 preempta T6 e é capaz de obter S2



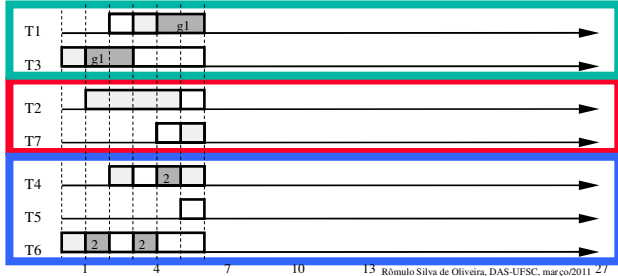
### Exemplo 9/20

- Em  $t=7$  a tarefa T1 libera SG1 e T2 obtém este semáforo por ser a tarefa pendente em SG1 com mais alta prioridade
  - A tarefa T1 continua sua execução em P1
- Em P3 a tarefa T5 obtém o semáforo local S3



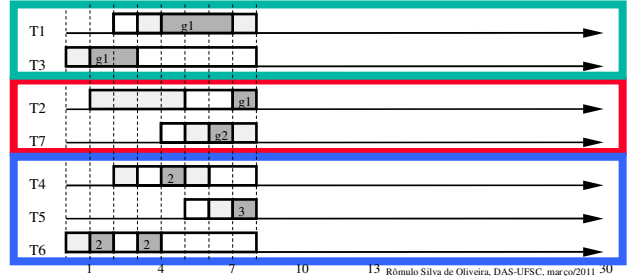
### Exemplo 7/20

- Em  $t=5$  a tarefa T1 continua a execução de sua seção crítica global
- A tarefa T2 tenta obter SG1 mas é bloqueada pois T1 detém este semáforo
  - T7 inicia sua execução
- Em P3 a tarefa T5 chega mas não é capaz de preemptar T4, que libera S2 e continua



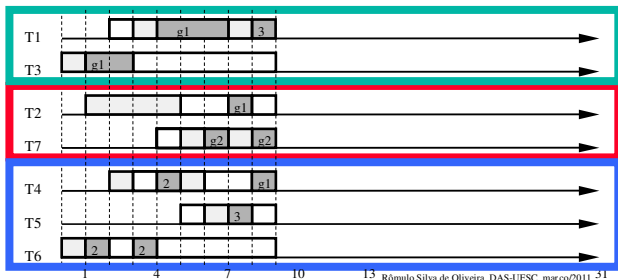
### Exemplo 10/20

- Ainda em  $t=7$ , a tarefa T2 executa sua seção crítica global com a prioridade PH+P1, a qual é mais alta que a prioridade PH+p5 na qual T7 executa sua seção crítica global
- Desta forma, T2 preempta T7 e inicia a execução de sua seção crítica global



### Exemplo 11/20

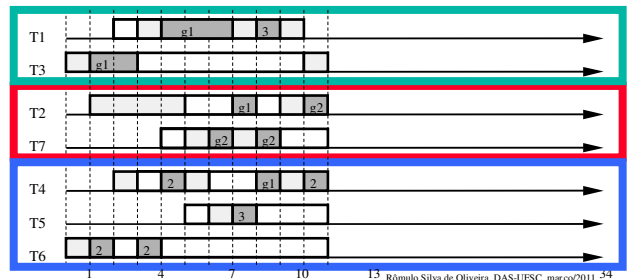
- Em  $t=8$  a tarefa T1 obtém o semáforo local S1 em P1
- Em P2 a tarefa T2 libera SG1 o qual é passado para T4
  - T2 retorna para prioridade normal e é preemptada por T7 prioridade PH+p5
- Em P3 a tarefa T4 entra em seção crítica global (prioridade mais alta) e preempta T5



13 Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 31

### Exemplo 14/20

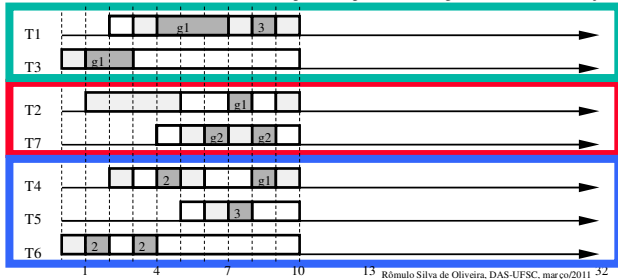
- Ainda em  $t=10$ , em P3 a tarefa T4 tenta obter S2
  - sua prioridade é mais alta que a prioridade teto do semáforo local S3 bloqueado
  - T4 obtém o semáforo S2 e continua sua execução



13 Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 34

### Exemplo 12/20

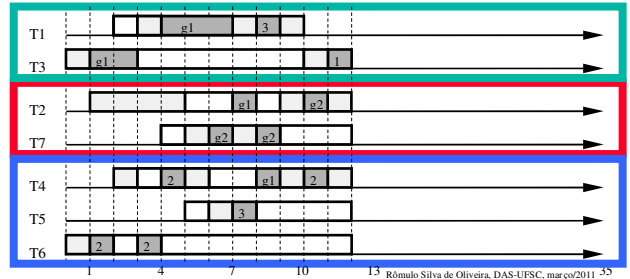
- Em  $t=9$ , T1 libera S1 e continua sua execução em P1
- Em P2 a tarefa T7 libera SG2 e retorna para sua prioridade original
  - Sendo então preemptada por T2
- Em P3 a tarefa T4 libera SG1 e retorna para sua prioridade original, continua execução



