
Seções Críticas em Sistemas de Tempo Real



Fundamentos dos Sistemas de Tempo Real

Rômulo Silva de Oliveira

eBook Kindle, 2018

www.romulosilvadeoliveira.eng.br/livrotemporeal

Outubro/2018

- Processador é o recurso mais importante
 - Mais essencial
- Outros recursos também são usados
 - Canais de comunicação
 - Estruturas de dados
 - Arquivos
- Necessidade de alocar e liberar tais recursos
 - Exigem exclusão mútua no acesso

- Problema da Seção crítica
 - O compartilhamento de recursos entre tarefas.
- Solução tradicional
 - Mutex
- Problema que o Mutex pode ocasionar em sistemas de tempo real
 - Gerar situações onde ocorre uma demorada **inversão de prioridades** entre as tarefas.
- Solução do problema da seção crítica para sistemas de tempo real
 - Mutex especiais

- Existem várias questões a serem tratadas:
- Quando e sob quais condições os pedidos de recursos são atendidos ?
- Como são escalonados os jobs que estão esperando por um dado recurso ?
- Quanto tempo uma dada tarefa precisa esperar até conseguir acessar um dado recurso ?
- Como a existência de recursos outros além do processador afeta a escalonabilidade do sistema ?

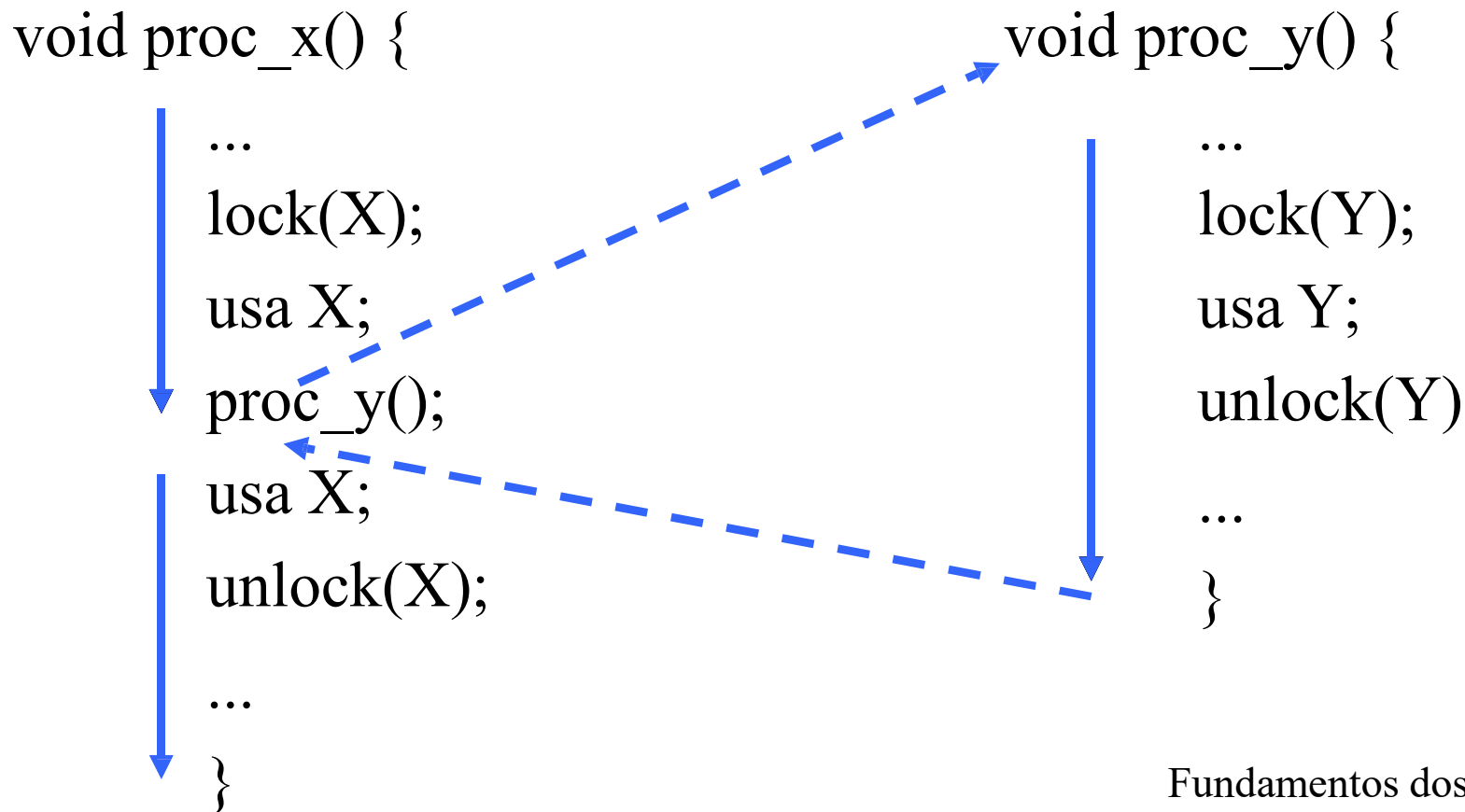
- Sistema monoprocessador
- Podem existir diferentes recursos compartilhados
 - Estruturas de dados independentes
 - Periféricos
 - Etc
- Recursos não podem ser preemptados e requerem exclusão mútua
- Existe apenas uma unidade de cada recurso
 - Mais adiante esta suposição será revista
- Não estamos preocupados com recursos que
 - Podem ser acessados simultaneamente (exemplo: tabela read-only)
 - Existe grande número de unidades (exemplo: bytes na memória)

