# Protocolos de Acesso a Recursos


**Fundamentos dos Sistemas de Tempo Real**

Rômulo Silva de Oliveira

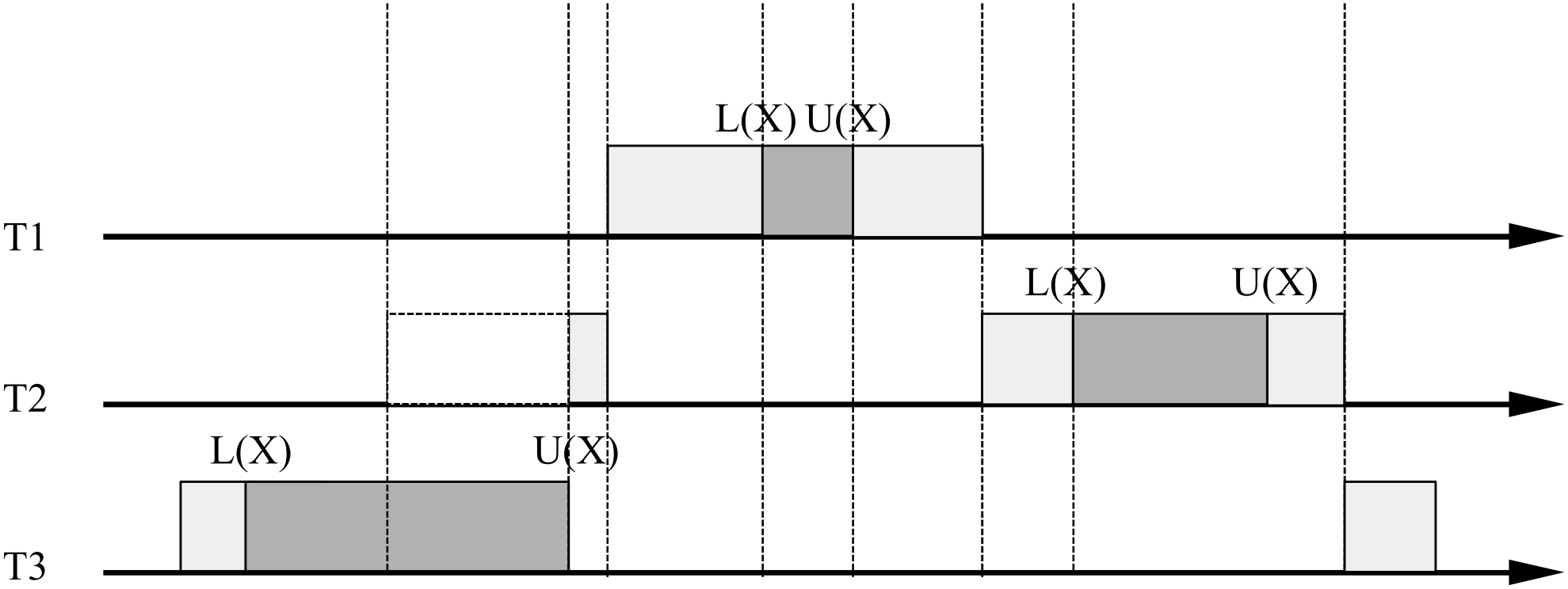
eBook Kindle, 2018

www.romulosilvadeoliveira.eng.br/livrotemporeal

Outubro/2018

# (1) Desliga Preempção

- Desliga a preempção antes de acessar qualquer recurso
- Todas as seções críticas executam de forma não preemptiva
- Tempo máximo de bloqueio Bi da tarefa Ti
  - Dado pela duração da maior seção crítica de qualquer tarefa com prioridade mais baixa do que Ti



# (1) Desliga Preempção

- Forma mais simples de resolver o problema
- Inversão de prioridade descontrolada não acontece
- Deadlock não acontece
- Qualquer tarefa pode ser bloqueada no máximo uma vez

- Obviamente só funciona em monoprocessador
- Qualquer tarefa pode ser bloqueada por qualquer tarefa de prioridade mais baixa
  - Mesmo que elas não compartilhem recursos entre si

- Solução razoável quando todas as seções críticas forem pequenas
- Corresponde ao “Desabilita Interrupções” de sistemas pequenos

# (2) Herança de Prioridade

- Cada tarefa possui uma prioridade nominal fixa
  - Dada por RM, DM, etc
- Cada job possui uma prioridade efetiva
  - A prioridade efetiva varia ao longo do tempo
  - A prioridade efetiva é usada para decidir quem executa a seguir
- Inicialmente,
  a prioridade efetiva do job é igual à prioridade nominal da sua tarefa
- A prioridade efetiva pode mudar em decorrência da alocação de recursos pelo job em questão