13 Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 32

### Exemplo 15/20

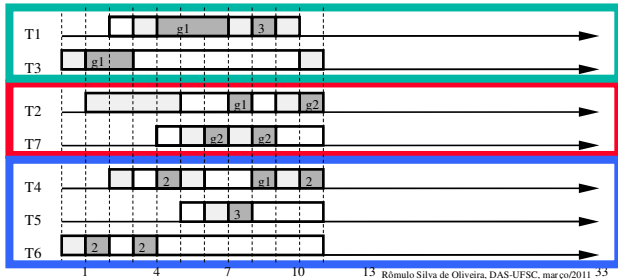
- Em  $t=11$  a tarefa T3 obtém o semáforo local S1
- Em P2 a tarefa T2 libera SG2 e retorna para sua prioridade original
- Em P3 a tarefa T4 libera S2 e continua sua execução



13 Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 35

### Exemplo 13/20

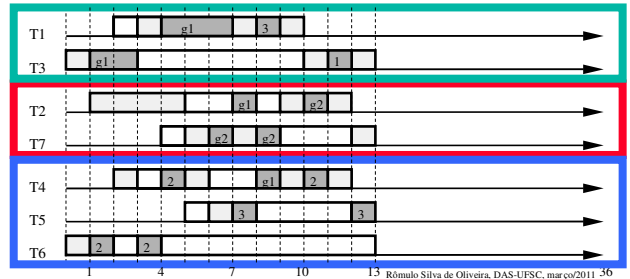
- Em  $t=10$  a tarefa T1 completa sua execução em P1
  - tarefa T3 retoma a execução
- Em P2 a tarefa T2 obtém o semáforo global SG2
  - assume a prioridade da sua seção crítica global



13 Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 33

### Exemplo 16/20

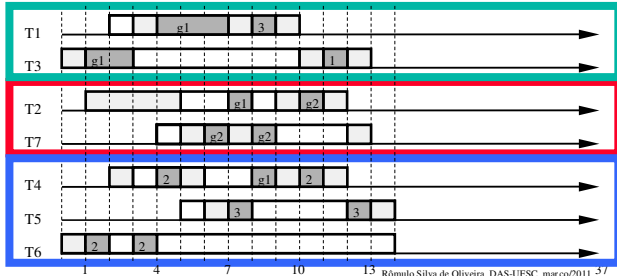
- Em  $t=12$ , T3 libera o semáforo local S1
- As tarefas T2 e T4 completam
- A tarefa T7 retoma sua execução em P2 enquanto T5 retoma a execução de sua seção crítica local em P3



13 Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 36

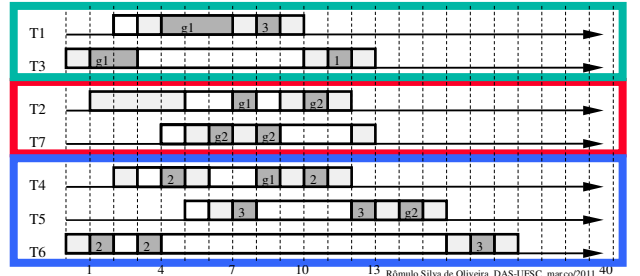
### Exemplo 17/20

- Em  $t=13$ , T3 conclui sua execução
- A tarefa T5 libera S3, continua a executar



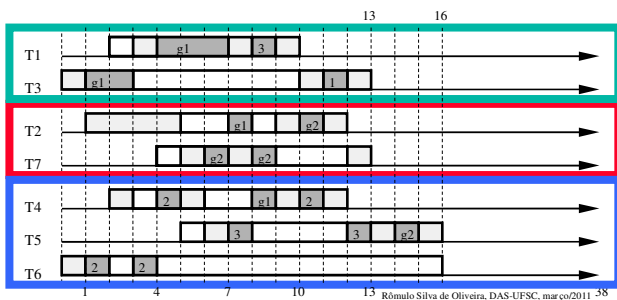
### Exemplo 20/20

- Diversos tipos de bloqueios são observados
  - Bloqueio direto local e remoto
  - Bloqueio por herança local e remoto
  - Em diferentes momentos da tarefa (executando, suspensa)



### Exemplo 18/20

- Em  $t=14$ , T5 obtém o semáforo SG2
- T5 libera SG2 em  $t=15$
- T5 conclui em  $t=16$

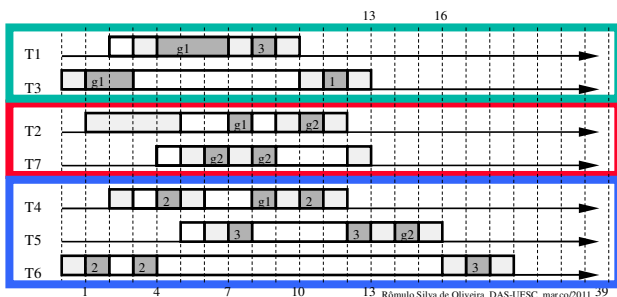


### Tempos Máximos de Bloqueio

- Objetivo fundamental é obter limites para os tempos de espera (bloqueios) sofridos por tarefas tentando usar recursos compartilhados
- Os limites de tempo de espera serão usados em testes para determinar a escalonabilidade das tarefas

### Exemplo 19/20

- Finalmente, em  $t=16$ , T6 pode executar, aloca e libera S3 e então completa sua execução em P3



### Tempos Máximos de Bloqueio – Deadlocks

- O MPCP impede deadlines, pois:
- O PCP impede deadlocks envolvendo recursos locais
- A premissa da não existência de aninhamento envolvendo recursos globais impede a participação dos mesmos em deadlocks
- Caso seções críticas globais aninhadas sejam permitidas, é necessário usar uma ordenação parcial explícita dos recursos globais para prevenir deadlocks