- Mecanismo de sincronização é usado pelas tarefas para controlar o acesso aos recursos compartilhados
- Um simples MUTEX será usado para controlar o acesso
- LOCK(X) é usado pela tarefa para indicar que ela requer o uso do recurso X
 - Tarefa fica bloqueada até o acesso ao recurso ser autorizado
 - Enquanto bloqueada, ela sai da fila de aptos e não disputa o processador
- UNLOCK(X) é usado pela tarefa para indicar que ela não mais requer acesso ao recurso X
 - Retorno imediato
 - Poderá liberar outra tarefa que estava esperando para usar o recurso X

- Seção Crítica
 - Trecho de código onde a tarefa acessa algum recurso compartilhado X
 - Limitado pelas operações LOCK(X) e UNLOCK(X)
- Seções críticas podem ser aninhadas
- É suposto que apenas aninhamentos perfeitos são usados

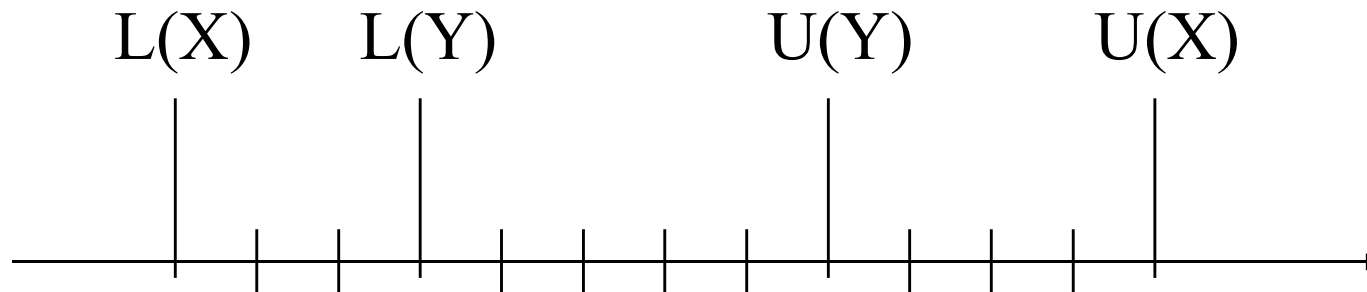
- Situação mais comum:

- Chama rotina `proc_x` que tem `LOCK(X)` no início e `UNLOCK(X)` no fim
- Enquanto da rotina `proc_x` tem uma chamada para rotina `proc_y`
- Rotina `proc_y` tem `LOCK(Y)` no início e `UNLOCK(Y)` no fim



