

HW4 Report

B10505047 邱郁喆

Q7.3 :

The CPU scheduler determines the execution order of the two threads. If it allows **do_work_one** to acquire **first_mutex** and **do_work_two** to acquire **second_mutex** before either thread can acquire the other mutex, then a deadlock will occur since both threads are blocked waiting for each other.

Q7.6 :

- a. Increase **Available** (new resources added): If the system was in safe state before, adding new resources means more resources are free for allocation, making the system remain in the safe state and become safer.
- b. Decrease **Available** (resource permanently removed from system): Removing resources may cause the system to go into unsafe state.
- c. Increase **Max** for one process (the process needs or wants more resources than allowed): Increasing Max for one process means the process may require more resources to complete, so it may cause the system to go into unsafe state.
- d. Decrease **Max** for one process (the process decides it does not need that many resources): If the system was in safe state before, decreasing Max for one process means its demand for resources decreases, making the system remain in the safe state and become safer.
- e. Increase the number of processes: Increasing the number of processes will increase the demand for resources and may cause the system to go into unsafe state.
- f. Decrease the number of processes: If the system was in safe state before, decreasing the number of processes will decrease the demand for resources, making the system remain in the safe state and become safer.

Q7.12 :

a. Available = (0, 3, 0, 1)

Available = (0, 3, 0, 1)					Available = (3, 4, 2, 2)					Available = (5, 6, 3, 2)						
Need					Need					Need						
	A	B	C	D		A	B	C	D		A	B	C	D		
P ₀	2	1	0	3	P ₂ finish →	P ₀	2	1	0	3	P ₁ finish →	P ₀	2	1	0	3
P ₁	1	0	0	1		P ₁	1	0	0	1		P ₁	0	0	0	0
P ₂	0	2	0	0		P ₂	0	0	0	0		P ₂	0	0	0	0
P ₃	4	1	0	2		P ₃	4	1	0	2		P ₃	4	1	0	2
P ₄	2	1	1	3		P ₄	2	1	1	3		P ₄	2	1	1	3
Available = (5, 13, 9, 2)					P ₃ finish →					Need						
	A	B	C	D	can't finish P ₀ or P ₄						A	B	C	D		
P ₀	2	1	0	3	∴ unsafe					P ₀	2	1	0	3		
P ₁	0	0	0	0						P ₁	0	0	0	0		
P ₂	0	0	0	0						P ₂	0	0	0	0		
P ₃	0	0	0	0						P ₃	0	0	0	0		
P ₄	2	1	1	3						P ₄	2	1	1	3		

b. Available = (1, 0, 0, 2)

Available = (1, 0, 0, 2)	Available = (3, 2, 1, 2)	Available = (6, 3, 3, 2)
--------------------------	--------------------------	--------------------------

Need					P ₁ finish →	Need					P ₂ finish →	Need				
A	B	C	D	A		B	C	D	A	B		C	D			
P ₀	2	1	0	3	P ₀	2	1	0	3	P ₀	2	1	0	3		
P ₁	1	0	0	1	P ₁	0	0	0	0	P ₁	0	0	0	0		
P ₂	0	2	0	0	P ₂	0	2	0	0	P ₂	0	0	0	0		
P ₃	4	1	0	2	P ₃	4	1	0	2	P ₃	4	1	0	2		
P ₄	2	1	1	3	P ₄	2	1	1	3	P ₄	2	1	1	3		

Available = (9, 3, 4, 7)	Available = (9, 8, 5, 7)	
--------------------------	--------------------------	--

Need					P ₃ finish →	Need					P ₄ finish →	Need				
A	B	C	D	A		B	C	D	A	B		C	D			
P ₀	0	0	0	0	P ₀	0	0	0	0	P ₀	0	0	0	0		
P ₁	0	0	0	0	P ₁	0	0	0	0	P ₁	0	0	0	0		
P ₂	0	0	0	0	P ₂	0	0	0	0	P ₂	0	0	0	0		
P ₃	4	1	0	2	P ₃	0	0	0	0	P ₃	0	0	0	0		
P ₄	2	1	1	3	P ₄	2	1	1	3	P ₄	0	0	0	0		

done ⇒ complete order

P₁, P₂, P₀, P₃, P₄