Rate Monotonic Analysis

Introduction

Periodic tasks

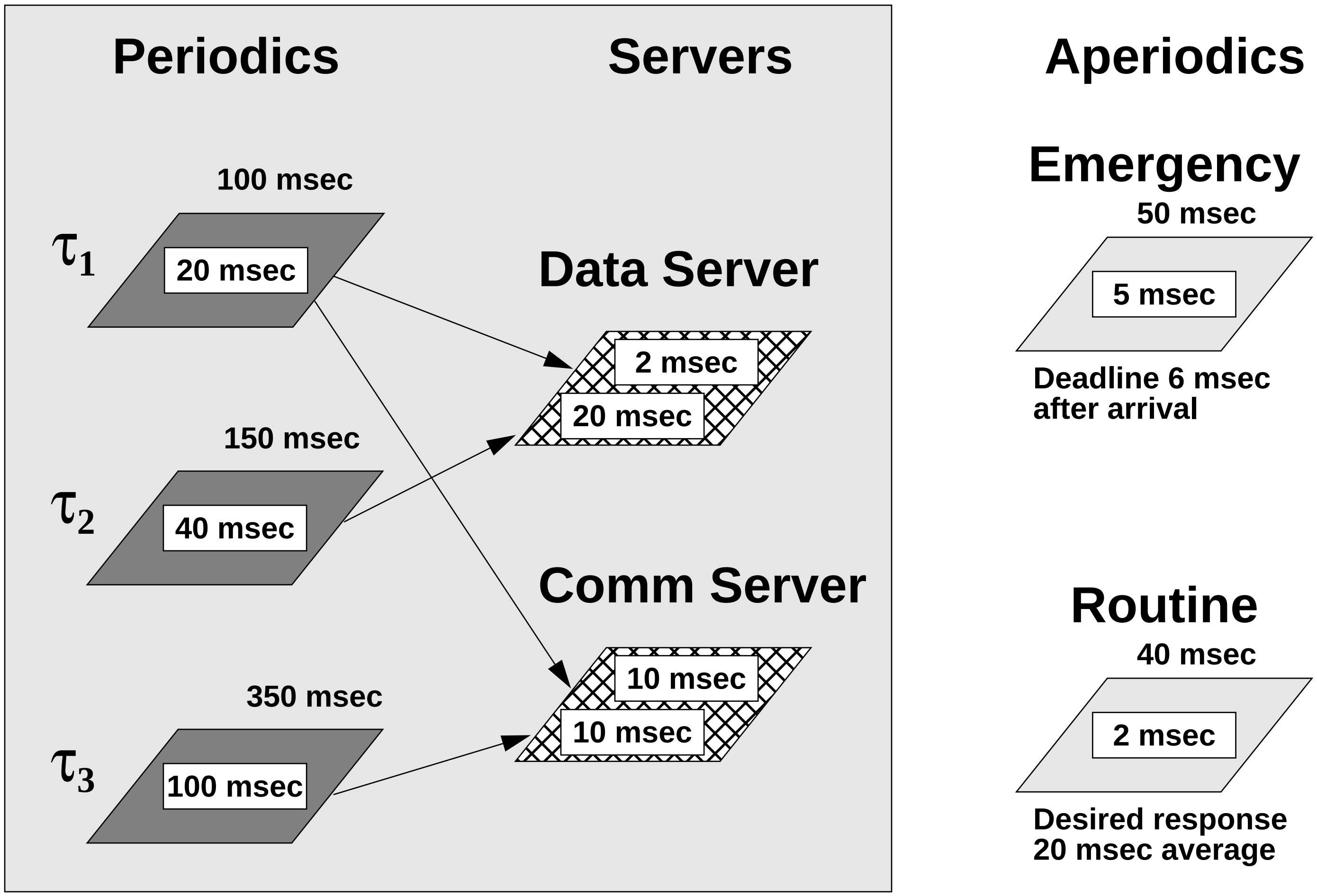
Extending basic theory

Synchronization and priority inversion

Aperiodic servers

Case study: BSY-1 Trainer

Sample Problem: Synchronization

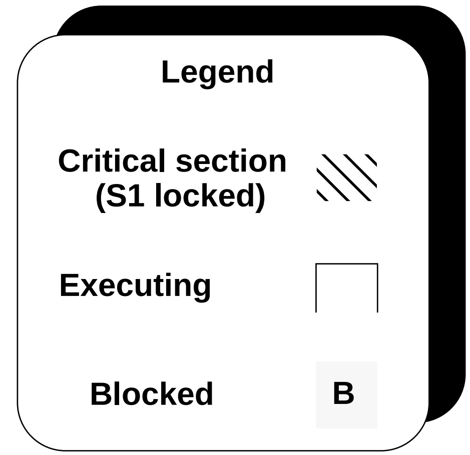


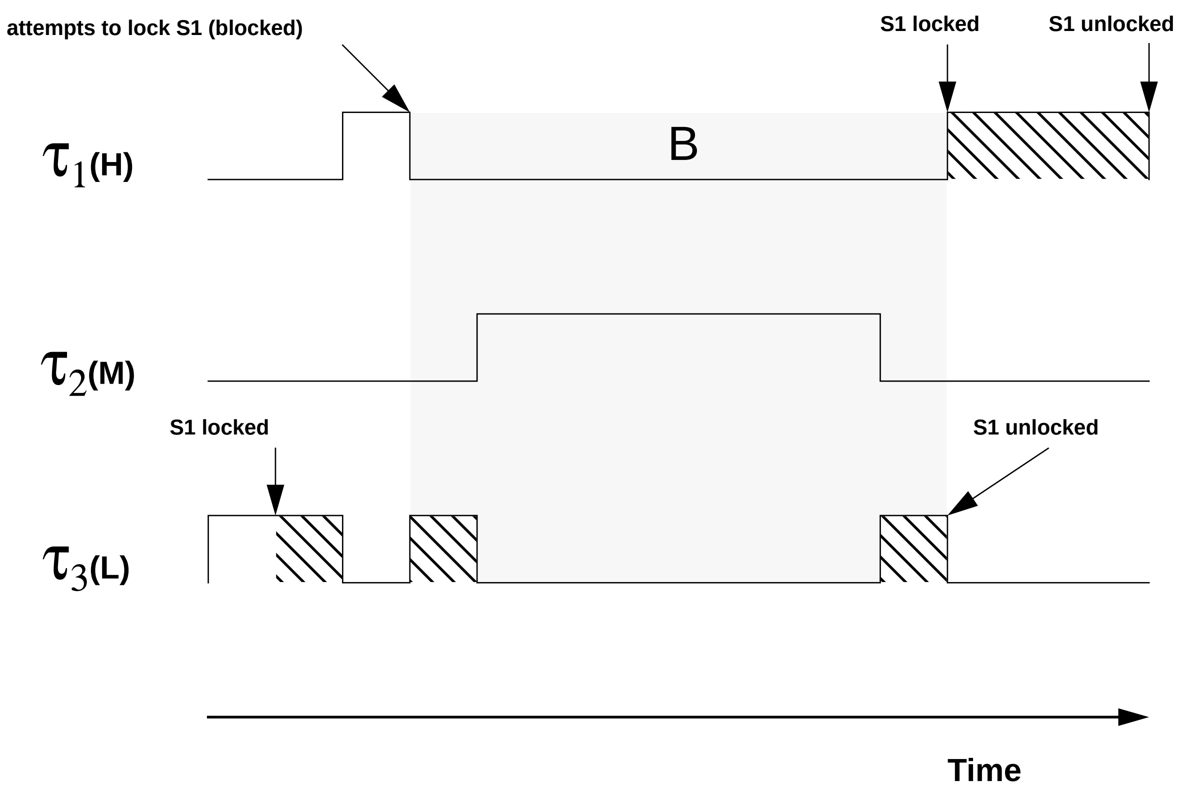
τ_2's deadline is 20 msec before the end of each period.

Priority Inversion in Synchronization

τ_1:{...P(S1)...V(S1)...}

τ_3:{...P(S1)...V(S1)...}





Priority Inversion

Delay to a task's execution caused by interference from lower priority tasks is known as *priority inversion*.

Priority inversion is modeled by blocking time.

Identifying and evaluating the effect of sources of priority inversion is important in schedulability analysis.

Sources of Priority Inversion

Synchronization and mutual exclusion

Non-preemptable regions of code

FIFO (first-in-first-out) queues

Accounting for Priority Inversion

Recall that task schedulability is affected by

- **preemption: two types of preemption**