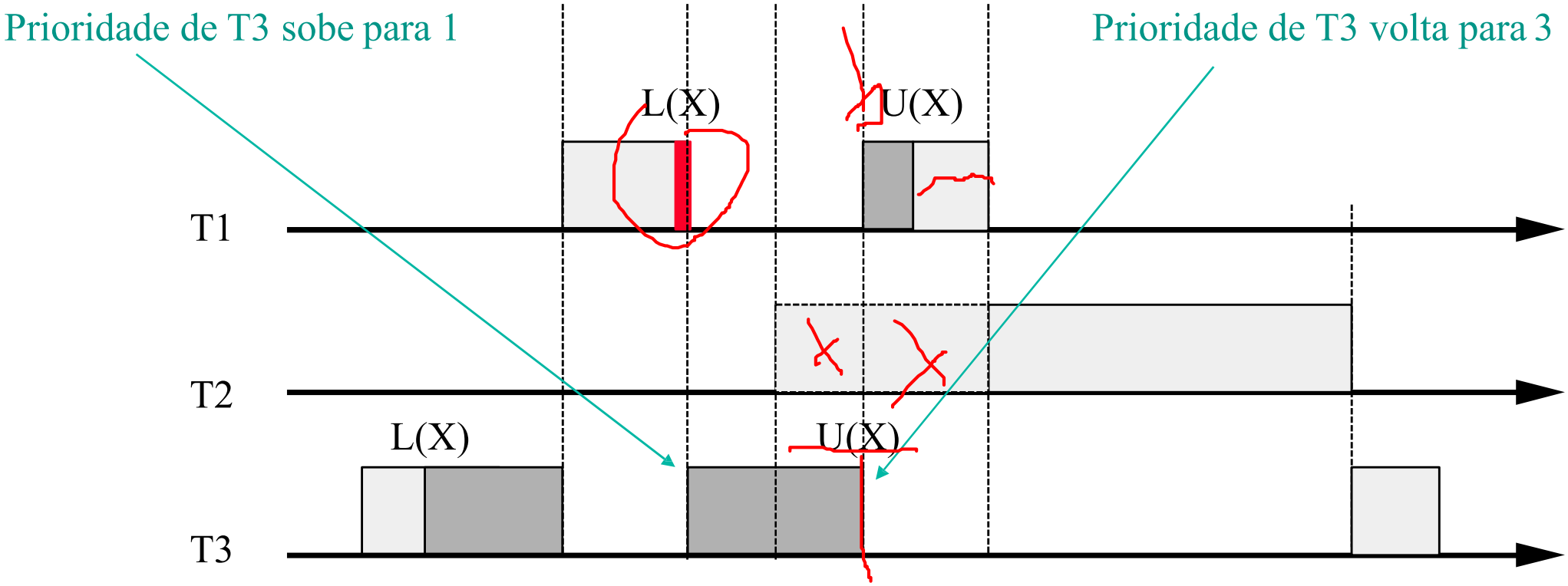
# (2) Herança de Prioridade: Regras

- Regra de alocação: Quando um Job J requer (LOCK) um recurso R no instante t:
  - Se R está livre, R é alocado para J até que J libere o recurso (UNLOCK)
  - Se R está ocupado, J bloqueia

- Regra da Herança de Prioridade:
  - Seja Jx o job que detém o recurso R solicitado por J
  - Job Jx herda a prioridade efetiva de J
  - Job Jx executa com a prioridade efetiva de J até liberar R
  - Quando liberar R, Jx retorna para a prioridade efetiva que tinha em t
  - Herança de prioridade é transitiva

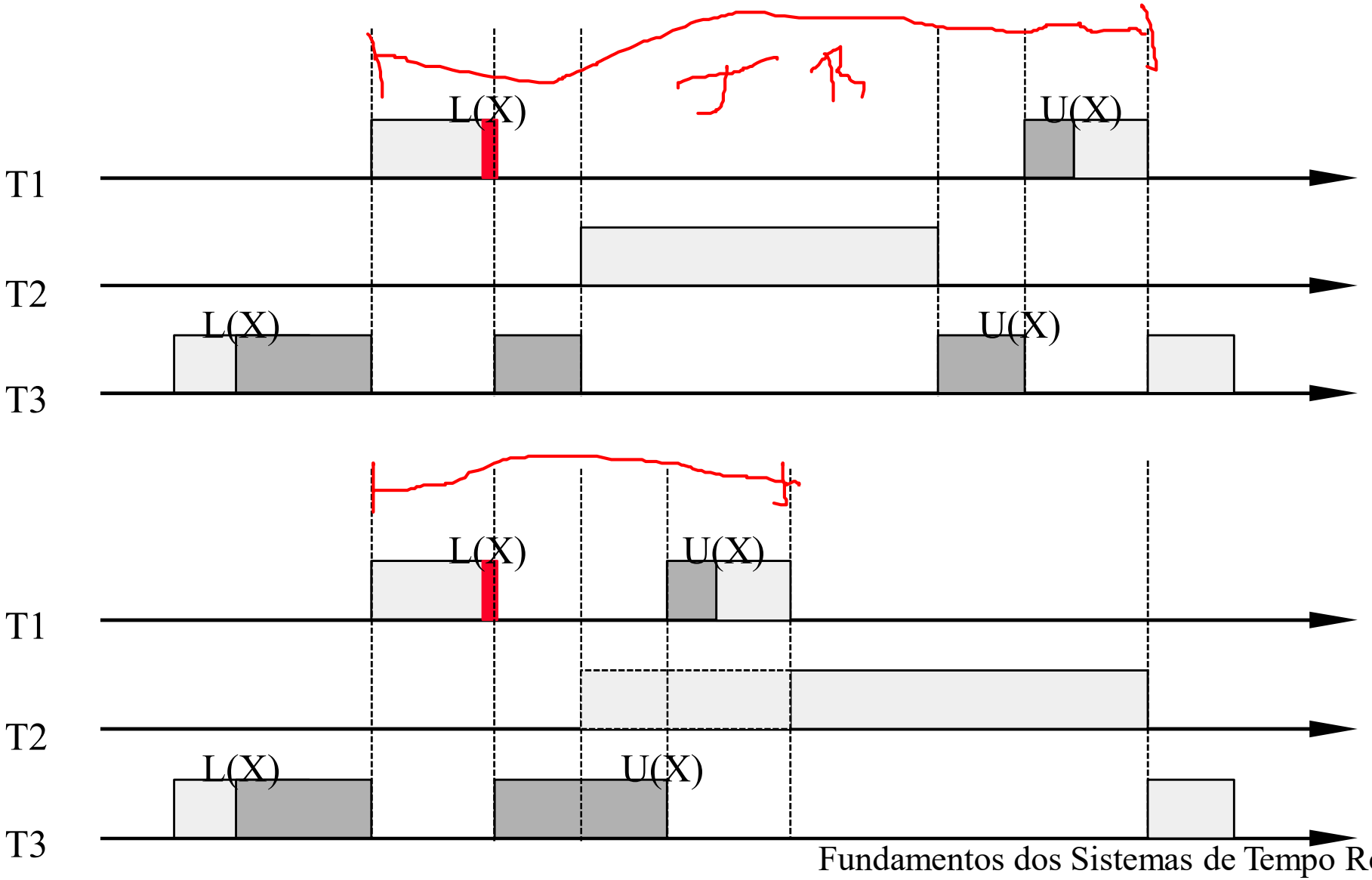
# (2) Herança de Prioridade: Exemplo

- Inversão de prioridades descontrolada é evitada
- T3 termina a seção crítica mais cedo
- T1 sofre bloqueio direto de T3
- T2 sofre bloqueio por herança de T3

