

Assignment-7

Problem-01:

A single processor system has three resource types X, Y and Z, which are shared by three processes. There are 5 units of each resource type. Consider the following scenario, where the column alloc denotes the number of units of each resource type allocated to each process, and the column request denotes the number of units of each resource type requested by a process in order to complete execution. Which of these processes will finish LAST?

1. P0
2. P1
3. P2
4. None of the above since the system is in a deadlock

| | Alloc | | | Request | | |
|----|-------|---|---|---------|---|---|
| | X | Y | Z | X | Y | Z |
| P0 | 1 | 2 | 1 | 1 | 0 | 3 |
| P1 | 2 | 0 | 1 | 0 | 1 | 2 |
| P2 | 2 | 2 | 1 | 1 | 2 | 0 |

Solution-

According to question-

- Total = [X Y Z] = [5 5 5]
- Total _Alloc = [X Y Z] = [5 4 3]

Now,

Available

$$= \text{Total} - \text{Total_Alloc}$$

$$= [5\ 5\ 5] - [5\ 4\ 3]$$

$$= [0\ 1\ 2]$$

Step-01:

- With the instances available currently, only the requirement of the process P1 can be satisfied.
- So, process P1 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then,

Available

$$= [0\ 1\ 2] + [2\ 0\ 1]$$

$$= [2\ 1\ 3]$$

Step-02:

- With the instances available currently, only the requirement of the process P0 can be satisfied.
- So, process P0 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [2\ 1\ 3] + [1\ 2\ 1]$$

$$= [3\ 3\ 4]$$

Step-03:

- With the instances available currently, the requirement of the process P2 can be satisfied.
- So, process P2 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [3 \ 3 \ 4] + [2 \ 2 \ 1]$$

$$= [5 \ 5 \ 5]$$

Thus,

- There exists a safe sequence P1, P0, P2 in which all the processes can be executed.
- So, the system is in a safe state.
- Process P2 will be executed at last.

Thus, Option (C) is correct.

Problem-02:

An operating system uses the banker's algorithm for deadlock avoidance when managing the allocation of three resource types X, Y and Z to three processes P0, P1 and P2. The table given below presents the current system state. Here, the Allocation matrix shows the current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.

| | Allocation | | | Max | | |
|--|------------|---|---|-----|---|---|
| | X | Y | Z | X | Y | Z |

| | | | | | | |
|-----------|---|---|---|---|---|---|
| P0 | 0 | 0 | 1 | 8 | 4 | 3 |
| P1 | 3 | 2 | 0 | 6 | 2 | 0 |
| P2 | 2 | 1 | 1 | 3 | 3 | 3 |

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available. The system is currently in safe state. Consider the following independent requests for additional resources in the current state-

REQ1: P0 requests 0 units of X, 0 units of Y and 2 units of Z

REQ2: P1 requests 2 units of X, 0 units of Y and 0 units of Z

Which of the following is TRUE?

1. Only REQ1 can be permitted
2. Only REQ2 can be permitted
3. Both REQ1 and REQ2 can be permitted
4. Neither REQ1 nor REQ2 can be permitted

Solution-

According to question,

$$\text{Available} = [X \ Y \ Z] = [3 \ 2 \ 2]$$

Now,

$$\text{Need} = \text{Max} - \text{Allocation}$$

So, we have-

| | Allocation | Max | Need |
|--|-------------------|------------|-------------|
|--|-------------------|------------|-------------|

| | X | Y | Z | X | Y | Z | X | Y | Z |
|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| P0 | 0 | 0 | 1 | 8 | 4 | 3 | 8 | 4 | 2 |
| P1 | 3 | 2 | 0 | 6 | 2 | 0 | 3 | 0 | 0 |
| P2 | 2 | 1 | 1 | 3 | 3 | 3 | 1 | 2 | 2 |

Currently, the system is in safe state.

(It is given in question. If we want, we can check)

Checking Whether REQ1 Can Be Entertained-

- Need of P0 = [0 0 2]
- Available = [3 2 2]

Clearly,

- With the instances available currently, the requirement of REQ1 can be satisfied.
- So, banker's algorithm assumes that the request REQ1 is entertained.
- It then modifies its data structures as-

| | Allocation | | | Max | | | Need | | |
|-----------|-------------------|----------|----------|------------|----------|----------|-------------|----------|----------|
| | X | Y | Z | X | Y | Z | X | Y | Z |
| P0 | 0 | 0 | 3 | 8 | 4 | 3 | 8 | 4 | 0 |
| P1 | 3 | 2 | 0 | 6 | 2 | 0 | 3 | 0 | 0 |
| P2 | 2 | 1 | 1 | 3 | 3 | 3 | 1 | 2 | 2 |

Available

$$= [3 \ 2 \ 2] - [0 \ 0 \ 2]$$

$$= [3 \ 2 \ 0]$$

- Now, it follows the safety algorithm to check whether this resulting state is a safe state or not.
- If it is a safe state, then REQ1 can be permitted otherwise not.

Step-01:

- With the instances available currently, only the requirement of the process P1 can be satisfied.
- So, process P1 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [3 \ 2 \ 0] + [3 \ 2 \ 0]$$

$$= [6 \ 4 \ 0]$$

Now,

- It is not possible to entertain any process.
- The system has entered the deadlock state which is an unsafe state.
- Thus, REQ1 will not be permitted.