  - **can occur several times per period**
  - **can only occur once per period**
- **execution: once per period**
- **blocking: at most once per period for each source**

The schedulability formulas are modified to add a "blocking" or "priority inversion" term to account for inversion effects.

UB Test with Blocking

Include blocking while calculating effective utilization for each tasks:

$$f_i = \boxed{\sum_{j \in Hn} \frac{C_j}{T_j}} + \boxed{\frac{C_i}{T_i}} + \boxed{\frac{B_i}{T_i}} + \boxed{\frac{1}{T_i} \sum_{k \in H1} C_k}$$

***Hn* Preemption (can hit *n* times)**

Execution

Blocking

***H1* Preemption (can hit once)**

RT Test with Blocking

Blocking is also included in the RT test:

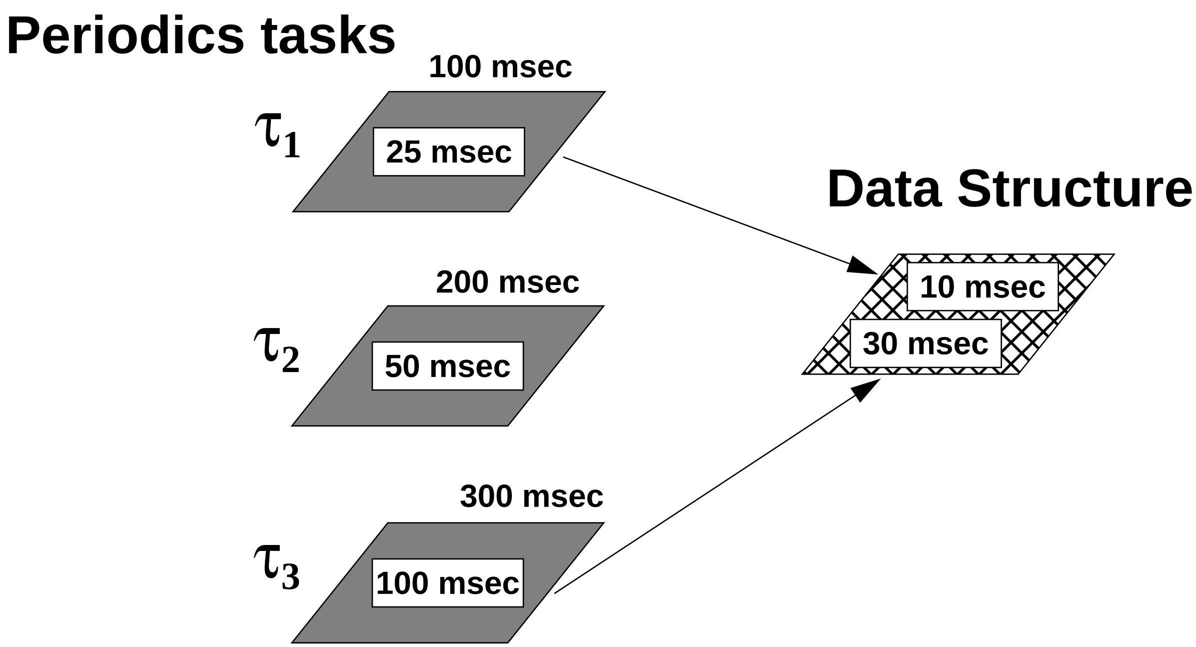
$$a_{n+1} = B_{\boldsymbol{i}} + \boldsymbol{C}_{\boldsymbol{i}} + \sum_{j=1}^{i-1} \left\lceil \frac{a_n}{T_j} \right\rceil C_j$$

$$\text{where } a_0 = B_{\boldsymbol{i}} + \sum_{j=1}^{i} C_j$$

Perform test as before, including blocking effect.