## MPCP – Análise de Escalonabilidade 1/12

- Notação

by  $\tau_i : ((C_{i,1}, C'_{i,1}, C_{i,2}, C'_{i,2}, \dots, C'_{i,s(i)-1}, C_{i,s(i)}, T_i)$   
 where,  
 $s(i)$  is no. of normal execution segments of  $\tau_i$ .  
 $s(i) - 1$  is no. of critical section execution segments of  $\tau_i$ .  
 $C_{i,j}$ : WCET of the  $j^{th}$  normal execution of  $\tau_i$ .  
 $C'_{i,k}$ : WCET of the  $k^{th}$  critical section of  $\tau_i$ .  
 $T_i$  denotes the period of task  $\tau_i$  (with implicit deadline).  
 $\tau_{i,j}$  denotes the  $j^{th}$  normal execution segment of task  $\tau_i$   
 $\tau'_{i,j}$  denotes the  $j^{th}$  critical section of task  $\tau_i$   
 Tasks are ordered in strictly decreasing order of priorities  
 i.e.  $i < j$  implies that task  $\tau_i$  has higher priority than task  $\tau_j$ .

$$C_i = \sum_{j=1}^{s(i)} C_{i,j} + \sum_{k=1}^{s(i)-1} C'_{i,k}$$

Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 43

## MPCP – Análise de Escalonabilidade 4/12

- Considerando as outras tarefas locais de prioridade mais alta
- Na perspectiva de  $\sigma_i$  o bloqueio remoto de uma dessas outras tarefas locais de alta prioridade tem o mesmo efeito que a auto-suspensão dela
  - Não tem mais um comportamento normal baseado em prioridades
- Efeito semelhante ao release jitter
  - Gera interferência adicional sobre  $\sigma_i$
- Esta interferência adicional é limitada pela máxima interferência que cada tarefa de prioridade mais alta  $\sigma_h$  pode gerar em um dado intervalo de tempo:
  - $S(h)-1$  é o número de seções críticas globais da tarefa  $\sigma_h$

$$(B_h^r = \sum_{q=1}^{s(h)-1} B_{h,q}^r) \quad \sum_{h < i \text{ \& } \tau_h \in P(\tau_i)} \lceil \frac{W_i^n + B_h^r}{T_h} \rceil C_h$$

Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 46

## MPCP – Análise de Escalonabilidade 2/12

- $P_i$  denota o  $i$ -ésimo processador
  - A função  $P(i)$  retorna o processador onde a tarefa  $\sigma_i$  foi alocada
- $R(i,j)$  representa o recurso associado com a  $j$ -ésima seção crítica de  $\sigma_i$
- Todas as seções críticas são globais
- Bloqueio remoto  $B_{i,j}$ 
  - Tempo máximo de bloqueio remoto devido a  $j$ -ésima seção crítica global de  $\sigma_i$

$$(B_h^r = \sum_{q=1}^{s(h)-1} B_{h,q}^r)$$

Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 44

## MPCP – Análise de Escalonabilidade 5/12

- Considerando as tarefas locais de mais baixa prioridade
- Inversões de prioridade múltiplas podem surgir
- As seções críticas globais de cada tarefa de mais baixa prioridade podem afetar a execução dos segmento normais da tarefa de mais alta prioridade
- A interferência por segmento é dada por:
  - $C'_{l,k}$  é o WCET da  $k$ -ésima seção crítica da tarefa  $\sigma_l$
  - $S(l)$  é o número de segmentos normais da tarefa  $\sigma_l$
  - A tarefa  $\sigma_l$  tem prioridade mais baixa que a tarefa  $\sigma_i$
  - As tarefas  $\sigma_l$  e  $\sigma_i$  ficam no mesmo processador

$$\sum_{l > i \text{ \& } \tau_l \in P(\tau_i)} \max_{1 \leq k \leq s(l)} C'_{l,k}$$

Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 47

## MPCP – Análise de Escalonabilidade 3/12

- Tempo de resposta da tarefa  $\sigma_i$  é composto por 4 componentes:
- Tempo de execução  $C_i$
- Tempo de bloqueio no acesso às seções críticas globais  $B_i$
- Interferência de tarefas locais com prioridade +alta
- Bloqueio por tarefas locais com prioridade +baixa mas que herdam prioridade +alta quando acessam seções críticas

Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 45

## MPCP – Análise de Escalonabilidade 6/12

- Tempo de resposta da tarefa  $\sigma_i$  ignorando as interferências que ela sofre em seu processador local:  $C^*i = C_i + B_i$
- Seria o tempo de resposta se
  - $T_i$  jamais fosse preemptada no processador local
  - Estivesse executando sozinha no processador
- O tempo máximo de resposta da tarefa  $\sigma_i$  é limitado pela convergência  $W_i$  de:
  - $W_i = C^*i$

$$W_i^{n+1} = C_i^* + \sum_{h < i \text{ \& } \tau_h \in P(\tau_i)} \lceil \frac{W_i^n + B_h^r}{T_h} \rceil C_h$$

$$+ s(i) \sum_{l > i \text{ \& } \tau_l \in P(\tau_i)} \max_{1 \leq k \leq s(l)} C'_{l,k}$$

Rômulo Silva de Oliveira, DAS-UFSBC, março/2011 48



### MPCP – Análise de Escalonabilidade 7/12

- Falta calcular  $B_i$
- $B_i = \text{SOMA}_{j \in \sigma(i)} B_{i,j}$
- $B_{i,j}$  representa o bloqueio que a tarefa  $\sigma_i$  sofre ao solicitar a seção crítica global  $C_{i,j}$ 
  - Devido a tarefas de mais baixa prioridade
  - Devido a tarefas de mais alta prioridade
- Uma seção crítica não é preemptável, mas a fila do recurso é ordenada pela prioridade

Rômulo Silva de Oliveira, DAS-UFG, março/2011 49

### MPCP – Análise de Escalonabilidade 10/12

- $B_{i,j}$  representa o bloqueio remoto que a tarefa  $\sigma_i$  sofre na solicitação a seção crítica global  $C_{i,j}$
- $B_{i,j}$  é limitado pela convergência abaixo

$$B_{i,j}^{r,n+1} = \max_{l > i \text{ \& } (\tau_{l,u}^r) \in R(\tau_i, j)} (W'_{l,u}) + \sum_{h < i \text{ \& } (\tau_{h,v}^r) \in R(\tau_i, j)} \left( \left\lceil \frac{B_{i,j}^{r,n}}{T_h} \right\rceil + 1 \right) (W'_{h,v})$$

$$B_{i,j}^{r,0} = \max_{l > i \text{ \& } (\tau_{l,u}^r) \in R(\tau_i, j)} (W'_{l,u})$$