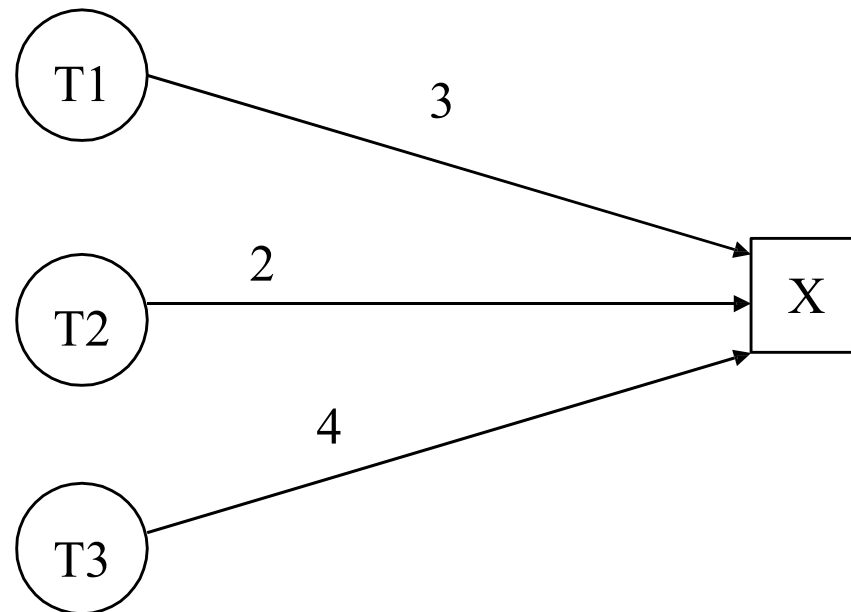
Modelo de Recursos

- Será usada uma notação baseada no livro da Jane Liu (não igual)
- $[X,3]$
 - Tarefa precisa do recurso X por 3 unidades de tempo
- Recursos aninhados são representados por colchetes aninhados
- $[X,3[Y,5]4]$
 - Aloca X , usa por 3 unidades de tempo, aloca Y , usa ambos por 5 unidades de tempo, libera Y , usa X por mais 4 unidades de tempo, libera X



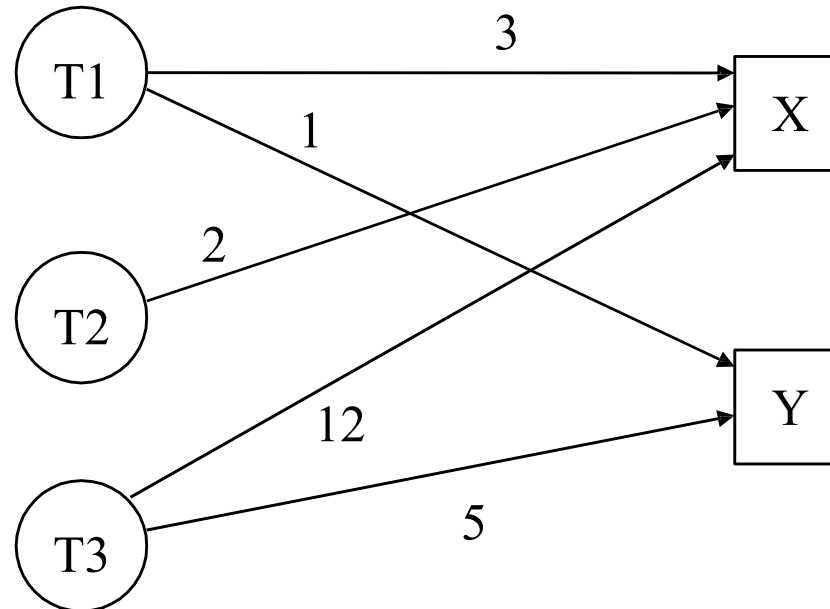
Especificação das Demandas de Recursos

- Grafo dirigido acíclico
- Arco da tarefa para o recurso indica que a tarefa requer o recurso
- Anotação no arco indica a duração da seção crítica
- Exemplo
 - T1:[X,3]
 - T2:[X,2]
 - T3:[X,4]