Example: Considering Blocking

Consider the following example:



What is the worst-case blocking effect (priority inversion) experienced by each task?

Example: Adding Blocking

Task τ_2 does not use the data structure. Task τ_2 experiences no priority inversion.

Task τ_1 shares the data structure with τ_3. Task τ_1 could have to wait for τ_3 to complete its critical section. But worse, if τ_2 preempts while τ_1 is waiting for the data structure, τ_1 could have to wait for τ_2's entire computation.

This is the resulting table:

Task	Period	Execution Time	Priority	Blocking Delays	Deadline
τ_1	100	25	High	30+50	100
τ_2	200	50	Medium	0	200
τ_3	300	100	Low	0	300

UB Test for Example

Recall UB test with blocking:

$$f_i = \sum_{j \in Hn} \frac{C_j}{T_j} + \frac{C_i}{T_i} + \frac{B_i}{T_i} + \frac{1}{T_i} \sum_{k \in H1} C_k$$

$$f_1 = \frac{C_1}{T_1} + \frac{B_1}{T_1} = \frac{25}{100} + \frac{80}{100} = 1.05 > 1.00$$ Not schedulable

$$f_2 = \frac{C_1}{T_1} + \frac{C_2}{T_2} = \frac{25}{100} + \frac{50}{200} = 0.50 < U(2)$$

$$f_3 = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{25}{100} + \frac{50}{200} + \frac{100}{300} = 0.84 > U(3)$$

RT test shows
τ_3 is schedulable

Synchronization Protocols

No preemption

Basic priority inheritance

Highest locker’s priority

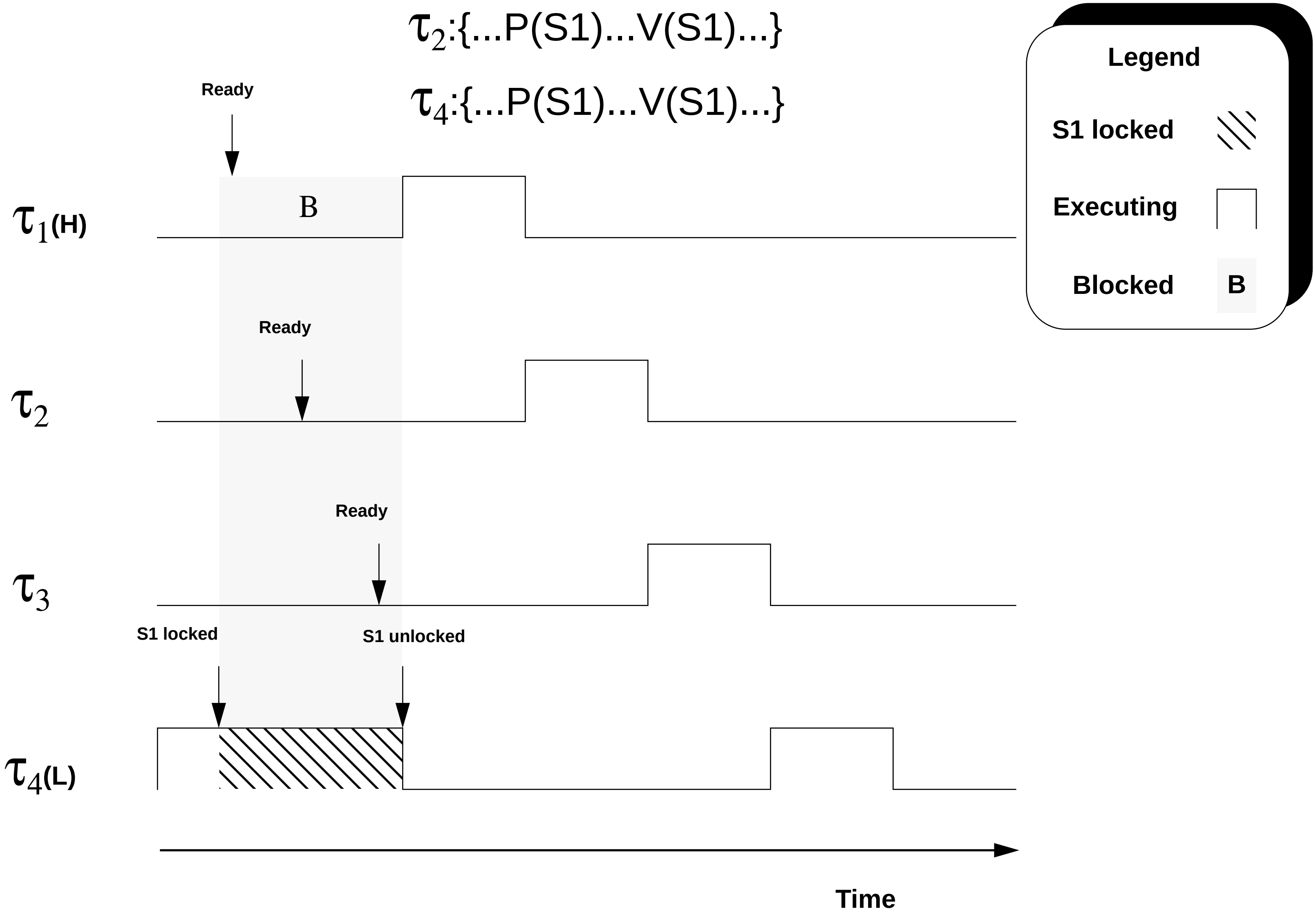
Priority ceiling

Each protocol prevents unbounded priority inversion.