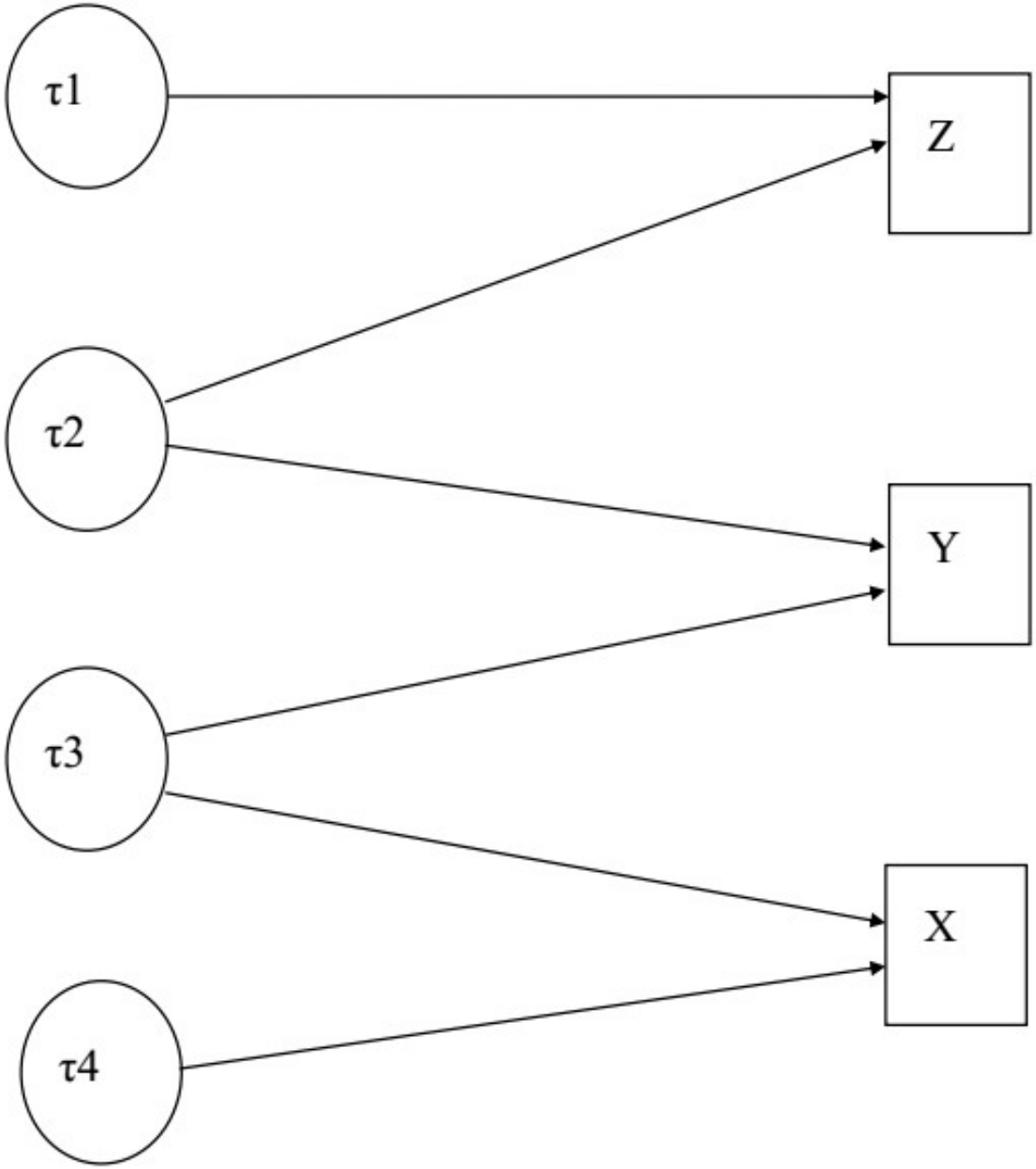


# (2) Herança de Prioridade: Exemplo

- Sem herança e com herança de prioridades



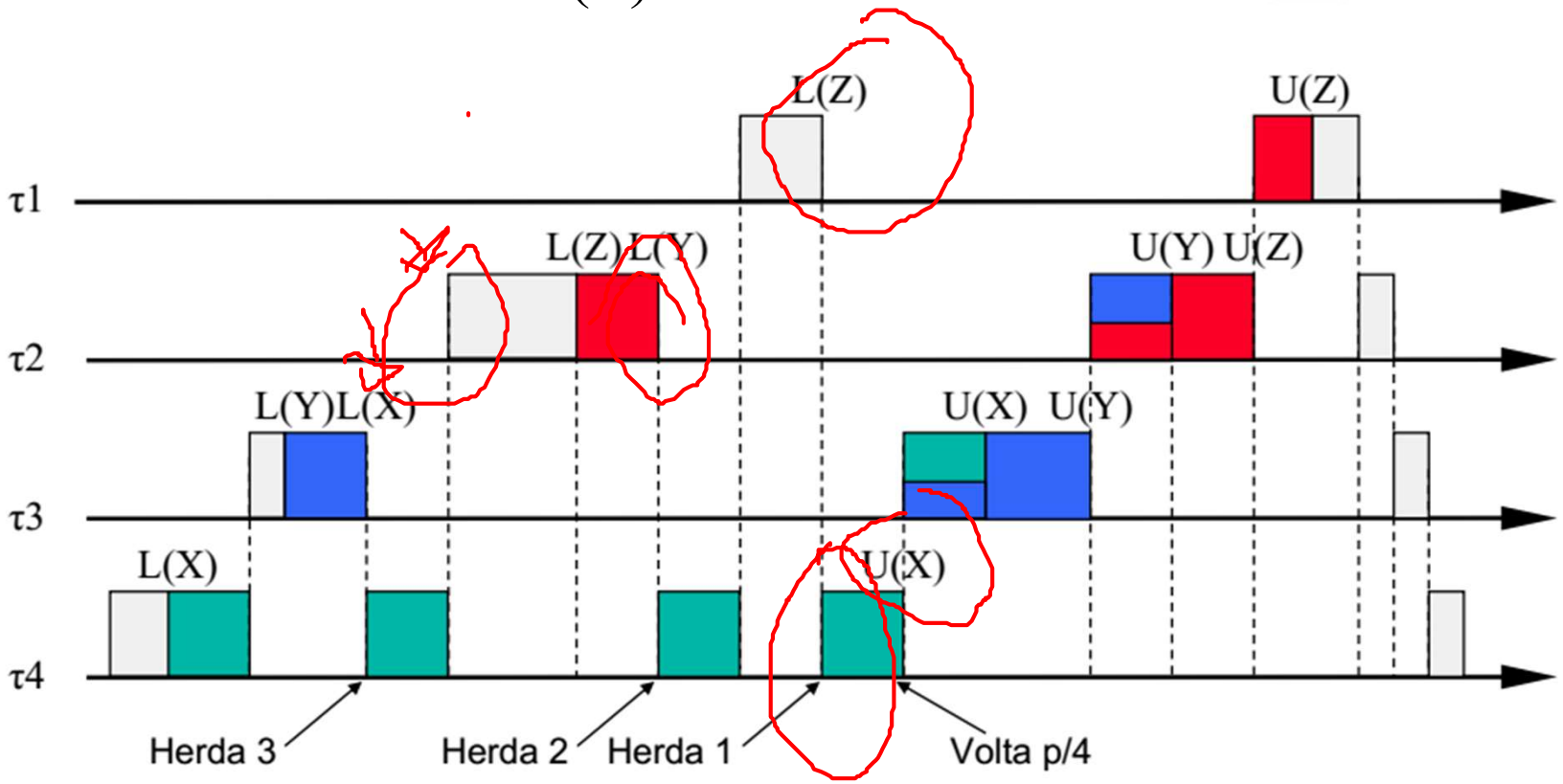
# (2) Herança de Prioridade: Exemplo




Grafo descrevendo sistema com 4 tarefas e 3 recursos.