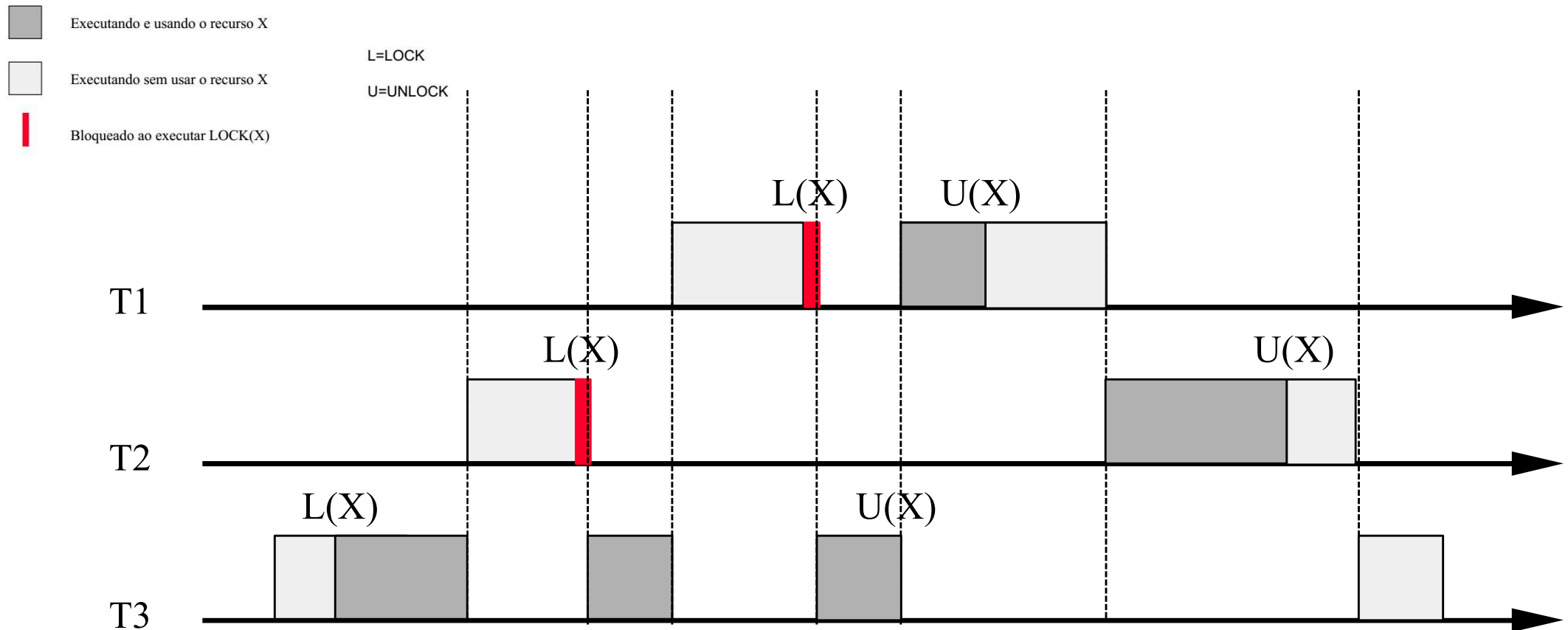
Especificação das Demandas de Recursos

- Exemplo
 - T1:[X,3][Y,1]
 - T2:[X,2]
 - T3:[X,3][Y,5]