Nonpreemption Protocol

τ_2:{...P(S1)...V(S1)...}

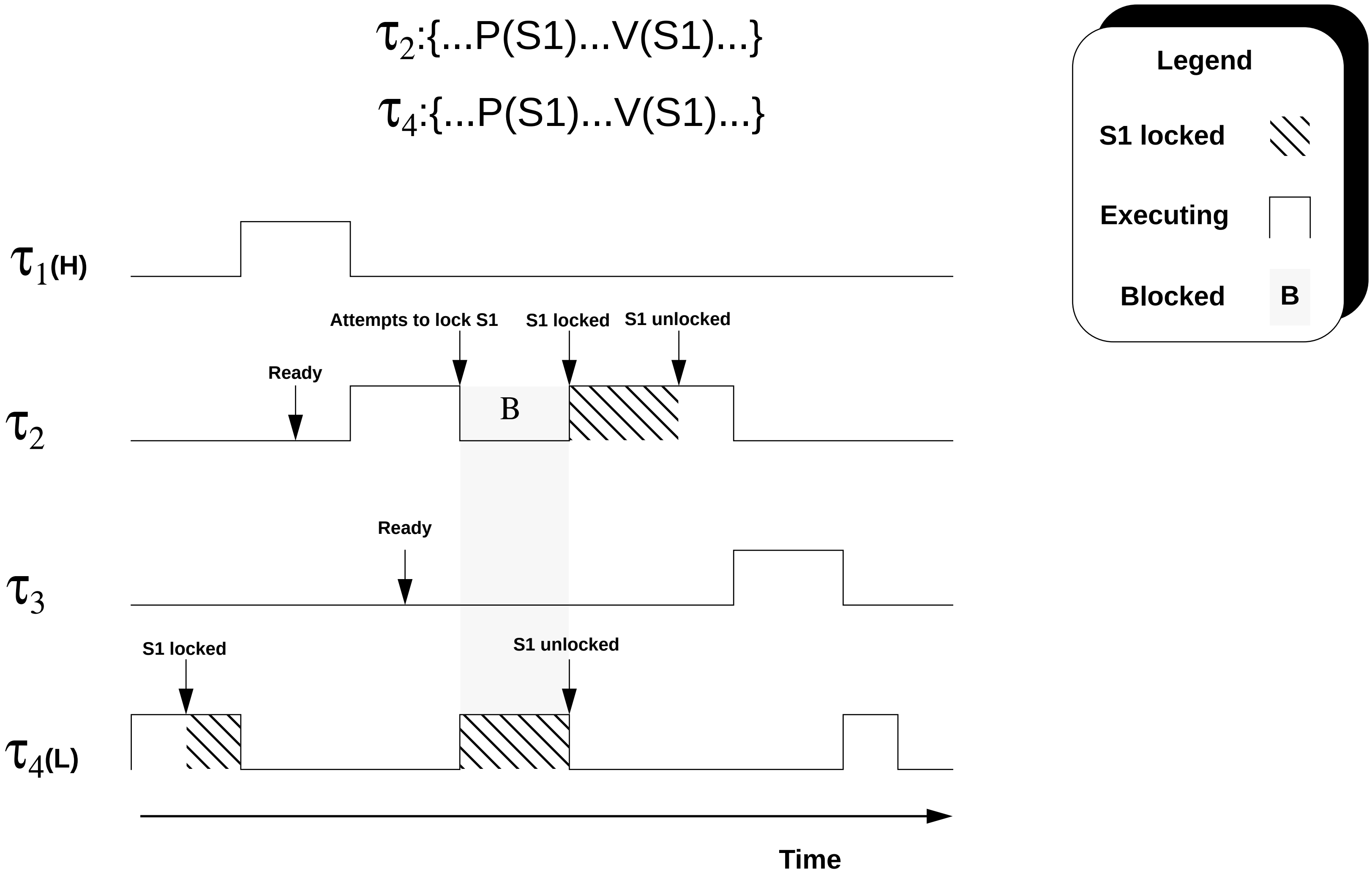
τ_4:{...P(S1)...V(S1)...}



Basic Inheritance Protocol (BIP)

