HÁSKÓLI ÍSLANDS
Iðnaðarverkfræði-, vélaverkfræði- og tölvunarfræðideild

TÖL401G: Stýrikerfi / Operating Systems · Vormisseri 2022

Assignments 17–18 · To be solved until 22.3.2022, 13:00

Assignment 17

Consider the following two different attempts to solve the dining-philosophers problem described in chapter 6:

First attempt: Each of the five philosopher behaves as follows: Think for a while, then (when hungry) wait until the chopstick to the right is available and grab it, after having grabbed it, wait until the chopstick to the left is available and grab it. After both chopsticks have been grabbed: eat. After finishing eating, release first the left chopstick, then the right chopstick.

Second attempt: Four of the five philosophers are right handers and therefore behave as described in the first attempt. However, one of the five philosophers behaves as a left hander, i.e. has the same behaviour as the right handers, however, left and right are swapped.

For each of the two attempts, prove or disprove whether it may lead to a deadlock or not!
Hint:

- Consider the four necessary conditions for a deadlock and argue (no 100% formal mathematical proof required) for each condition whether it holds or not.
  - When arguing about the four deadlock conditions, some are trivial, still you have to explain why they hold.
  - To argue about circular wait, you may use a resource allocation graphs.
- For proving the presence of a deadlock, it is sufficient to show one scenario where it occurs.
- However for proving the absence, you would in theory have to show the absence each of the possible (3^5=243) scenarios. To avoid that, try to find a philosopher who always breaks the circular wait: show that in all possible cases of having chopsticks no deadlock occurs and then argue that all 243 cases involve on of the possible cases of that philosopher.
- Do not make any assumptions on the speed of the philosophers, i.e. some philosopher may think or eat longer than other philosophers.

Assignment 18

Processes A, B, and C use two different non-sharable, non-preemptable resource types R_1 and R_2; for each resource type, *9 instances* exist in the system. The processes announce their maximum resource usage as follows: A: `max_R1(6)`, A: `max_R2(3)`; B: `max_R1(8)`, B: `max_R2(2)`; C: `max_R1(4)`, C: `max_R2(1)`. (X: `max_R`i`(`n`)` specifies that process X will never request and use more than n resource instances of resource type R_i.)

Initially, all resources are available. Then, the processes make the following resource allocation requests that are successfully granted: A: `alloc_R1(2)`, A: `alloc_R2(1)`; B: `alloc_R1(3)`, B: `alloc_R2(1)`; C: `alloc_R1(2)`, C: `alloc_R2(1)`. (X: `alloc_R`i`(`n`)` specifies that process X requests n resource instances of type R_i with the aim of using them.)

1. Is the resulting state a safe state? Justify your answer by using the Safety algorithm! If the state is safe, provide a safe schedule!

Now (after the above requests have been granted), process A releases one instance of resource type R_1: A: `release_R1(1)` and after that, B requests two further instances of resource type R_1: B: `alloc_R1(2)`.

2. Use the Banker's algorithm to decide whether the system should grant the above request of B for two further resources or not. (Show the values of the involved matrices/vectors.) If the allocation request is not to be granted: what action is appropriate on the scheduler level concerning process B?

Assume that the request of B for two further instances of resource type R_1 has not been granted in the above part 2: Just after not granting this request, process C terminates and releases all resources it is currently holding. The operating systems therefore re-runs the Banker's algorithm to decide whether maybe now the pending request of process B (B: `alloc_R1(2)`) can be granted.

3. Use the Banker's algorithm to decide whether the system should now grant the above request of B for two further resources or not. (Show the values of the involved matrices/vectors.) What action is now appropriate on the scheduler level concerning process B?

Preparation

Read end of chapter 6 and chapter 7 (and watch videos as necessary) we will work on assignments 17–18 which will be about deadlocks.