Rômulo Silva de Oliveira, DAS-UFG, março/2011 52

### MPCP – Análise de Escalonabilidade 8/12

- Quando  $i$  solicita uma seção crítica global, uma tarefa de mais baixa prioridade pode ter recém bloqueado o recurso
- $\sigma_i$  precisa esperar que esta tarefa libere o recurso
- Isto pode acontecer somente uma vez
  - A fila do recurso é ordenada por prioridade
- $W'_{l,u}$  é o tempo que  $\sigma_l$  leva para liberar este recurso

$$\max_{l > i \text{ \& } (\tau_{l,u}^r) \in R(\tau_i, j)} (W'_{l,u})$$

Rômulo Silva de Oliveira, DAS-UFG, março/2011 50

### MPCP – Análise de Escalonabilidade 11/12

- Quanto tempo uma tarefa  $i$  demora para liberar a seção crítica global  $C_{i,k}$  ?
  - A prioridade teto global da seção crítica  $C_{i,k}$  é  $gc(i,k)$
- O máximo tempo de resposta associado com o segmento crítico  $C_{i,k}$  é limitado por:
  - Tarefas locais tem prioridade mais baixa que qualquer seção crítica global
  - Mas tarefas locais podem herdar a prioridade teto de uma outra seção crítica com prioridade teto mais alta que  $C_{i,k}$  e interferir com a execução de  $C_{i,k}$
  - Isto pode acontecer apenas uma vez para cada tarefa local

$$W'_{i,k} = C'_{i,k} + \sum_{\tau_u \in P(\tau_i)} \max_{1 \leq v \leq s(u) \text{ \& } gc(u,v) > gc(i,k)} C'_{u,v}$$

Rômulo Silva de Oliveira, DAS-UFG, março/2011 53

### MPCP – Análise de Escalonabilidade 9/12

- Tarefas com prioridade mais alta que  $\sigma_i$  podem executar várias vezes a seção crítica global enquanto  $\sigma_i$  espera
- Todas as tarefas com prioridade mais alta que  $\sigma_i$  que acessem este recurso podem fazer  $\sigma_i$  esperar
- Cada tarefa pode atrapalhar várias vezes
  - Depende do período da tarefa de mais alta prioridade
- $W_h$  é o tempo que  $\sigma_h$  leva para liberar este recurso

$$\sum_{h < i \text{ \& } (\tau_{h,v}^r) \in R(\tau_i, j)} \left( \left\lceil \frac{B_{i,j}^{r,n}}{T_h} \right\rceil + 1 \right) (W'_{h,v})$$

Rômulo Silva de Oliveira, DAS-UFG, março/2011 51

### MPCP – Análise de Escalonabilidade 12/12

- Equações

$$W_i^{n+1} = C_i^* + \sum_{h < i \text{ \& } \tau_h \in P(\tau_i)} \left\lceil \frac{W_i^n + B_h^r}{T_h} \right\rceil C_h$$

$$+ s(i) \sum_{l > i \text{ \& } \tau_l \in P(\tau_i)} \max_{1 \leq k < s(l)} C'_{l,k}$$

$$B_{i,j}^{r,n+1} = \max_{l > i \text{ \& } (\tau_{l,u}^r) \in R(\tau_i, j)} (W'_{l,u}) + \sum_{h < i \text{ \& } (\tau_{h,v}^r) \in R(\tau_i, j)} \left( \left\lceil \frac{B_{i,j}^{r,n}}{T_h} \right\rceil + 1 \right) (W'_{h,v})$$

$$W'_{i,k} = C'_{i,k} + \sum_{\tau_u \in P(\tau_i)} \max_{1 \leq v \leq s(u) \text{ \& } gc(u,v) > gc(i,k)} C'_{u,v}$$

Rômulo Silva de Oliveira, DAS-UFG, março/2011 54

### MPCP:Spin 1/4

- O principal benefício do uso de suspensão no MPCP é que o tempo livre durante a suspensão é disponibilizado para a execução de outras tarefas
- As desvantagens são
  - Jitter das tarefas de mais alta prioridade aumentando sua interferência sobre tarefas locais de mais baixa prioridade
  - Múltiplas inversões de prioridade por tarefa
- Uma abordagem alternativa, que elimina as desvantagens, é:
  - Não liberar o processador
  - Ficar em busy-waiting até a seção crítica ser obtida
- Chamado MPCP:Spin

Rômulo Silva de Oliveira, DAS-UFG, março/2011 55

### MPCP:Spin 4/4

- O uso de spin-lock reduz as inversões de prioridade devido à herança remota de prioridade
  - Com suspensão pode ser 1 vez inicial mais 1 vez sempre que a tarefa de mais alta prioridade fica suspensa e permite a tarefa de baixa prioridade executar
  - Com spin-lock pode ser apenas 1 vez inicial
- Entretanto, spinning resulta em interferência adicional das tarefas de mais alta prioridade sobre as tarefas locais de mais baixa prioridade
  - Com suspensão era Ch por vez
  - Com spin-lock é C<sup>n</sup>h por vez

Rômulo Silva de Oliveira, DAS-UFG, março/2011 58

### MPCP:Spin 2/4

- Tarefas executam um laço enquanto esperam a liberação da seção crítica global
- Isto evita qualquer interferência futura de tarefas locais
  - com prioridade mais baixa
  - que venham a obter seções críticas globais
  - e tenham então suas prioridades elevadas
- Na prática, pode ser implementado como “virtual spinning”
  - Outras tarefas podem executar
  - Mas não podem solicitar recursos globais
  - Neste caso elas seriam suspensas
- O número de inversões de prioridade devido a tarefas de mais baixa prioridades é um
- O efeito de auto-suspensão também é eliminado
- O tempo gasto na espera pela seção crítica torna-se parte do tempo de execução da tarefa
- A tarefa nunca suspende durante sua execução
  - Isto melhora o desempenho médio
  - Mas não garante melhora do pior caso