τ_2:{...P(S1)...V(S1)...}

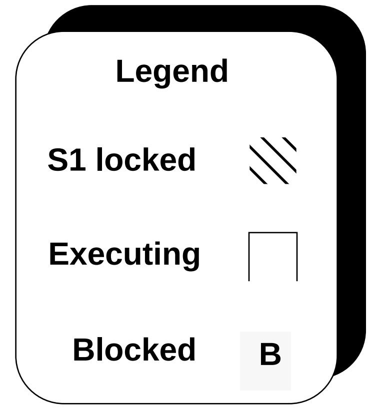
τ_4:{...P(S1)...V(S1)...}

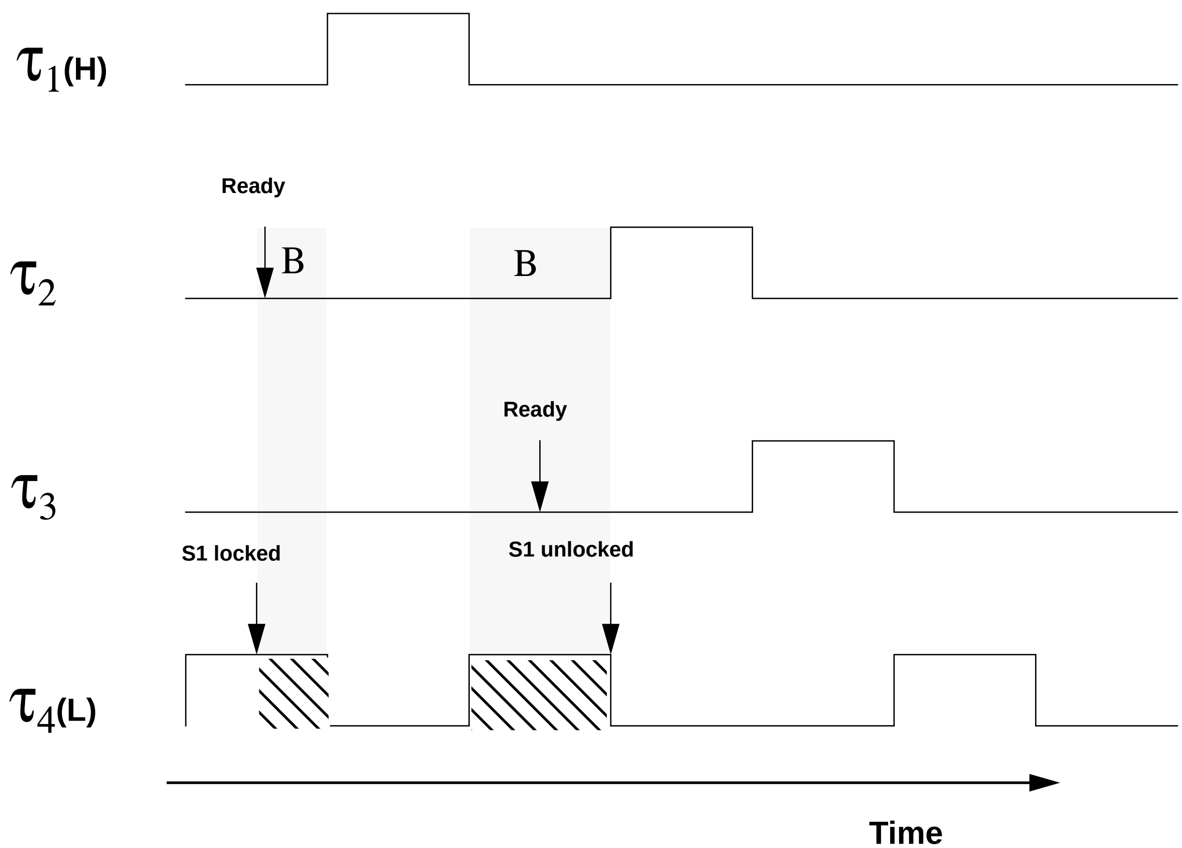


Highest Locker's Priority Protocol

τ_2:{...P(S1)...V(S1)...}

τ_4:{...P(S1)...V(S1)...}

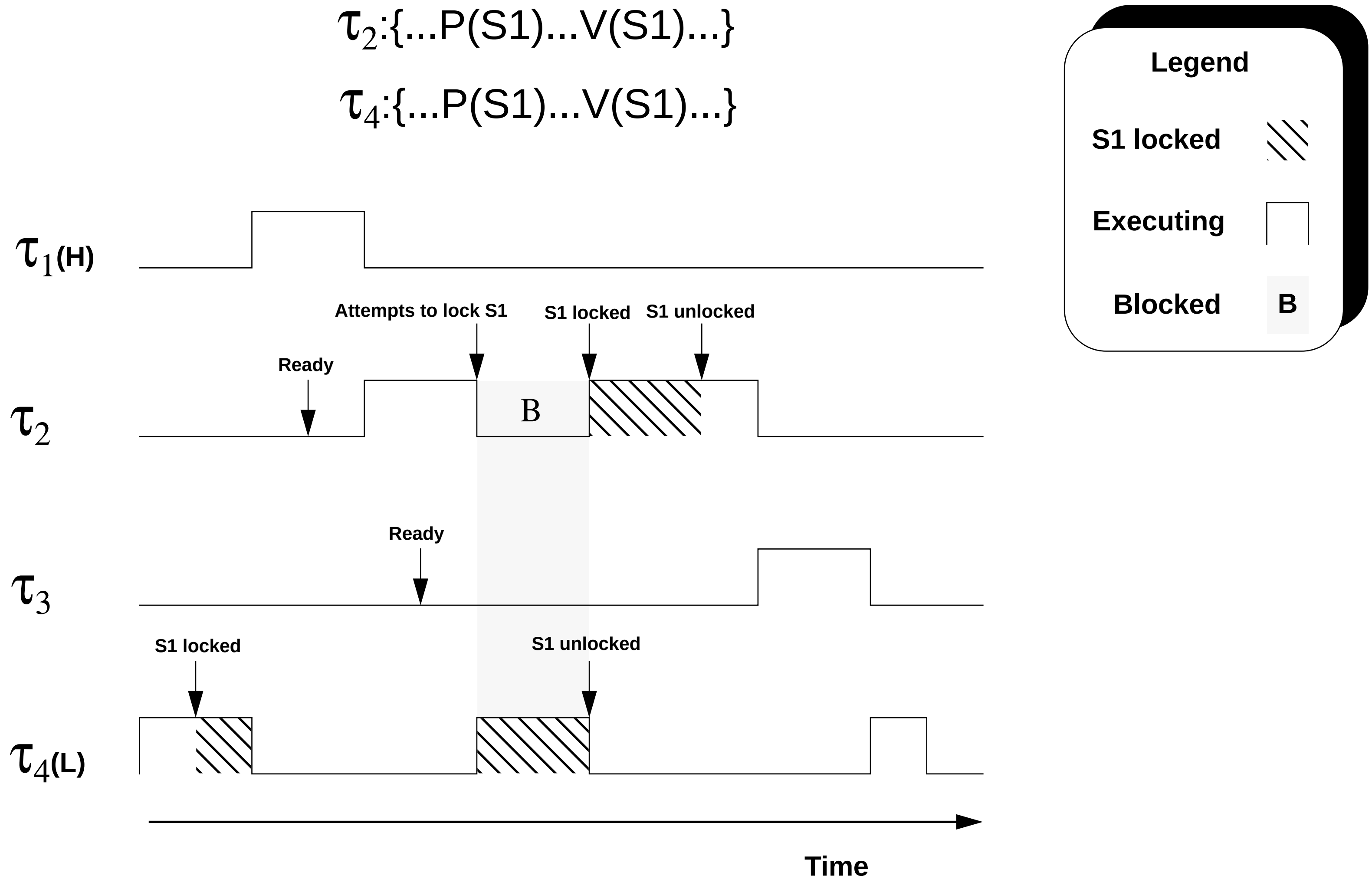




Priority Ceiling Protocol (PCP)

