

Date: 2022-11-22

Group MHA

Murtadha Alobaidi , mhaao@kth.se

Hasan Alzubeidi , haz@kth.se

Ahmed Alkhalaf, ahmedalk@kth.se

1) What is the meaning of the term busy waiting? Can busy waiting be avoided altogether? Explain your answer.

**Answer:**

**Busy waiting** it means the process is waiting to receive a condition to be satisfied to the loop without sacrificing the processor. To avoid **busy waiting**, puts waiting process to **sleep** and when the corresponding program state is reached **wakeup** the busy waiting. Putting the process to **sleep** and **wakeup** later can make the process overhead associated.

2) Show that, if the wait() and signal() semaphore operations are not executed atomically, then mutual exclusion may be violated.

**Answer:**

**wait()** the operations atomically decrease the value with **Semaphores**. If two **wait()** operations executed in the **semaphore**. When the first operations have value 1 the operations not performed atomically. The no other processes can be executing and both operations decrement the semaphore value.

3) Consider the following snapshot of a system, and answer the following questions using the banker's algorithm:

	<u>Allocation</u>	<u>Max</u>	<u>Available</u>
	<u>A B C D</u>	<u>A B C D</u>	<u>A B C D</u>
$T_0$	0 0 1 2	0 0 1 2	1 5 2 0
$T_1$	1 0 0 0	1 7 5 0	
$T_2$	1 3 5 4	2 3 5 6	
$T_3$	0 6 3 2	0 6 5 2	
$T_4$	0 0 1 4	0 6 5 6	

- What is the content of the matrix Need?
- Is the system in a safe state?
- If a request from thread  $T_1$  arrives for (0, 4, 2, 0), can the request be granted immediately?

**Answer:**

a. The content of the matrix **Need** for the processes 0 through processes 4 are :

P0 (0,0,0,0)

P1 (0,7,5,0)

P2 (1,0,0,2)

P3 (0,0,2,0)

P4 (0,6,4,2)

b. The system in a **safe state**:

	Allocation				Max				Available				Need				
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	
P <sub>0</sub>	0	0	1	2	0	0	1	2	1	5	2	0	0	0	0	0	✓
P <sub>1</sub>	1	0	0	0	1	7	5	0	1	5	3	2	0	7	5	0	X
P <sub>2</sub>	1	3	5	4	2	3	5	6	2	8	9	6	1	0	0	2	✓
P <sub>3</sub>	0	6	3	2	0	6	5	2	2	14	11	8	0	0	2	0	✓
P <sub>4</sub>	0	0	1	4	0	6	5	6	2	14	12	12	0	6	4	2	✓
<div>3 14 12 12</div> <div>total number</div> <div>of resources</div>																	

We have the **total available resources**: 3 14 12 12

The **safe sequence** we have : p<sub>0</sub>, p<sub>2</sub>, p<sub>3</sub>, p<sub>4</sub> and p<sub>1</sub>

**The system is in the safe state**, we can see after executing all the processes the resource available is equal to the **total available resources**.

**The answer is YES** the system in a safe state.

c. The request can be granted immediately.

The value of Available: (1, 1, 0, 0)

The processes ordering: p<sub>0</sub>, p<sub>2</sub>, p<sub>3</sub>, p<sub>1</sub> and p<sub>4</sub>

4) Consider the following resource-allocation policy. Requests for and releases of resources are allowed at any time. If a request for resources cannot be satisfied because the resources are not available, then we check any threads that are blocked waiting for resources. If a blocked thread has the desired resources, then these resources are taken away from it and are given to the requesting thread. The vector of resources for which the blocked thread is waiting is increased to include the resources that were taken away.

For example, a system has three resource types, and the vector Available is initialized to (4, 2, 2). If thread T0 asks for (2, 2, 1), it gets them. If T1 asks for (1, 0, 1), it gets them. Then, if T0 asks for (0, 0, 1), it is blocked (resource not available). If T2 now asks for (2, 0, 0), it gets the available one (1, 0, 0), as well as one that was allocated to T0 (since T0 is blocked). T0 Allocation vector goes down to (1, 2, 1), and its Need vector goes up to (1, 0, 1).

- a. Can deadlock occur? If you answer “yes”, give an example. If you answer “no”, specify which necessary condition cannot occur.
- b. Can deadlock occur? If you answer "yes", give an example. If you answer "no", specify which necessary condition cannot occur.

**Answer:**

- a. NO, Deadlock can't occur because preemption exists.
- b. Yes, the process maybe never build all the resource he needed if preemption exists.  
Example process C.

**SOURCES:**

(ABRAHAM SILBERSCHATZ, 2018)