Fall 2023 Set -A

2 b) Suppose, in a workspace, we have a set of resource types, R = {R1, R2, R3, R4} and a set of processes, P = {P1, P2, P3 P4}. R1, R2, R3, and R4 have 2, 2, 2, and 2 instances respectively.

* P1 is holding 1 instance of R4
* P2 is holding 1 instance of R1
* P3 is holding 1 instance of R1
* P4 is holding 1 instance of R4
* P4 is holding 1 instance of R2
* P2 requests 1 instance of R3
* P2 is holding 1 instance of R2
* P1 requests 1 instance of R1
* P3 is holding 1 instance of R3
* P4 is holding 1 instance of R3
* P3 requests 1 instance of R4
* P4 requests 2 instances of R1

Construct a resource allocation graph for the above scenario and identify the cycle (if any) and decide whether there is deadlock or not. [4]

###################################

Fall 2023 Set -B

2 b) Suppose, in a workspace, we have a set of resource types, R = {R1, R2, R3, R4} and a set of processes, P = {P1, P2, P3 P4}. R1, R2, R3, and R4 have 2, 3, 2, and 3 instances respectively.

* P1 is holding 1 instance of R2
* P2 is holding 2 instances of R2
* P3 is holding 1 instance of R1
* P3 requests 1 instance of R2
* P4 is holding 2 instances of R4
* P2 holding 1 instance of R1
* P1 requests 1 instance of R1
* P4 requests 1 instance of R3
* P3 is holding 1 instance of R3
* P1 is holding 1 instance of R3
* P3 is holding 1 instance of R4

Construct a resource allocation graph for the above scenario and identify the cycle (if any) and decide whether there is deadlock or not. [4]

##################################

Fall 2023 Set -C

2 b) Suppose, in a workspace, we have a set of resource types, R = {R1, R2, R3, R4} and a set of processes, P = {P1, P2, P3 P4, P5}. R1, R2, R3, and R4 have 3, 2, 4, and 2 instances respectively.

* P1 is holding 2 instances of R1
* P2 is holding 1 instance of R3
* P3 is holding 1 instance of R4
* P5 requests 2 instances of R3
* P4 is holding 1 instance of R4
* P3 requests 1 instance of R2
* P2 requests 1 instance of R1
* P2 is holding 1 instance of R2
* P1 is requesting 1 instance of R4
* P3 is holding 1 instance of R3
* P4 is holding 1 instance of R3
* P5 holding 1 instance of R2

Construct a resource allocation graph for the above scenario and identify the cycle (if any) and decide whether there is deadlock or not. [4]

############################

Summer 2023 set-A

2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a)** Consider the following snapshot of a system:   |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | **Allocation** | | |  | **Max** | | |  | **Available** | | | |  |  | **A** | **B** | **C** |  | **A** | **B** | **C** |  | **A** | **B** | **C** | | **P0** |  | 5 | 2 | 2 |  | 9 | 9 | 8 |  | 7 | 10 | 5 | | **P1** |  | 3 | 2 | 0 |  | 9 | 9 | 10 |  |  |  |  | | **P2** |  | 5 | 3 | 3 |  | 6 | 8 | 5 |  |  |  |  | | **P3** |  | 3 | 0 | 0 |  | 6 | 7 | 9 |  |  |  |  | | **P4** |  | 4 | 2 | 1 |  | 5 | 7 | 6 |  |  |  |  | |
| 1. **Calculate** the Need Matrix. Is this system in a **safe state**? If yes, then find the safe sequence using Banker’s Safety algorithm otherwise, provide the necessary explanation. [1 + 3] |
| 1. What happens if the process **P4** requests at this moment for **(0, 3, 1)?** Whether Banker’s algorithm grants the request or not? [5]   ##########################  Spring 2022 set-A  2 d)   1. Consider the following snapshot of a system:  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | **Allocation** | | | |  | **Max** | | | | |  |  | **A** | **B** | **C** | **D** |  | **A** | **B** | **C** | **D** | | **P1** |  | 0 | 0 | 1 | 2 |  | 0 | 0 | 2 | 3 | | **P2** |  | 1 | 0 | 0 | 0 |  | 1 | 2 | 2 | 0 | | **P3** |  | 1 | 3 | 5 | 4 |  | 2 | 3 | 5 | 6 | | **P4** |  | 0 | 0 | 0 | 1 |  | 2 | 2 | 0 | 1 | |  |  |  |  |  |  |  |  |  |  |  | |  |  | **Available** | | | |  |  |  |  |  | |  |  | 1 | 2 | 2 | 0 |  |  |  |  |  |   i. Is the system in a safe state? [3]  ii. Can P3’s request (1 0 0 0) be safely granted immediately? [2]  iii. If P3’s request is granted immediately, does the system enter a deadlock? [3] |