τ_2:{...P(S1)...V(S1)...}

τ_4:{...P(S1)...V(S1)...}

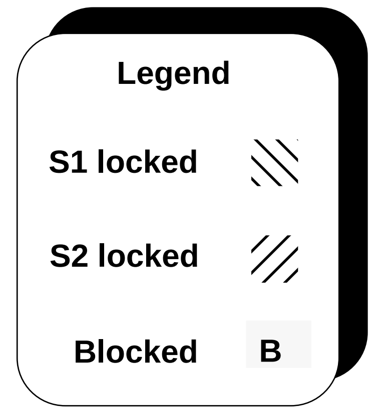


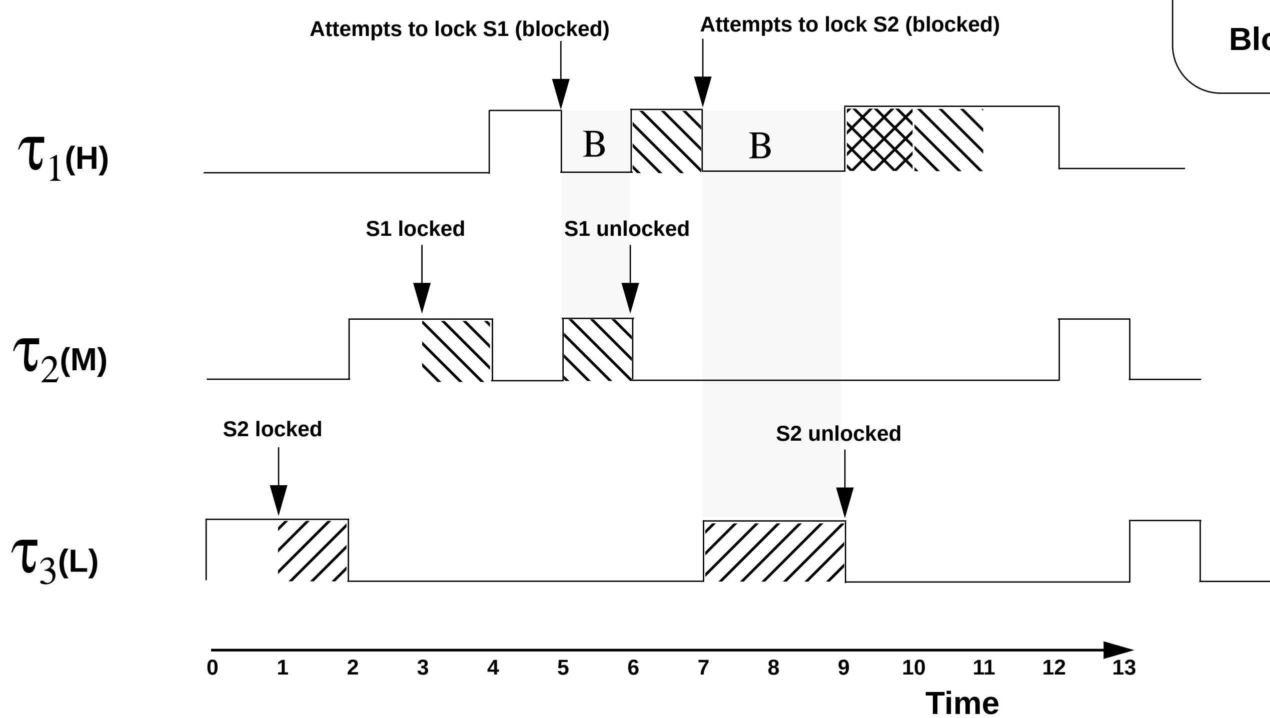
Example Of Chained Blocking (BIP)