# (2) Herança de Prioridade: Exemplo

- Bloqueio direto de τ2 sobre τ1 por Z
- Bloqueio transitivo de τ3 sobre τ1 via τ2(Y)
- Bloqueio transitivo de τ4 sobre τ1 via τ3(X) e τ2(Y)
- Suponha τ4 executa Lock(Z) em t



# (2) Herança de Prioridade

- Impede a inversão de prioridade descontrolada
- Uma tarefa pode ser bloqueada diretamente por outra tarefa de mais baixa prioridade somente uma vez
  - Isto pode acontecer para cada seção crítica usada
- Cada uma das tarefas de mais baixa prioridade é capaz de bloquear a tarefa de mais alta prioridade
  - Por no máximo uma vez
- Tipos de bloqueio
  - Direto ($\tau 2$ bloqueia $\tau 1$)
  - Transitivo ($\tau 3$ bloqueia $\tau 2$ que por sua vez está bloqueando $\tau 1$)
  - Por herança de prioridade

# (2) Herança de Prioridade

- Não impede o deadlock por si só
  - Mecanismo adicional é necessário
- Não minimiza tempo de bloqueio no pior caso
- Determinação do B para o pior caso pode ficar complexo
  - se os padrões de uso dos recursos forem complexos

# (3) Teto de Prioridade

- Estende a herança de prioridade para
  - Eliminar a possibilidade de deadlock
  - Reduzir o tempo de bloqueio no pior caso
- Assume que é sabido antes da execução quais recursos cada tarefa usa
- Priority Ceiling = Teto de prioridade

- O teto de prioridade de um recurso Ri corresponde a mais alta prioridade entre todas as tarefas que usam Ri
  - Denotado por $\Pi i$
- Uma prioridade imaginária mais baixa que todas as prioridades do sistema será denotada por $\Omega$
- O teto de prioridade do sistema $\Pi j(t)$ corresponde ao maior teto de prioridade entre todos os recursos que se encontram alocados no instante t, desconsiderando-se aqueles alocados por $\tau j$

# (3) Teto de prioridade

- Cada tarefa possui uma prioridade nominal fixa
  - Dada por RM, DM, etc
- Cada job possui uma prioridade efetiva
  - A prioridade efetiva varia ao longo do tempo
  - A prioridade efetiva é usada para decidir quem executa a seguir
- Inicialmente,
  a prioridade efetiva do job é igual à prioridade nominal da sua tarefa
- A prioridade efetiva pode mudar em decorrência da alocação de recursos pelo job em questão
- As prioridades efetivas dos jobs são usadas para fins de escalonamento
  - Executa o job não bloqueado com a prioridade efetiva mais alta

# (3) Teto de prioridade: Regra de Alocação

- Regra de Alocação
- Um job J requer um recurso R no instante t

- Se R está alocado para outro job
  - O pedido é negado e J fica bloqueado

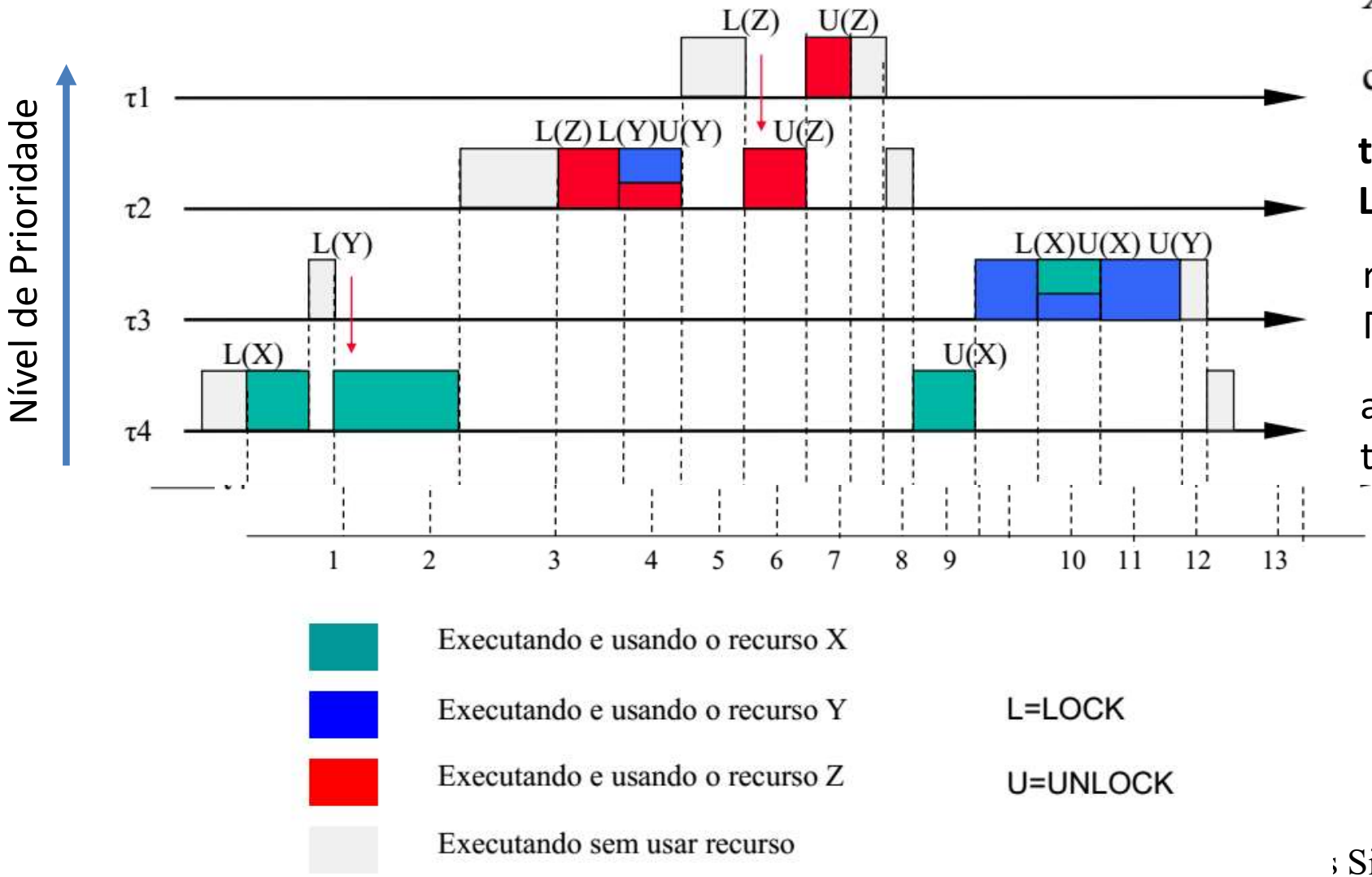
- Se R está livre
  - O pedido será aceito se a prioridade efetiva de J for maior que o teto de prioridade do sistema Πj, o qual desconsidera os recursos que J alocou
  - Caso contrário, o pedido é negado e J fica bloqueado