Rômulo Silva de Oliveira, DAS-UFG, março/2011 56

### Tempos Máximos de Bloqueio – Aninhamento

- O protocolo não muda quando seções críticas globais são aninhadas entre si
- Deadlocks devem ser evitados explicitamente via alocação ordenada dos recursos
- Análise apresentada somente é válida quando não existe aninhamento
- Seções críticas globais aninhadas
  - Aumentam n<sup>i</sup>
  - Aumentam bloqueios remotos via transitividade
- Um job J detendo um semáforo global em um processador
  - Pode bloquear em um semáforo aninhado bloqueado por tarefa em outro processador
  - A qual o qual pode ficar bloqueada em outro semáforo aninhado
  - E assim por diante
- Bloqueios transitivos surgem por todo o sistema

Rômulo Silva de Oliveira, DAS-UFG, março/2011 59

### MPCP:Spin 3/4

- Um limite máximo para o tempo de computação da tarefa T<sub>i</sub> é dado por:  
 $C^*i = C_i + B_i$
- O tempo de resposta no pior caso da tarefa T<sub>i</sub> é limitado pela convergência de W<sub>i</sub>
  - W<sub>i</sub> = C<sup>\*</sup>i
  - Os termos de bloqueio B<sub>i,j</sub> são os mesmos de antes

$$W_i^{n+1} = C_i^* + \sum_{h < i \text{ e } \tau_h \in P(\tau_i)} \lceil \frac{W_i^n}{T_h} \rceil C_h^* \\ + \sum_{l > i \text{ e } \tau_l \in P(\tau_i)} \max_{1 \leq k < a(l)} C_{l,k}^*$$

Rômulo Silva de Oliveira, DAS-UFG, março/2011 57

### Multiprocessor Priority Ceiling Protocol – Resumo

- Baseado no Priority Ceiling Protocol
- Definido para sistemas particionados usando prioridade fixa
- A prioridade teto de recursos globais são definidas como mais elevadas do que qualquer tarefa no sistema
- Quando uma tarefa tenta acessar um recurso global bloqueado
  - ela é suspensa
  - espera em uma fila associada com aquele recurso, ordenada pelas prioridades
- Isto permite tarefas locais de baixa prioridade continuarem sua execução
- Quando o recurso em questão é liberado
  - a primeira tarefa da fila deste recurso é retomada e
  - executa na prioridade teto do recurso

Rômulo Silva de Oliveira, DAS-UFG, março/2011 60

## Multiprocessor Priority Ceiling Protocol – Resumo

- MPCP permite que tarefas de baixa prioridade executem enquanto uma tarefa de mais alta prioridade no mesmo processador está bloqueada esperando por um recurso global
- Isto pode gerar mais inversão de prioridade
  - A tarefa de mais baixa prioridade pode tentar acessar um outro recurso global bloqueado com um teto mais alto
  - e então executar antes da tarefa de mais alta prioridade
  - mesmo quando aquele primeiro recurso for liberado
- MPCP não permite aninhamento de seções críticas globais
- MPCP também não permite aninhamento de seções críticas locais e globais
- Com estas restrições MPCP provê um tempo de bloqueio limitado

Rômulo Silva de Oliveira, DAS-UFG, março/2011 61

## MSRP – Multiprocessor Stack Resource Policy

- Não é permitido aninhar seções críticas globais
- Quando a tarefa  $T_i$  executando no processador  $M_k$  tenta acessar um recurso global  $R_j$ 
  - a tarefa torna-se não preemptável
- Se o recurso  $M_k$  está livre
  - O recurso fica bloqueado e  $T_i$  executa sua seção crítica global
- Se o recurso  $M_k$  está bloqueado por outra tarefa (de outro processador)
  - A tarefa  $T_i$  é inserida em uma fila FIFO associada com  $R_j$
  - A tarefa  $T_i$  fica em busy-waiting (spin-lock)
- Quando uma tarefa solicita uma seção crítica global sua prioridade é elevada para a máxima prioridade no seu processador (torna-se não preemptável)

Rômulo Silva de Oliveira, DAS-UFG, março/2011 64

## Exclusão Mútua em Sistemas Particionados

- MPCP – Multiprocessor Priority Ceiling Protocol
  - Prioridade fixa
  - Existe uma versão para EDF
- MSRP – Multiprocessor Stack Resource Policy
  - Prioridade fixa ou EDF

Rômulo Silva de Oliveira, DAS-UFG, março/2011 62

## MSRP – Multiprocessor Stack Resource Policy

- Quando a tarefa  $T_i$  executando no processador  $M_k$  libera um recurso global  $R_j$ 
  - Necessário verificar a fila FIFO associada com  $R_j$
- Se não existe tarefa esperando por  $R_j$ 
  - O recurso fica livre
- Caso existam tarefas esperando por  $R_j$ 
  - A primeira da fila recebe o recurso, ela estava no spin-lock não preemptável
- A tarefa  $T_i$  torna-se preemptável novamente
- A prioridade atribuída à tarefa  $T_i$  enquanto ela acessa um recurso compartilhado
  - não depende do estado das tarefas em outros processadores
  - nem de suas prioridades

Rômulo Silva de Oliveira, DAS-UFG, março/2011 65

## MSRP – Multiprocessor Stack Resource Policy

- Definido para multiprocessador particionado
  - Baseado no Stack Resource Policy
- Recursos podem ser locais ou globais
  - Locais: usados apenas por tarefas alocadas ao mesmo processador
  - Globais: usados por tarefas alocadas a diferentes processadores
- Para recursos locais é idêntico ao SRP
  - Cada tarefa possui um nível de preempção
  - Cada recurso local possui uma prioridade teto
  - Cada processador possui um teto do sistema
- É permitido aninhar seções críticas locais