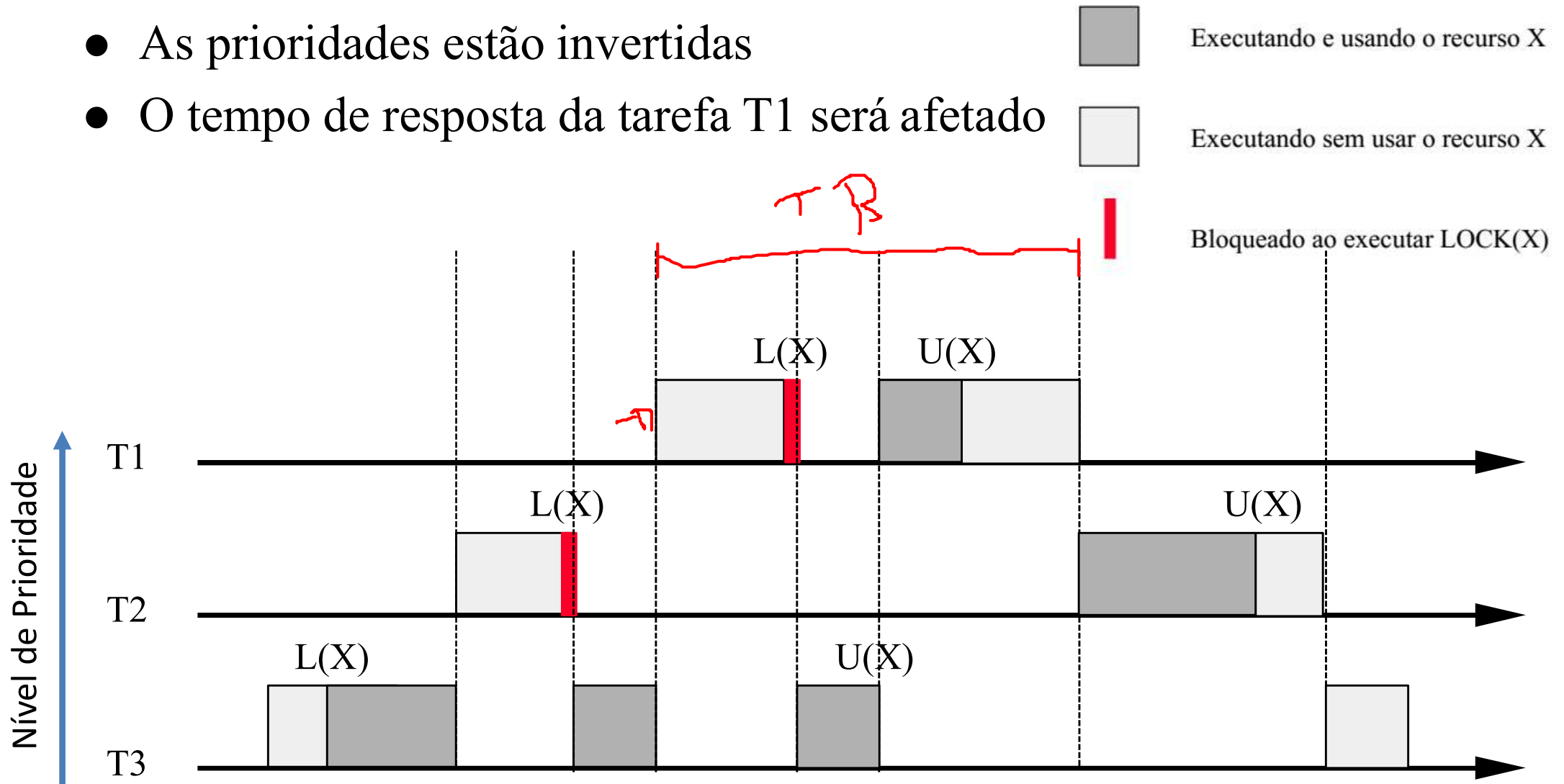
Modelo de Recursos

- Duas tarefas entram em conflito quando precisam usar o mesmo recurso ao mesmo tempo
- LOCK pode fazer a tarefa ficar bloqueada por algum tempo