# (3) Teto de prioridade: Regra de Herança

- Regra de herança de prioridade
- Quando o job J tenta alocar o recurso R e fica bloqueado
  - o job Jx que detém o recurso R herda a prioridade efetiva de J
- Jx mantém esta prioridade até o momento que libera o recurso R
  - Neste instante ele retorna para sua prioridade efetiva anterior
- A herança de prioridade é transitiva

# (3) Teto de Prioridade: Exemplo

- Tetos de prioridade:
  - $\Pi(X)=3, \Pi(Y)=2, \Pi(Z)=1$




- Inicialmente chega τ4

X está livre e Π4(1)=Ω

chega τ3 e preempta τ4

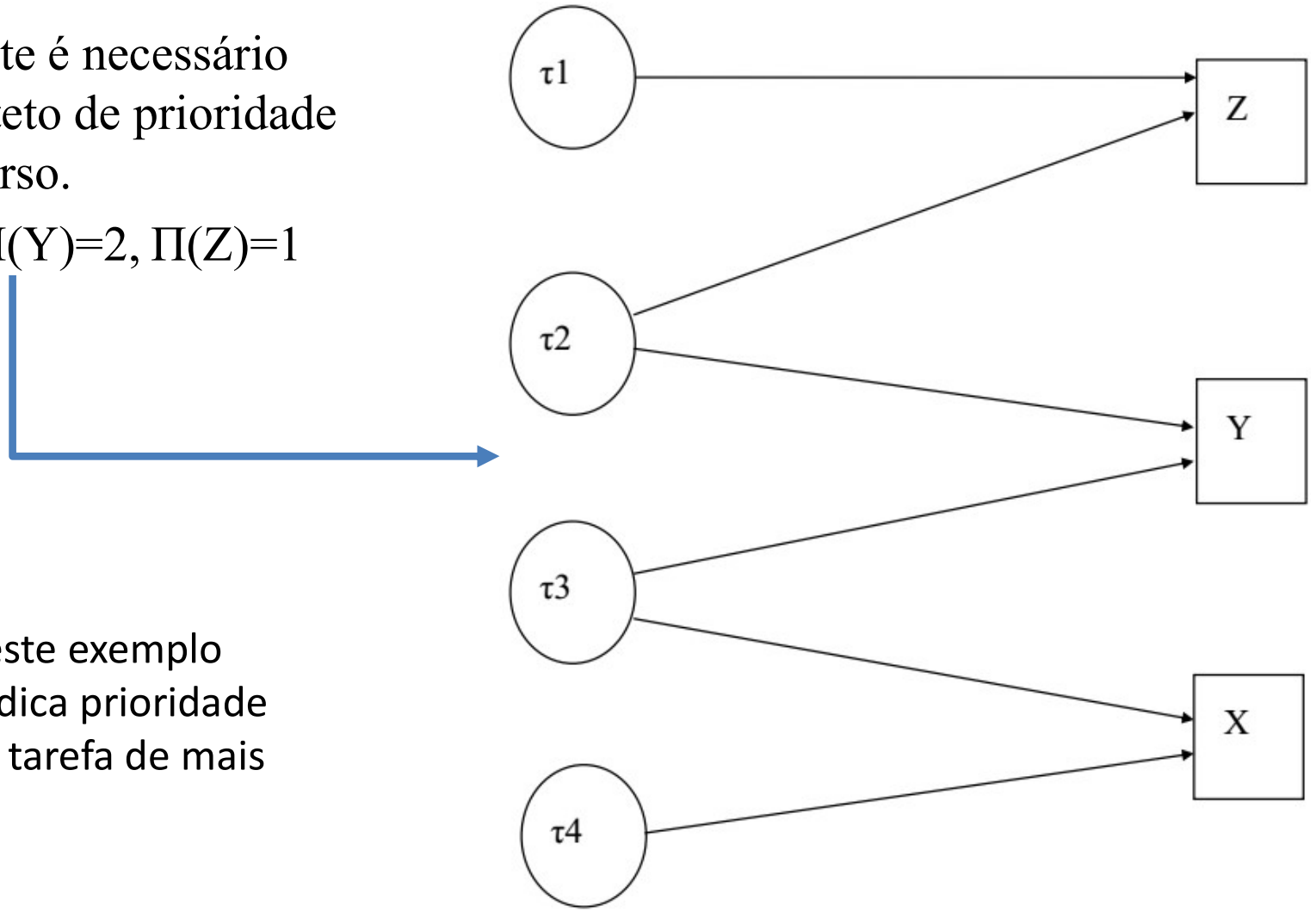
**tarefa τ3 chama LOCK(Y) no instante 2**

recurso Y está livre mas Π3(2)=3

a tarefa τ3 bloqueia e a tarefa τ4 herda sua ridade.