Rômulo Silva de Oliveira, DAS-UFG, março/2011 63

## MSRP – Tempos de Bloqueio 1/4

- A duração do spin-lock que uma tarefa  $T_i$  alocada ao processador  $M_k$  precisa executar ao acessar um recurso global  $R_j$  é limitada
- No pior caso a tarefa  $T_i$  encontrará
  - O recurso global  $R_j$  recém ocupado
  - A fila FIFO deste recurso formada por uma tarefa de cada outro processador, com exceção do seu próprio e daquele onde está a tarefa que detém o recurso
  - A tarefa que detém o recurso ocupará  $R_j$  pela maior duração possível
  - Cada uma das tarefas da fila FIFO ocupará  $R_j$  pela maior duração possível
- $Spin(R_j, M_k) = \sum_{m \in \{M\}} MAX_{T_i \in T_m} [W_{i_m}]$

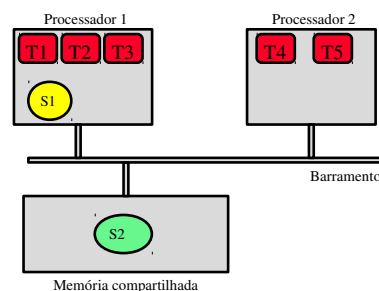
Rômulo Silva de Oliveira, DAS-UFG, março/2011 66

### MSRP – Tempos de Bloqueio 2/4

- O tempo de spin-lock aumenta a duração de cada seção crítica global
- Portanto, aumenta o WCET de uma tarefa  $T_i$  que acesse recursos globais
- O tempo de spin-lock também aumenta o tempo de bloqueio sofrido por tarefas alocadas no mesmo processador que  $T_i$ , que seriam capazes de preemptar  $T_i$
- O verdadeiro WCET  $C_i$  da tarefa  $T_i$ , denotado por  $C^*_i$ , é definido como o  $C_i$  original mais o tempo total gasto por  $T_i$  em spin-lock
- $C^*_i = C_i + \sum_{\text{seção crítica global de } T_i} \text{Spin}(R_j, M_k)$

### MSRP – Exemplo 1/19

- Exemplo com 2 processadores, 5 tarefas
  - 1 semáforo global, 1 semáforo local

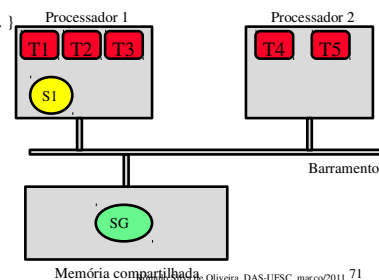


### MSRP – Tempos de Bloqueio 3/4

- MSRP mantém as mesmas propriedades básicas do SRP
- Uma vez que a tarefa inicia sua execução ela não pode ser bloqueada, mas apenas preemptada por tarefas de mais alta prioridade
- Uma tarefa pode sofrer bloqueio de tarefa com prioridade mais baixa no máximo igual a duração de uma única seção crítica (adicionado do tempo de spin-lock se o recurso for global)
- A execução de todas as tarefas alocadas em um processador é perfeitamente aninhada
  - Uma vez que a tarefa iniciou ela não pode ser bloqueada
- Todas as tarefas podem compartilhar a mesma pilha

### MSRP – Exemplo 2/19

- $T1 = \{ \dots \}$
- $T2 = \{ \dots, P(S1), \dots, V(S1), \dots \}$
- $T3 = \{ \dots, P(S1), \dots, P(SG), \dots, V(SG), \dots, V(S1), \dots \}$
- $T4 = \{ \dots \}$
- $T5 = \{ \dots, P(SG), \dots, V(SG), \dots \}$



### MSRP – Tempos de Bloqueio 4/4

- O tempo de bloqueio sofrido por uma tarefa pode ser devido a uma seção crítica local ou global:
 
$$B_i = \text{MAX}[ B_i^{\text{local}}, B_i^{\text{global}} ]$$

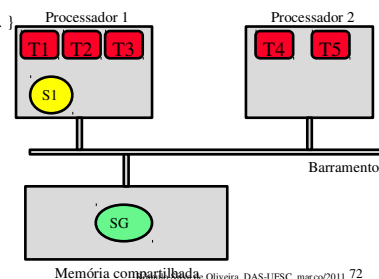
$$B_i^{\text{local}} = \max_{j,h,k} \{ \omega_{jh}^k \mid (\tau_j \in T_{P_i}) \wedge (\rho^k \text{ is local to } P_i) \wedge (\lambda_i > \lambda_j) \wedge (\lambda_i \leq \text{ceil}(\rho^k)) \}$$

$$B_i^{\text{global}} = \max_{j,h,k} \{ \omega_{jh}^k + \text{spin}(\rho^k, P_i) \mid (\tau_j \in T_{P_i}) \wedge (\rho^k \text{ is global}) \wedge (\lambda_i > \lambda_j) \}$$

### MSRP – Exemplo 3/19

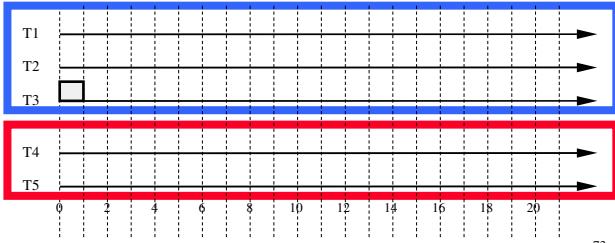
- $T1 = \{ \dots \}$
- $T2 = \{ \dots, P(S1), \dots, V(S1), \dots \}$
- $T3 = \{ \dots, P(S1), \dots, P(SG), \dots, V(SG), \dots, V(S1), \dots \}$
- $T4 = \{ \dots \}$
- $T5 = \{ \dots, P(SG), \dots, V(SG), \dots \}$

