

<u>Help</u>

selfpoised ~

Next >

<u>Course</u> <u>Progress</u> <u>Dates</u> <u>Discussion</u>

☆ Course / 19. Concurrency and Synchronization / Worked Examples



WE19.1

< Previous</pre>

 \square Bookmark this page

■ Calculator

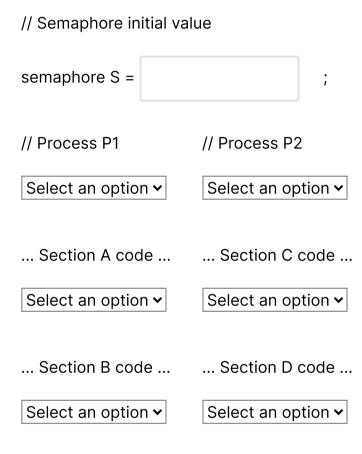
Video explanation of solution is provided below the problem.

Precedence Constraints with Semaphores

28 points possible (ungraded)

P1 and P2 are processes that run concurrently. P1 has two sections of code where section A is followed by section B. Similarly, P2 has two sections: C followed by D. Within each process execution proceeds sequentially, so we are guaranteed that A < B, i.e., A precedes B. Similarly we know that C < D. There is no looping; each process runs exactly once. You will be asked to add semaphores to the programs – you may need to use more than one semaphore. **Please give the initial values of any semaphores you use.** Your solution must use a minimum number of semaphores and it must not introduce any unnecessary precedence constraints.

1. Using a semaphore named S, please specify the initial value of semaphore S, and add wait(S) and signal(S) statements as needed in the spaces below so that the precedence constraint B < C is satisfied, i.e., execution of P1 finishes before execution of P2 begins. For each drop down, select the missing line of code. If no command is missing, select None.



2. Using a semaphore named S, please specify the initial value of semaphore S, and add wait(S) and signal(S) statements as needed in the spaces below so that that D < A or B < C, i.e., executions of P1 and P2 cannot overlap, but are allowed to occur in either order. For each drop down, select the missing line of code. If no command is missing, select None.

// Semaphore initial value	
semaphore S =	;
// Process P1	// Process P2
Select an option 🗸	Select an option 🗸
Section A code	
Select an option 🗸	Select an option >
Section B code	
Select an option 🗸	Select an option ~

3. Using two semaphores named S and T, please specify the initial values of semaphores S and T, and add wait(S), wait(T), signal(S), and signal(T) statements as needed in the spaces below so that A < D and C < B, i.e., the first section (A and C) of **both** processes completes execution before the second section (B or D) of **either** process begins execution.

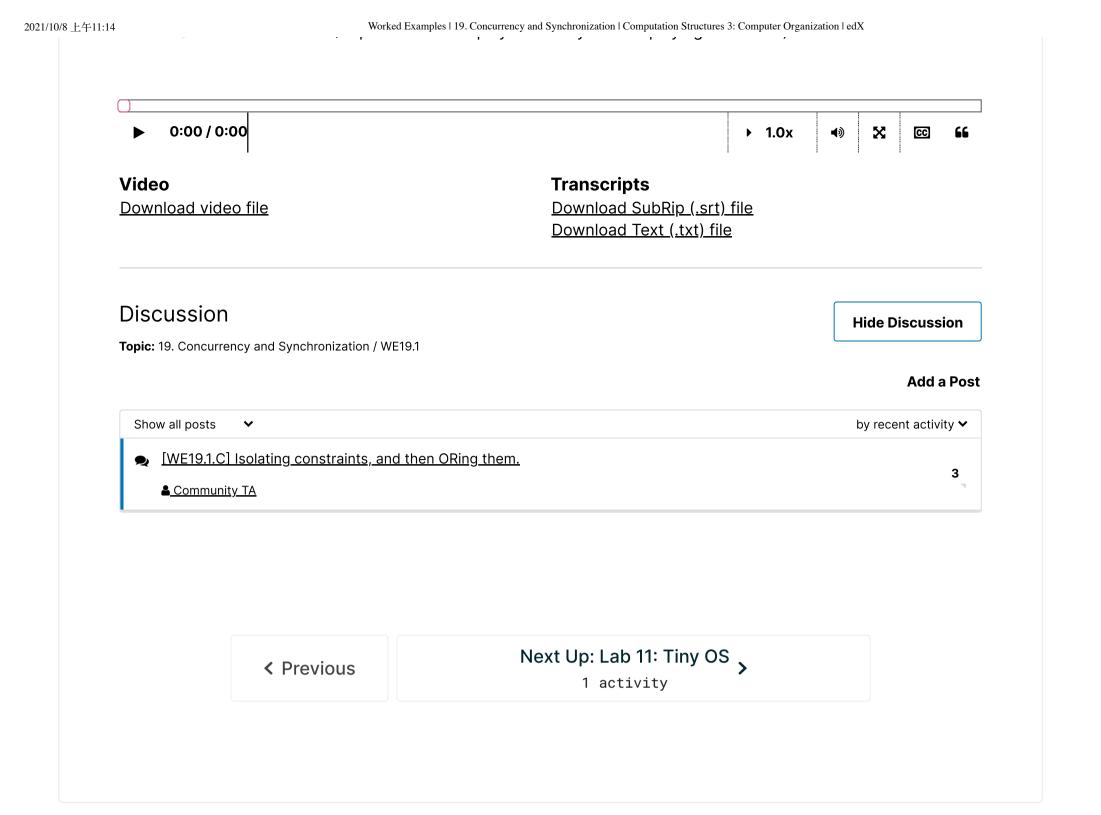
For each drop down, select the missing line of code. If a particular code region only requires one command, then select that command for the first drop down and select None for the second drop down. If no commands are needed in a region then select None for both answers. **Assume process P1 calls signal(S).**

// Semaphore initial values semaphore S = semaphore T = // Process P1 // Process P2 Select an option > Select an option > Select an option ✓ Select an option > ... Section A code Section C code ... Select an option > Select an option > Select an option ➤ Select an option > ... Section B code Section D code ... Select an option > Select an option > Select an option ➤ Select an option >

Submit

Semaphores





© All Rights Reserved



edX

About

Affiliates

edX for Business

Open edX

<u>Careers</u>

<u>News</u>

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

<u>Sitemap</u>

Connect

<u>Blog</u>



Contact Us

Help Center

Media Kit

Donate















© 2021 edX Inc. All rights reserved.

深圳市恒宇博科技有限公司 <u>粤ICP备17044299号-2</u>