# (3) Teto de Prioridade: Exemplo

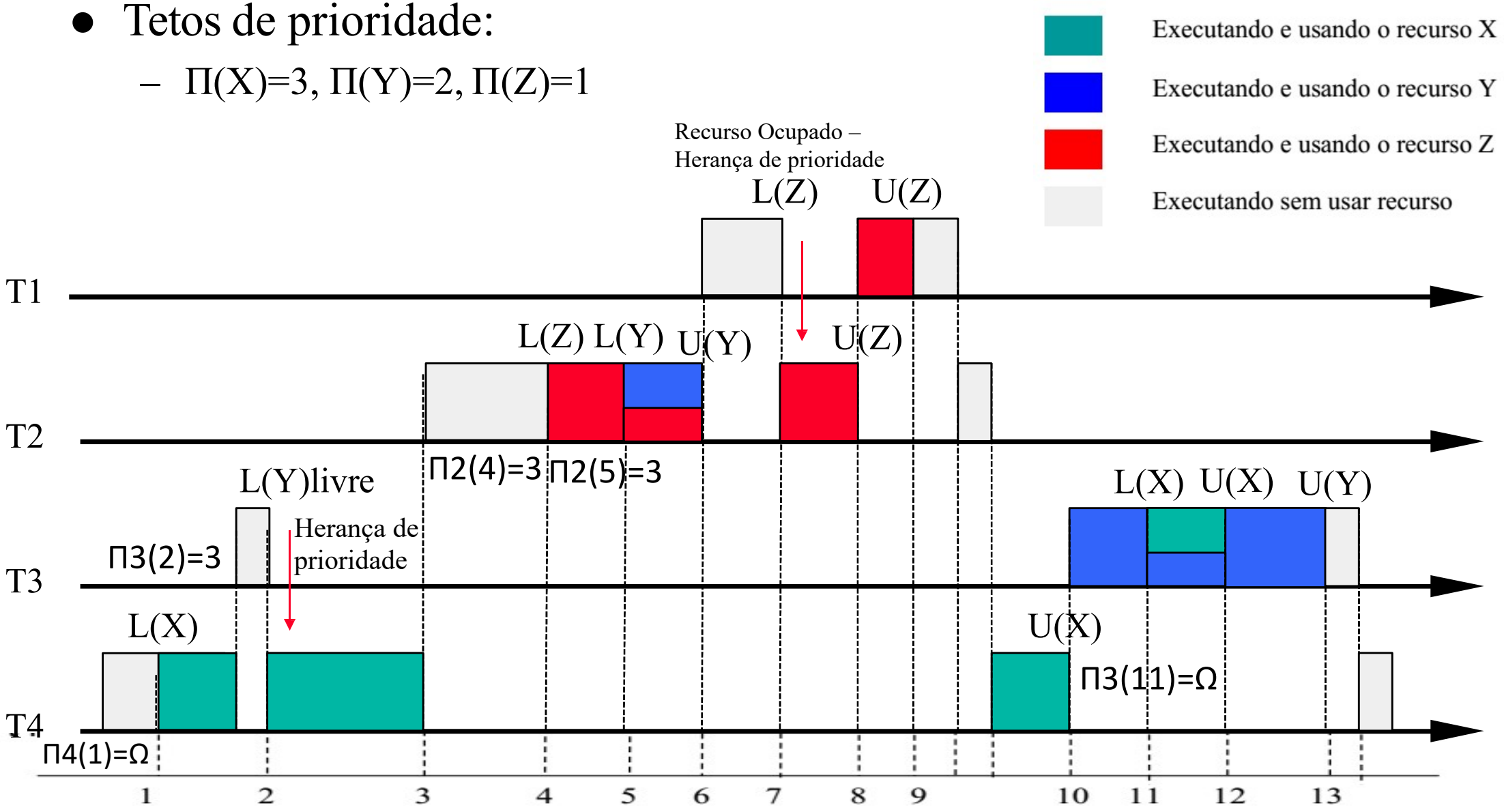
- Tetos de prioridade:
  - Inicialmente é necessário computar o teto de prioridade de cada recurso.
  - $\Pi(X)=3, \Pi(Y)=2, \Pi(Z)=1$




Lembre-se que neste exemplo número menor indica prioridade mais alta, e τ1 é a tarefa de mais alta prioridade.