- Teto de S1 é 2



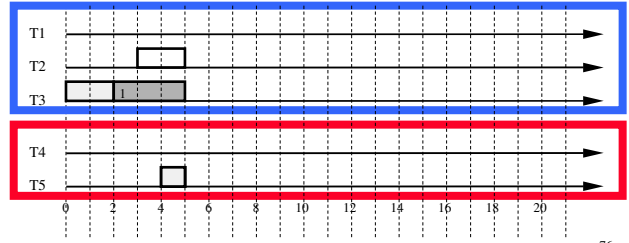
### MSRP – Exemplo 4/19

- Em  $t=0$  a tarefa T3 inicia sua execução



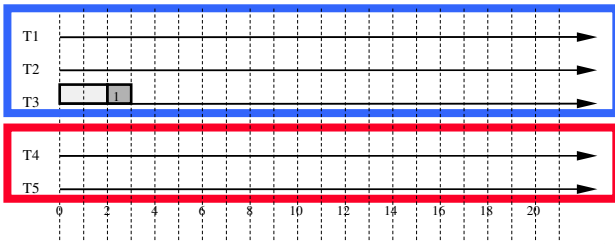
### MSRP – Exemplo 7/19

- Em  $t=4$  a tarefa T5 inicia sua execução



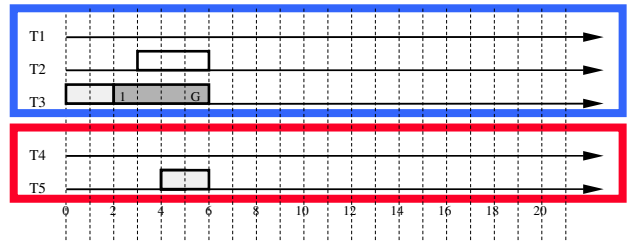
### MSRP – Exemplo 5/19

- Em  $t=2$ , a tarefa T3 aloca a seção crítica local S1



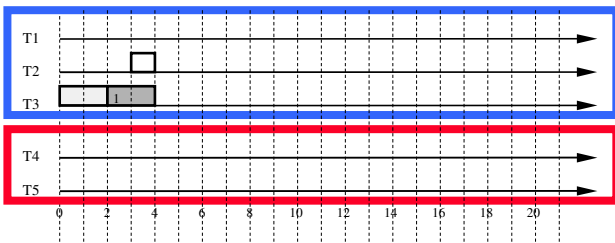
### MSRP – Exemplo 8/19

- Em  $t=5$  a tarefa T3 bloqueia o recurso global SG e torna-se não preemptável em P1



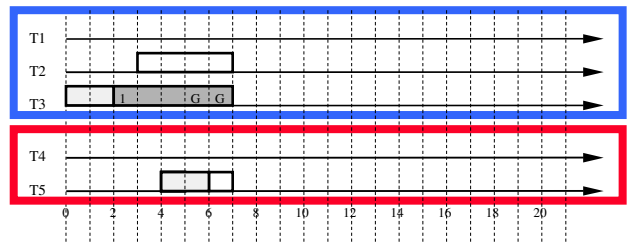
### MSRP – Exemplo 6/19

- Em  $t=3$  a tarefa T2 chega mas é bloqueada, pois seu nível de preempção é igual ao nível corrente no processador 1



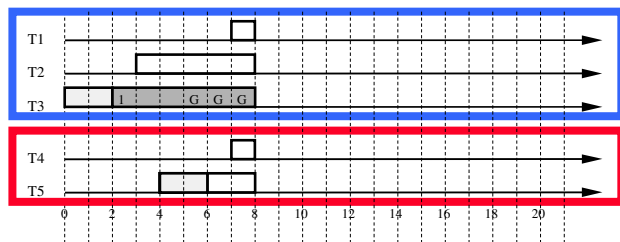
### MSRP – Exemplo 9/19

- Em  $t=6$ , a tarefa T5 tenta acessar o recurso global SG o qual está bloqueado por T2
  - Ela entra em busy-waiting não preemptável



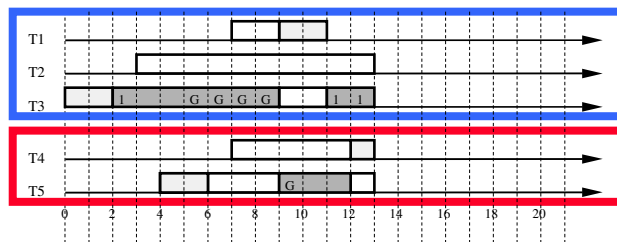
### MSRP – Exemplo 10/19

- Em  $t=7$ , as tarefas T1 e T4 estão bloqueadas, pois os teto de sistema nos dois processadores estão no máximo



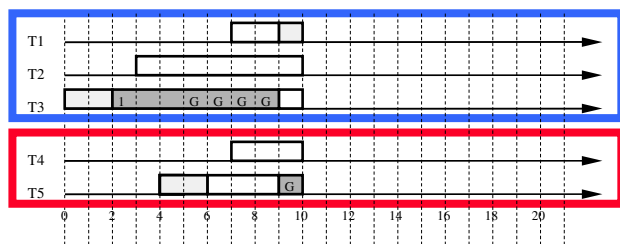
### MSRP – Exemplo 13/19

- Em  $t=12$ , a tarefa T5 libera o recurso global
- Ela é preemptada pela tarefa T4



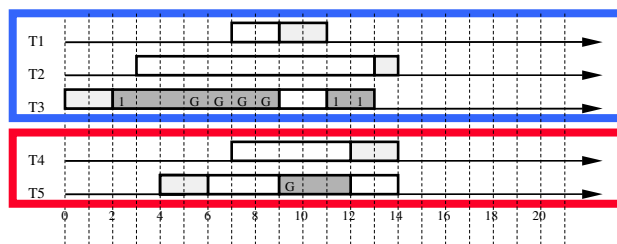
### MSRP – Exemplo 11/19

- Em  $t=9$ , a tarefa T3 libera o recurso global SG e a tarefa T5 inicia sua seção crítica global relativa a SG
  - O teto de sistema do processador 1 volta para 2 e a tarefa T1 consegue preemptar