Checking Whether REQ2 Can Be Entertained-

- Need of P1 = $[2 \ 0 \ 0]$
- Available = $[3 \ 2 \ 2]$

Clearly,

- With the instances available currently, the requirement of REQ1 can be satisfied.
- So, banker's algorithm assumes the request REQ2 is entertained.
- It then modifies its data structures as-

| | Allocation | | | Max | | | Need | | |
|-----------|------------|---|---|-----|---|---|----------|---|---|
| | X | Y | Z | X | Y | Z | X | Y | Z |
| P0 | 0 | 0 | 1 | 8 | 4 | 3 | 8 | 4 | 2 |
| P1 | 5 | 2 | 0 | 6 | 2 | 0 | 1 | 0 | 0 |
| P2 | 2 | 1 | 1 | 3 | 3 | 3 | 1 | 2 | 2 |

Available

$$= [3\ 2\ 2] - [2\ 0\ 0]$$

$$= [1\ 2\ 2]$$

- Now, it follows the safety algorithm to check whether this resulting state is a safe state or not.
- If it is a safe state, then REQ2 can be permitted otherwise not.

Step-01:

- With the instances available currently, only the requirement of the process P1 can be satisfied.
- So, process P1 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [1\ 2\ 2] + [5\ 2\ 0]$$

$$= [6\ 4\ 2]$$

Step-02:

- With the instances available currently, only the requirement of the process P2 can be satisfied.
- So, process P2 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [6\ 4\ 2] + [2\ 1\ 1]$$

$$= [8\ 5\ 3]$$

Step-03:

- With the instances available currently, the requirement of the process P0 can be satisfied.
- So, process P0 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [8\ 5\ 3] + [0\ 0\ 1]$$

$$= [8\ 5\ 4]$$

Thus,

- There exists a safe sequence P1, P2, P0 in which all the processes can be executed.
- So, the system is in a safe state.
- Thus, REQ2 can be permitted.

Thus, Correct Option is (B).

Problem-03:

A system has 4 processes and 5 allocatable resource. The current allocation and maximum needs are as follows-

| | Allocated | | | | | Maximum | | | | |
|----------|-----------|---|---|---|---|---------|---|---|---|---|
| A | 1 | 0 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 3 |
| B | 2 | 0 | 1 | 1 | 0 | 2 | 2 | 2 | 1 | 0 |
| C | 1 | 1 | 0 | 1 | 1 | 2 | 1 | 3 | 1 | 1 |
| D | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 2 | 2 | 0 |

If Available = [0 0 X 1 1], what is the smallest value of x for which this is a safe state?

Solution-

Let us calculate the additional instances of each resource type needed by each process.

We know,

$$\text{Need} = \text{Maximum} - \text{Allocation}$$

So, we have-

| | Need | | | | |
|---|------|---|---|---|---|
| A | 0 | 1 | 0 | 0 | 2 |
| B | 0 | 2 | 1 | 0 | 0 |
| C | 1 | 0 | 3 | 0 | 0 |
| D | 0 | 0 | 1 | 1 | 0 |

Case-01: For X = 0

If X = 0, then-

Available

= [0 0 0 1 1]

- With the instances available currently, the requirement of any process can not be satisfied.
- So, for X = 0, system remains in a deadlock which is an unsafe state.

Case-02: For X = 1

If X = 1, then-

Available

= [0 0 1 1 1]

Step-01:

- With the instances available currently, only the requirement of the process D can be satisfied.
- So, process D is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [0\ 0\ 1\ 1\ 1] + [1\ 1\ 1\ 1\ 0]$$

$$= [1\ 1\ 2\ 2\ 1]$$

- With the instances available currently, the requirement of any process can not be satisfied.
- So, for $X = 1$, system remains in a deadlock which is an unsafe state.

Case-02: For $X = 2$

If $X = 2$, then-

Available

$$= [0\ 0\ 2\ 1\ 1]$$

Step-01:

- With the instances available currently, only the requirement of the process D can be satisfied.
- So, process D is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [0\ 0\ 2\ 1\ 1] + [1\ 1\ 1\ 1\ 0]$$

$$= [1\ 1\ 3\ 2\ 1]$$

Step-02:

- With the instances available currently, only the requirement of the process C can be satisfied.
- So, process C is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

$$= [1\ 1\ 3\ 2\ 1] + [1\ 1\ 0\ 1\ 1]$$

$$= [2\ 2\ 3\ 3\ 2]$$

Step-03:

- With the instances available currently, the requirement of both the processes A and B can be satisfied.
- So, processes A and B are allocated the requested resources one by one.
- They complete their execution and then free up the instances of resources held by it.

Then-

Available

$$= [2\ 2\ 3\ 3\ 2] + [1\ 0\ 2\ 1\ 1] + [2\ 0\ 1\ 1\ 0]$$

$$= [5\ 2\ 6\ 5\ 3]$$

Thus,

- There exists a safe sequence in which all the processes can be executed.
- So, the system is in a safe state.
- Thus, minimum value of X that ensures system is in safe state = 2.