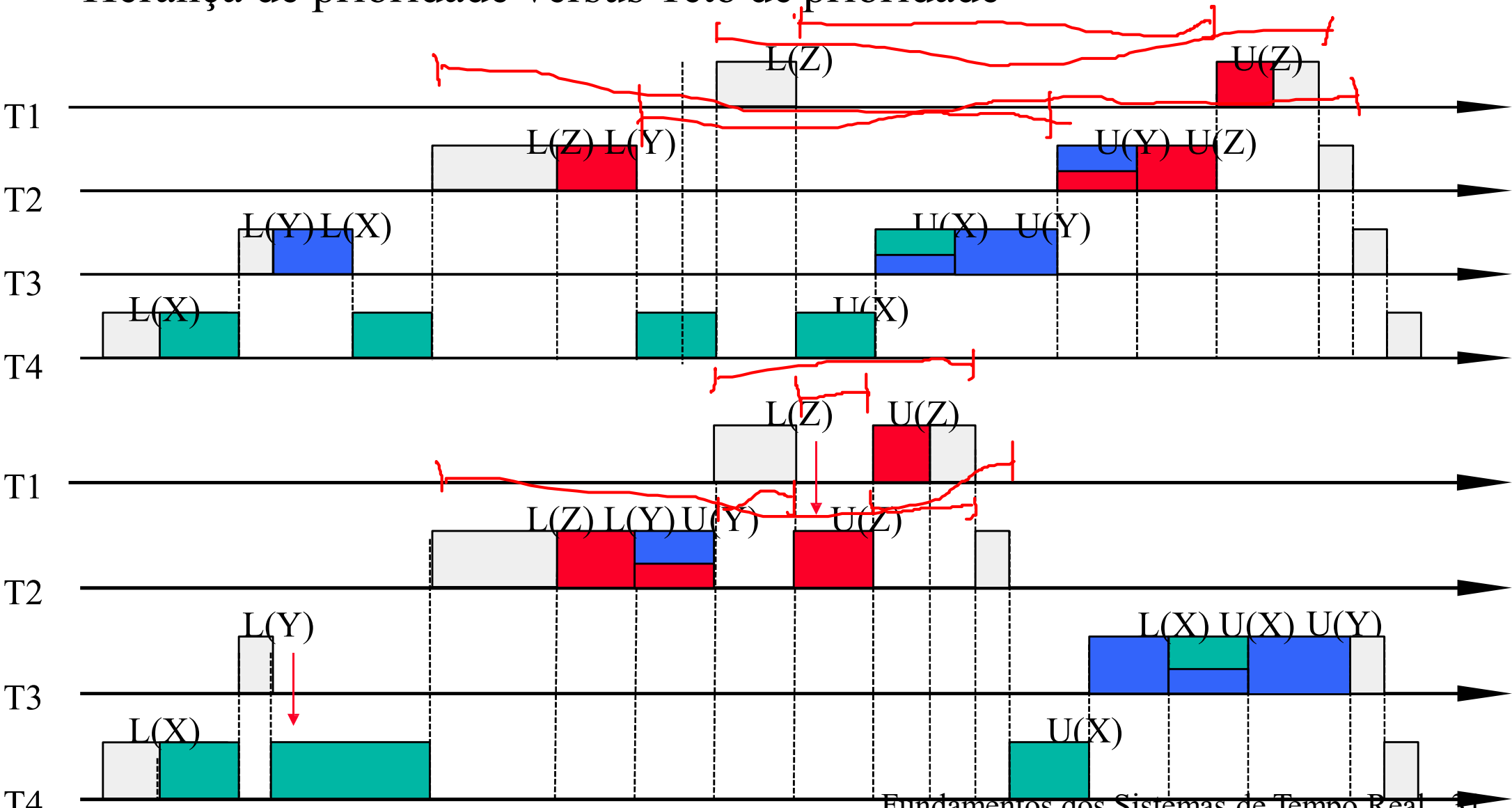
# (3) Teto de Prioridade: Exemplo

- Tetos de prioridade:
  - Π(X)=3, Π(Y)=2, Π(Z)=1



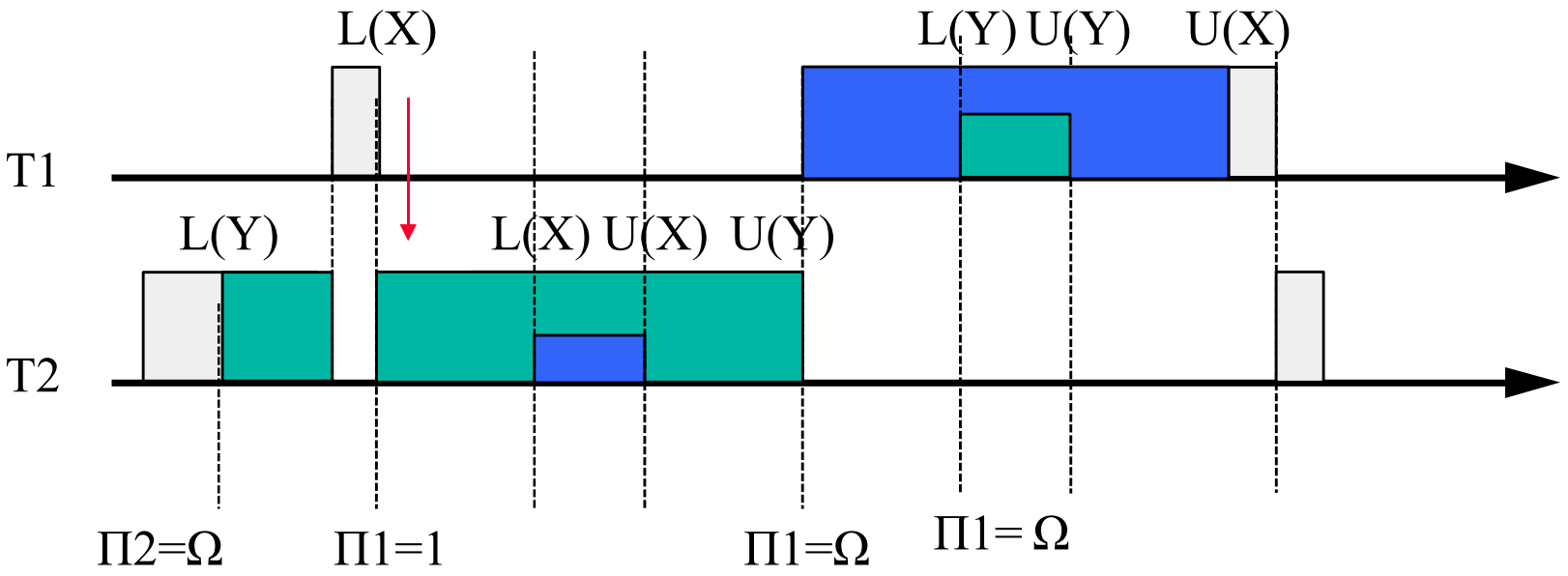
# (3) Teto de Prioridade: Exemplo

- Herança de prioridade versus Teto de prioridade



# (3) Teto de Prioridade

- Impossível ocorrer deadlock
  - T1:[X,1[Y,1]]
  - T2:[Y,1[X,1]]
  - Π(X)=1, Π(Y)=1



# (3) Teto de Prioridade

- Herança de Prioridade
  - Guloso
  - Se recurso estiver livre, o mesmo é alocado
- Teto de Prioridade
  - Conservador
  - Mesmo um recurso livre pode não ser alocado
  - Isto é feito para prevenir um comportamento pior mais adiante
- Semelhante aos protocolos para evitar deadlock
  - Algoritmo do banqueiro

# (3) Priority Ceiling: Tipos de Bloqueios

- Bloqueio pode ocorrer de três formas
- Bloqueio direto
  - Recurso está ocupado
- Bloqueio por herança de prioridade
  - Bloqueado por tarefa que herdou prioridade mais alta de outra tarefa
- Bloqueio por teto
  - Recurso está livre, mas teto do sistema é mais alto, alocação é negada

- Determinação do B é simplificada pois:
  - Um job pode ser bloqueado por no máximo a duração de uma seção crítica

# (3) Teto de Prioridade

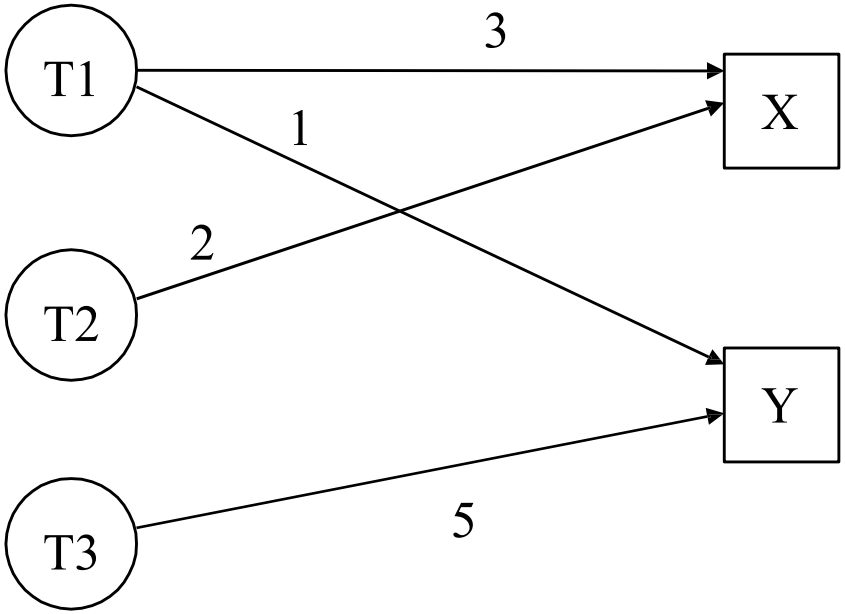
- Uma tarefa Ti pode ser bloqueada apenas uma vez por uma mesma tarefa de prioridade mais baixa Tx
  - Ao liberar o recurso a primeira vez, Tx não executa mais e não aloca novamente até o final de Ti

- Não é possível uma tarefa Ti ser bloqueada por Tx e Ty, se Tx e Ty tiverem prioridade mais baixa do que Ti
  - Após o primeiro bloqueio, a regra do teto impede um segundo bloqueio