τ_1:{...P(S1)...P(S2)...V(S2)...V(S1)...}

τ_2:{...P(S1)...V(S1)...}

τ_3:{...P(S2)...V(S2)...}



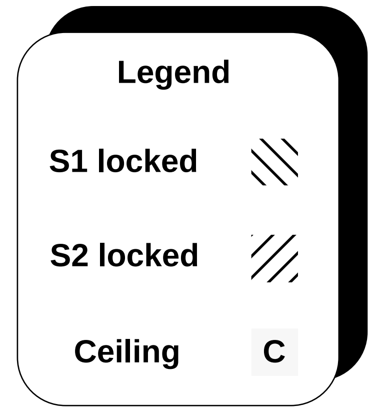


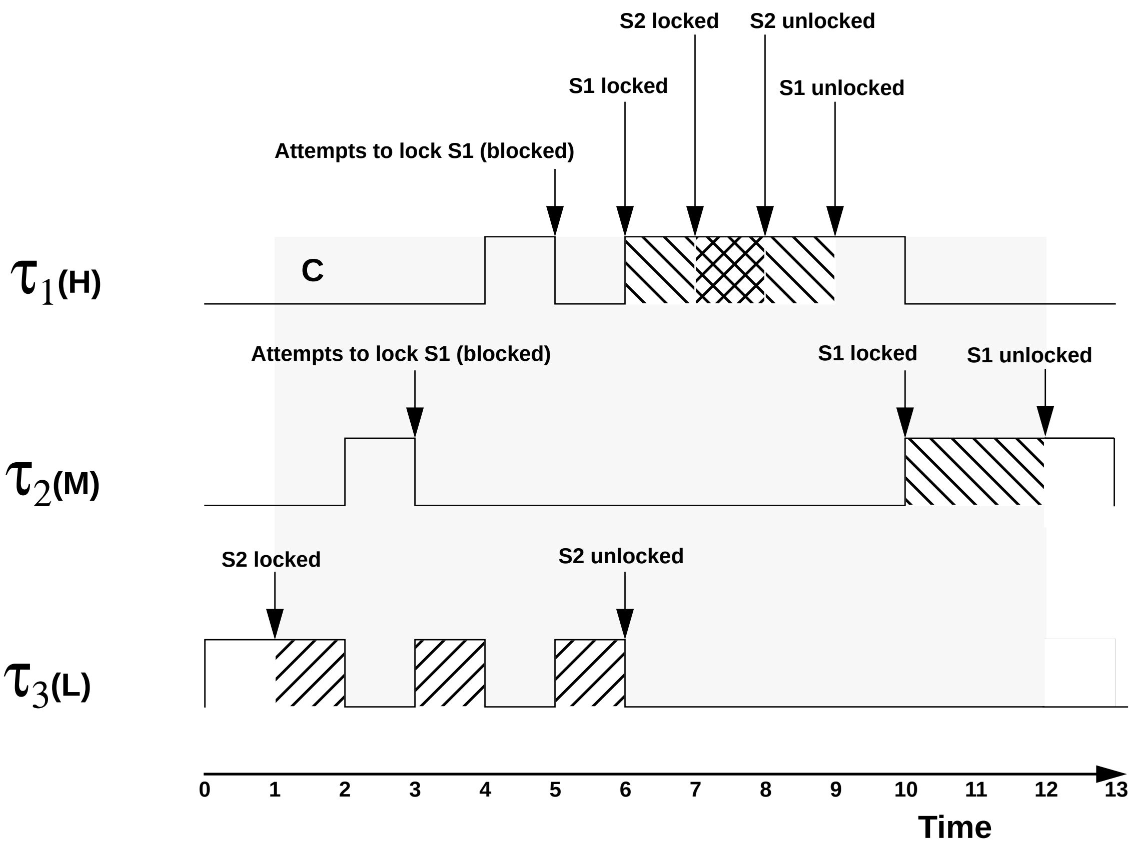
Blocked At Most Once (PCP)