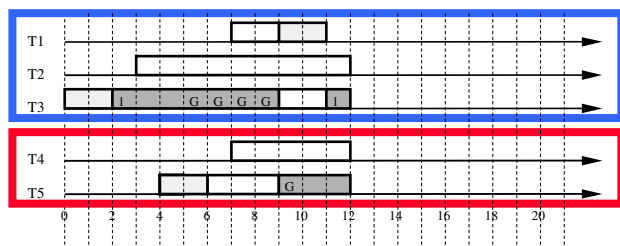
### MSRP – Exemplo 14/19

- Em  $t=13$ , a tarefa T3 libera o recurso local
  - Ela é imediatamente preemptada por T2



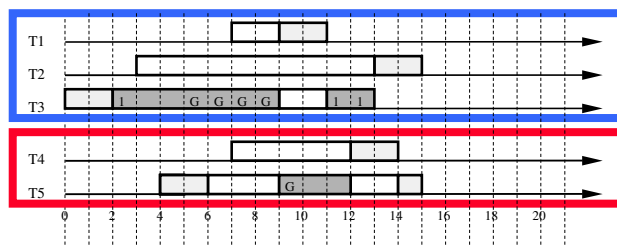
### MSRP – Exemplo 12/19

- Em  $t=11$ , a tarefa T1 conclui sua execução
- A tarefa T3 retoma o processador
  - pois ela ainda mantém o recurso local alocado



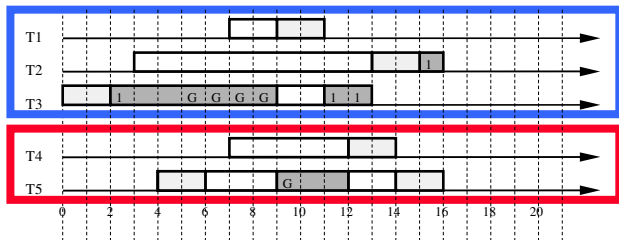
### MSRP – Exemplo 15/19

- Em  $t=14$ , a tarefa T4 conclui
- A tarefa T5 volta a executar



### MSRP – Exemplo 16/19

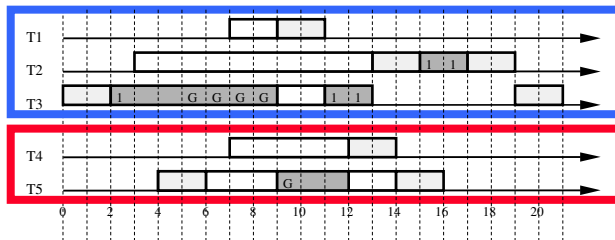
- Em  $t=15$ , a tarefa T2 aloca o recurso local
  - Seu nível de preempção permite isto



Rômulo Silva de Oliveira, DAS-UFSC, março/2011 85

### MSRP – Exemplo 19/19

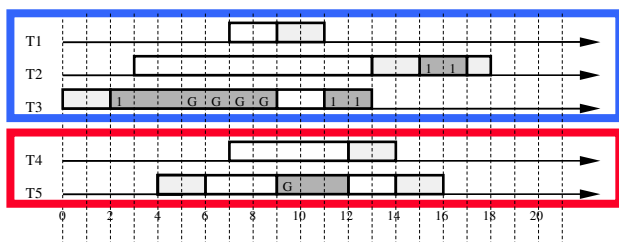
- Implementação simples
- Prioridades são elevadas para valores estáticos que independem da dinâmica do sistema



Rômulo Silva de Oliveira, DAS-UFSC, março/2011 88

### MSRP – Exemplo 17/19

- Em  $t=17$ , a tarefa T2 libera o recurso local



Rômulo Silva de Oliveira, DAS-UFSC, março/2011 86

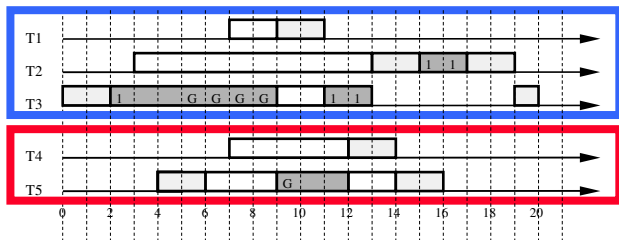
### Multiprocessor Stack Resource Policy – Resumo

- É baseado no Stack Resource Policy para uniprocessadores
- Definido para sistemas particionados usando prioridade fixa ou EDF
- Quando uma tarefa é bloqueada ao tentar acessar um recurso global, ela entra em busy-waiting e não é preemptada (spin-lock)
- Uma fila FIFO é usada para definir a ordem de acesso entre tarefas esperando para usar um recurso global
- MSRP provê
  - um tempo de bloqueio limitado
  - um aumento limitado no tempo de execução das tarefas devido aos spin-locks
- Como usando MSRP a execução das tarefas em cada processador é perfeitamente aninhada, todas as tarefas podem compartilhar uma única pilha
- É mais simples implementar MSRP do que MPCP

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 89

### MSRP – Exemplo 18/19

- Em  $t=19$ , a tarefa T2 conclui
  - T3 volta a executar



Rômulo Silva de Oliveira, DAS-UFSC, março/2011 87

### Resumo

- MPCP – Multiprocessor Priority Ceiling Protocol
  - Prioridade fixa
  - Existe uma versão para EDF
- MSRP – Multiprocessor Stack Resource Policy
  - Prioridade fixa ou EDF

Rômulo Silva de Oliveira, DAS-UFSC, março/2011 90