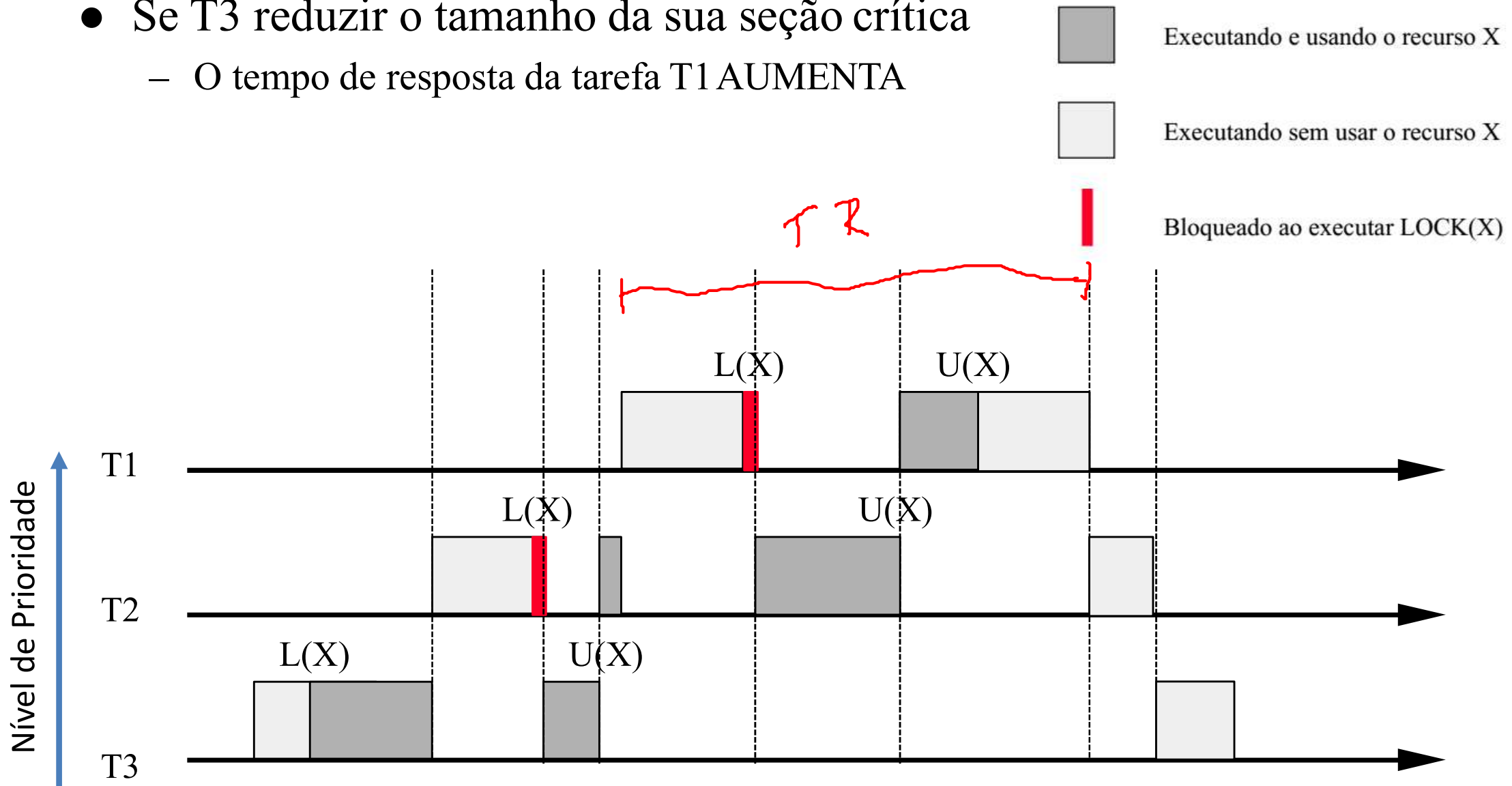
Exemplo de Inversão de Prioridade

- Temos em um dado momento a tarefa T1 esperando pela Tarefa T3
- As prioridades estão invertidas
- O tempo de resposta da tarefa T1 será afetado