τ_1:{...P(S1)...P(S2)...V(S2)...V(S1)...}

τ_2:{...P(S1)...V(S1)...}

τ_3:{...P(S2)...V(S2)...}

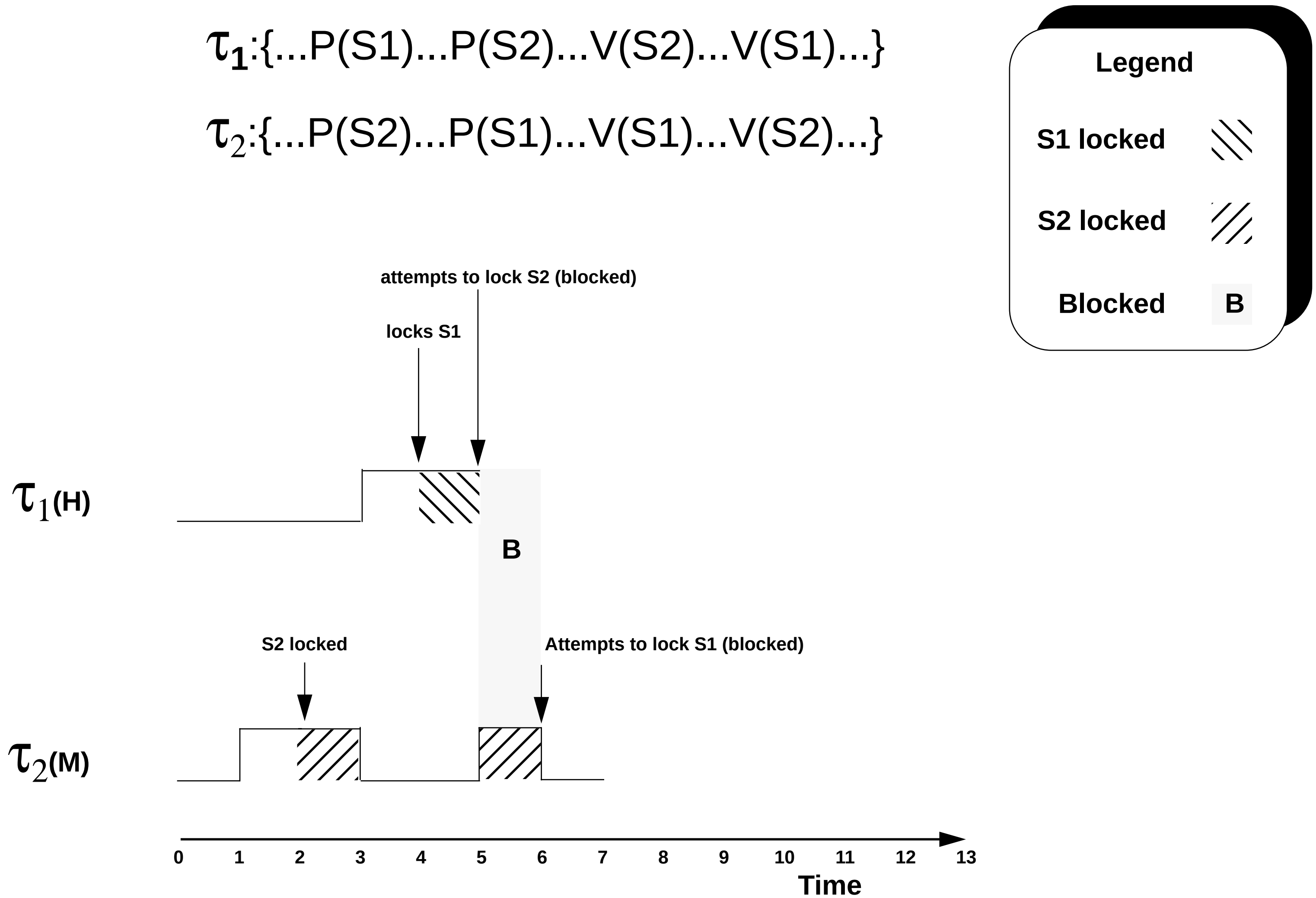




Deadlock: Using BIP

τ_1:{...P(S1)...P(S2)...V(S2)...V(S1)...}

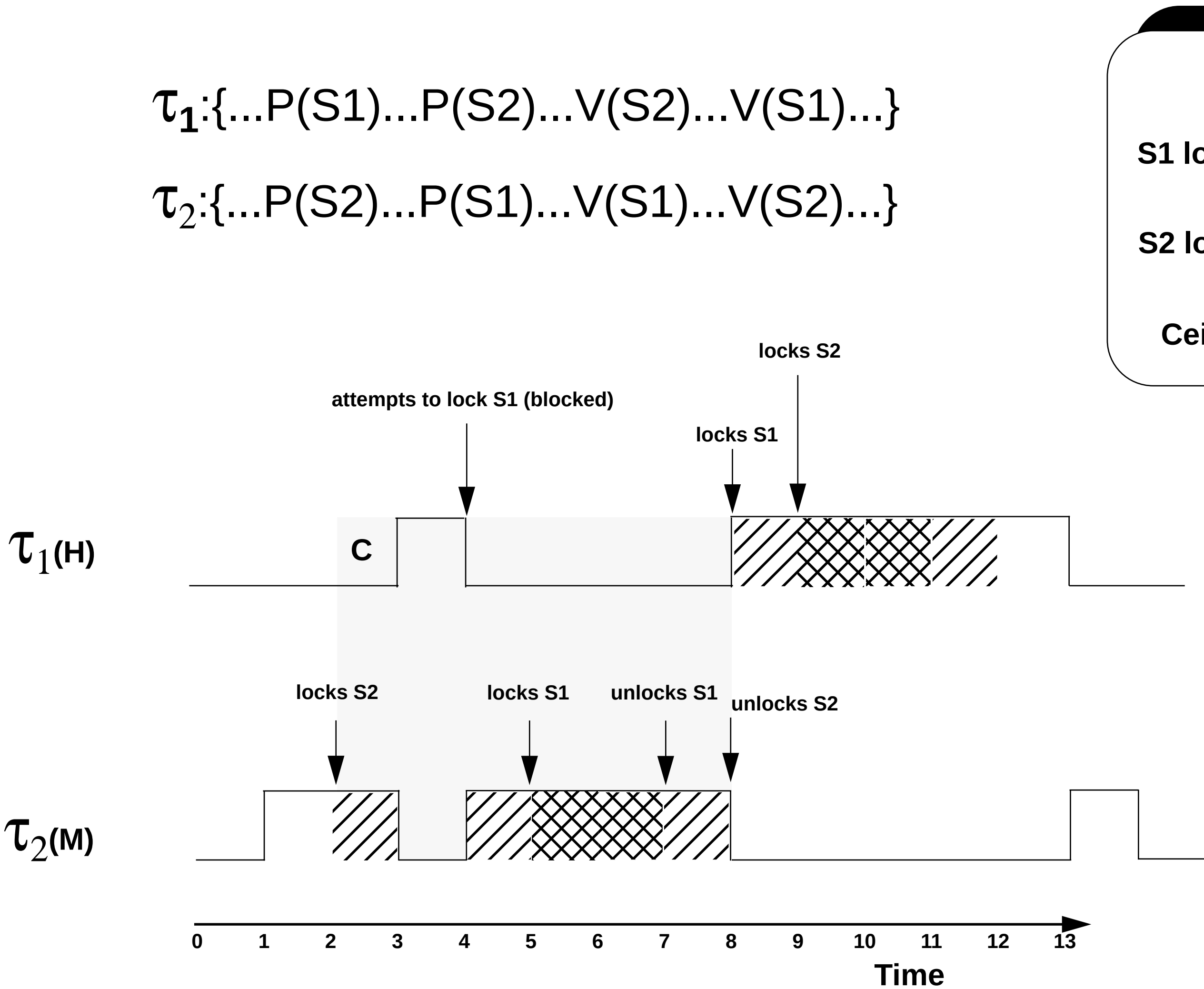
τ_2:{...P(S2)...P(S1)...V(S1)...V(S2)...}



Deadlock Avoidance: Using PCP

τ_1:{...P(S1)...P(S2)...V(S2)...V(S1)...}

τ_2:{...P(S2)...P(S1)...V(S1)...V(S2)...}



Summary of Synchronization Protocols

Protocol	Bounded Priority Inversion	Blocked at Most Once	Deadlock Avoidance
Nonpreemptible critical sections	Yes	Yes[1]	Yes[1]
Highest locker's priority	Yes	Yes[1]	Yes[1]
Basic inheritance	Yes	No	No
Priority ceiling	Yes	Yes[2]	Yes

[1] Only if tasks do not suspend within critical sections

[2] PCP is not affected if tasks suspend within critical sections

Sample Problem with Synchronization

When basic priority inheritance protocol is used:

Task	Period	Execution Time	Priority	Blocking Delays	Deadline
τ_1	100	20	High	20+10	100
τ_2	150	40	Medium	10	130
τ_3	350	100	Low	0	350

UB Test for Sample Problem

This format is sometimes called a schedulability model for the task set:

$$f_1 = \frac{C_1}{T_1} + \frac{B_1}{T_1} = \frac{20}{100} + \frac{30}{100} = 0.500 < U(1)$$

$$f_2 = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{B_2}{T_2} = \frac{20}{100} + \frac{40}{150} + \frac{10}{150} = 0.534 < 0.729$$

$$U(2, .80) = 0.729$$

$$f_3 = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{20}{100} + \frac{40}{150} + \frac{100}{350} = 0.753 < U(3)$$