- Uma tarefa Ti pode ser bloqueada no máximo pela duração de uma única seção crítica
  - Não importa quantas tarefas compartilham recursos com a tarefa Ti
  - Vale a seção crítica mais externa, quando aninhadas

# (3) Teto de Prioridade

- $\tau 1$
  - Bloqueio direto de $\tau 2$ por 2 ut
  - Bloqueio direto de $\tau 3$ por 5 ut
  - Bloqueio por teto de $\tau 3$ por 5 ut
  - Somente um dos dois é possível
- $\tau 2$
  - Bloqueio por teto de $\tau 3$ por 5 ut
  - Bloqueio por herança de $\tau 3$ por 5 ut
  - Somente um dos dois é possível
- $\tau 3$
  - Não sofre bloqueio por definição



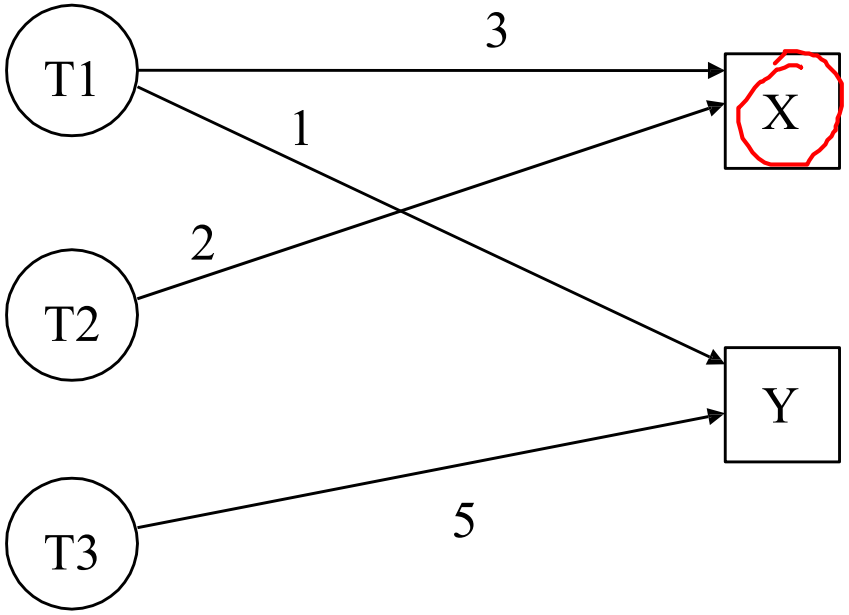
# Resumo – Tempo Máximo de bloqueio

- Desliga Preempção
  - Tempo máximo de bloqueio Bi da tarefa Ti
    - Dado pela duração da maior seção crítica de qualquer tarefa com prioridade mais baixa do que Ti

- Teto de prioridade
  - Tempo máximo de bloqueio Bi da tarefa Ti
    - Busque tarefas menos prioritárias que usam mutex com teto igual ou mais alto

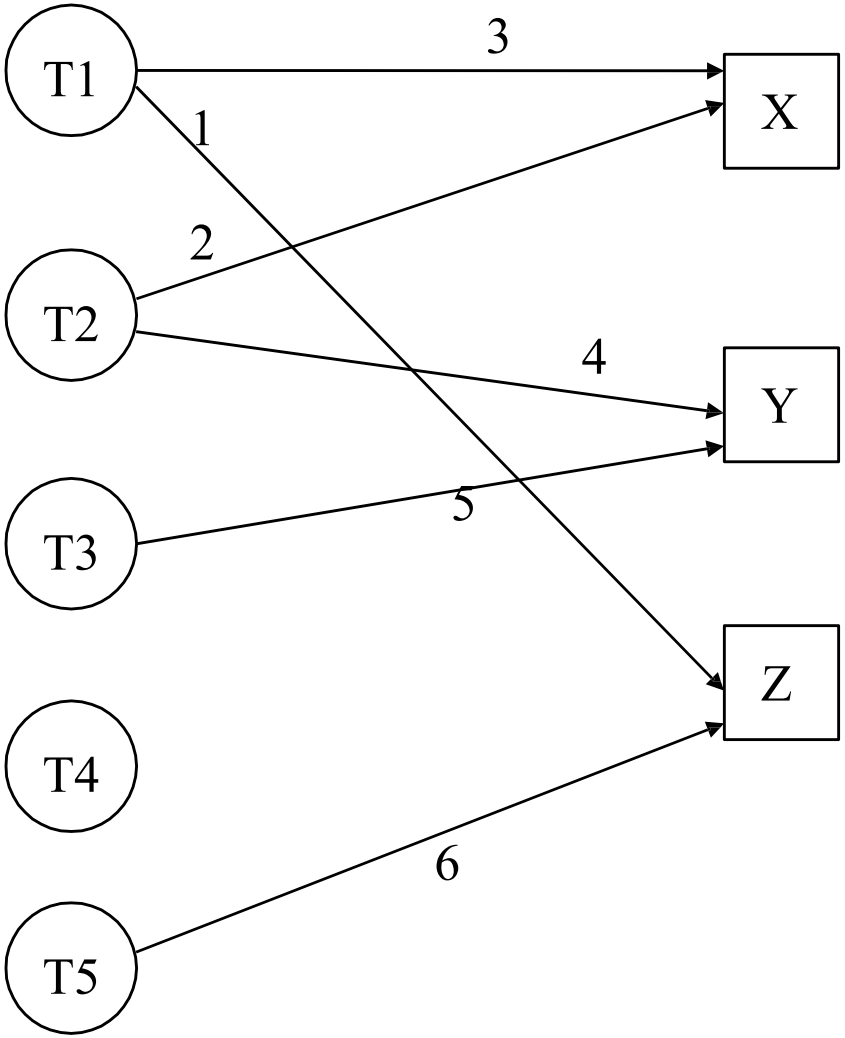
# (3) Teto de Prioridade

- A determinação de B3, B2 e B1 pode ser resumida desta forma:
  - Busque tarefas menos prioritárias que usam mutex com teto igual ou mais alto
- τ3: nenhuma tarefa com prioridade mais baixa
  - B3 = 0
- τ2:
  - τ3 usa [Y,5] Πy=1 ok 5
  - B2 = 5
- τ1:
  - τ3 usa [Y,5] Πy=1 ok 5
  - τ2 usa [X,2] Πx=1 ok 2
  - B1 = 5



# (3) Teto de Prioridade

- Exemplo mais complexo



# (3) Teto de Prioridade

- τ5: nenhuma tarefa com prioridade mais baixa
  - B5 = 0
- τ4: τ5 usa [Z,6] Πz=1 ok 6
  - B4 = 6
- τ3: τ5 usa [Z,6] Πz=1 ok 6
  - B3 = 6