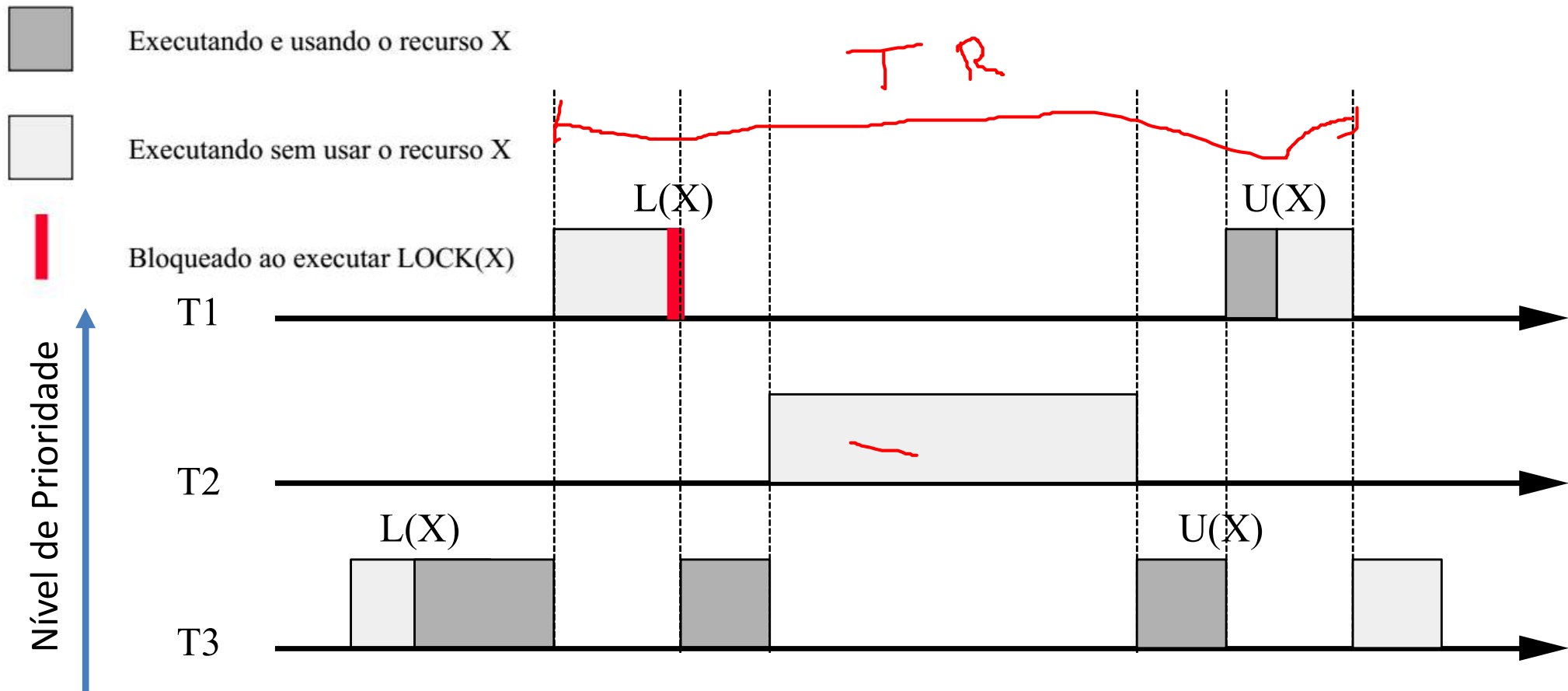
Anomalia da Inversão de Prioridades

- Se T3 reduzir o tamanho da sua seção crítica
 - O tempo de resposta da tarefa T1 AUMENTA



Inversão de Prioridade Descontrolada

- Pode ser ainda pior
- Agora Tarefa T1 espera por T3 e também por T2
- Tarefa T1 nem sequer compartilha qualquer recurso com T2



Protocolos de Acesso a Recursos

- Usar protocolos de acesso a recursos específicos
- Podem minimizar a inversão de prioridade
- Podem impedir deadlocks
 - Um conjunto de tarefas estão bloqueadas a espera de evento que somente pode ser causado por tarefas deste conjunto