**Student name:** Federico Watkins

**CSIS 3810:** Operating Systems

**Chapter:** 7 Deadlocks

**Assignment:** 5

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 🡪 7.11 | 1. 🡪 7.22 | 1. 🡪 7.23 | 1. 🡪 7.25 |

1. Consider the traffic deadlock depicted in Figure 7.10.
2. Show that the four necessary conditions for deadlock indeed hold in this example.
3. None of the cars have the ability to be preempted or removed from their position in traffic. That shows the ***no preemption*** condition.
4. In order to move forward in traffic each car has to wait. In doing so, each car is holding its spot in traffic in a hold and wait fashion. That shows the ***hold-and-wait*** condition.
5. Naturally each car can only take up one physical spot in traffic. That shows the ***mutual exclusion*** condition.
6. Each vehicle in this scenario is depending on the vehicle in front of them to move in order for them to advance. That shows the ***circular wait*** condition
7. State a simple rule for avoiding deadlocks in this system.

A simple rule for avoiding deadlocks in this system would be to prohibit drivers from entering the intersection space unless it is perfectly obvious that the vehicle will get through the intersection right away.

1. Consider the following snapshot of a system:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | **A** | **B** | **C** | **D** |  | **A** | **B** | **C** | **D** |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |
| P1 |  | 2 | 2 | 1 | 0 |  | 3 | 2 | 1 | 1 |
| P2 |  | 3 | 1 | 2 | 1 |  | 3 | 3 | 2 | 1 |
| P3 |  | 0 | 5 | 1 | 0 |  | 4 | 6 | 1 | 2 |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |

Using the banker’s algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 0 | 3 | 0 | 1 |
| P1 |  | 2 | 2 | 1 | 0 |  | 3 | 2 | 1 | 1 |  | 1 | 0 | 0 | 1 |  |  |  |  |  |
| P2 |  | 3 | 1 | 2 | 1 |  | 3 | 3 | 2 | 1 |  | 0 | 2 | 0 | 0 |  |  |  |  |  |
| P3 |  | 0 | 5 | 1 | 0 |  | 4 | 6 | 1 | 2 |  | 4 | 1 | 0 | 2 |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 3 | 4 | 2 | 2 |
| P1 |  | 2 | 2 | 1 | 0 |  | 3 | 2 | 1 | 1 |  | 1 | 0 | 0 | 1 |  |  |  |  |  |
| P2 | Finished during **first** step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | 0 | 5 | 1 | 0 |  | 4 | 6 | 1 | 2 |  | 4 | 1 | 0 | 2 |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 5 | 6 | 3 | 2 |
| P1 | Finished during **second** Step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 | Finished during **first** step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | 0 | 5 | 1 | 0 |  | 4 | 6 | 1 | 2 |  | 4 | 1 | 0 | 2 |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 5 | 11 | 4 | 2 |
| P1 | Finished during **second** Step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 | Finished during **first** step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 | Finished during **third** step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 5 | 11 | 4 | 2 |
| P1 | Finished during **second** Step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 | Finished during **first** step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 | Finished during **third** step | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

At this point P0 and P4 both are greater than the **Available** matrix. Thus, neither will be completed. Hence, the state is **unsafe**.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 1 | 0 | 0 | 2 |
| P1 |  | 2 | 2 | 1 | 0 |  | 3 | 2 | 1 | 1 |  | 1 | 0 | 0 | 1 |  |  |  |  |  |
| P2 |  | 3 | 1 | 2 | 1 |  | 3 | 3 | 2 | 1 |  | 0 | 2 | 0 | 0 |  |  |  |  |  |
| P3 |  | 0 | 5 | 1 | 0 |  | 4 | 6 | 1 | 2 |  | 4 | 1 | 0 | 2 |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 3 | 2 | 1 | 2 |
| P1 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | 3 | 1 | 2 | 1 |  | 3 | 3 | 2 | 1 |  | 0 | 2 | 0 | 0 |  |  |  |  |  |
| P3 |  | 0 | 5 | 1 | 0 |  | 4 | 6 | 1 | 2 |  | 4 | 1 | 0 | 2 |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 6 | 3 | 3 | 3 |
| P1 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | 0 | 5 | 1 | 0 |  | 4 | 6 | 1 | 2 |  | 4 | 1 | 0 | 2 |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 3 | 0 | 1 | 4 |  | 5 | 1 | 1 | 7 |  | 2 | 1 | 0 | 3 |  | 6 | 8 | 4 | 3 |
| P1 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | 4 | 2 | 1 | 2 |  | 6 | 3 | 2 | 5 |  | 2 | 1 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **fifth** step | | | | | | | | |  |  |  |  |  |  | 13 | 10 | 6 | 9 |
| P1 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | Finished during **fourth** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |

At this point the processes have all completed in the following order:

P1, P2, P3, P4, P0. Hence, the state is **safe.**

1. Consider the following snapshot of a system:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **A** | **B** | **C** | **D** |  | **A** | **B** | **C** | **D** |  | **A** | **B** | **C** | **D** |
| P0 |  | 2 | 0 | 0 | 1 |  | 4 | 2 | 1 | 2 |  | 3 | 3 | 2 | 1 |
| P1 |  | 3 | 1 | 2 | 1 |  | 5 | 2 | 5 | 2 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  |  |  |  |  |

Answer the following questions using the baker’s algorithm:

1. Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 2 | 0 | 0 | 1 |  | 4 | 2 | 1 | 2 |  | 2 | 2 | 1 | 1 |  | 3 | 3 | 2 | 1 |
| P1 |  | 3 | 1 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 2 | 1 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  | 0 | 1 | 1 | 2 |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 3 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 5 | 3 | 2 | 2 |
| P1 |  | 3 | 1 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 2 | 1 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  | 0 | 1 | 1 | 2 |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 3 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 6 | 6 | 3 | 4 |
| P1 |  | 3 | 1 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 2 | 1 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 3 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 7 | 10 | 6 | 6 |
| P1 |  | 3 | 1 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 2 | 1 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 10 | 11 | 8 | 7 |
| P1 |  | Finished during **fourth** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 12 | 12 | 8 | 10 |
| P1 |  | Finished during **fourth** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | Finished during the **fifth** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |

This illustrates that the system is in a safe state.

The safe sequence is P0 🡪 P3 🡪 P4 🡪 P1 🡪 P2.

1. If a request from process P1 arrives for (1, 1, 0, 0), can the request be granted immediately?

This request will result in the following changes:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 2 | 0 | 0 | 1 |  | 4 | 2 | 1 | 2 |  | 2 | 2 | 1 | 1 |  | 2 | 2 | 2 | 1 |
| P1 |  | 4 | 2 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 1 | 0 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  | 0 | 1 | 1 | 2 |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 3 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 4 | 2 | 2 | 2 |
| P1 |  | 4 | 2 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 1 | 0 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  | 0 | 1 | 1 | 2 |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 3 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 10 | 5 | 4 | 6 |
| P1 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  | 0 | 1 | 1 | 2 |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 3 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 11 | 8 | 5 | 8 |
| P1 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | Finished during **fourth** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | 1 | 4 | 3 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 3 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | Finished during **first** step | | | | | | | | |  |  |  |  |  |  | 12 | 12 | 8 | 10 |
| P1 |  | Finished during **second** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P2 |  | Finished during **third** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P3 |  | Finished during **fourth** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| P4 |  | Finished during **fifth** step | | | | | | | | |  |  |  |  |  |  |  |  |  |  |

Because all processes are completed **without** deadlock, the system will be in a **safe state** if the request of P1 is granted. Hence, the request would be **granted immediately**

1. If a request from process P4 arrives for (0, 0, 2, 0), can the request be granted immediately?

This request will result in the following **changes**:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 2 | 0 | 0 | 1 |  | 4 | 2 | 1 | 2 |  | 2 | 2 | 1 | 1 |  | 3 | 3 | 2 | 1 |
| P1 |  | 3 | 1 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 2 | 1 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  | 0 | 1 | 1 | 2 |  |  |  |  |  |
| P4 |  | 1 | 4 | 5 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 1 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Allocation** | | | |  | **Max** | | | |  | **Need** | | | |  | **Available** | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |  | A | B | C | D |
| P0 |  | 2 | 0 | 0 | 1 |  | 4 | 2 | 1 | 2 |  | 2 | 2 | 1 | 1 |  | 3 | 3 | 0 | 1 |
| P1 |  | 3 | 1 | 2 | 1 |  | 5 | 2 | 5 | 2 |  | 2 | 1 | 3 | 1 |  |  |  |  |  |
| P2 |  | 2 | 1 | 0 | 3 |  | 2 | 3 | 1 | 6 |  | 0 | 2 | 1 | 3 |  |  |  |  |  |
| P3 |  | 1 | 3 | 1 | 2 |  | 1 | 4 | 2 | 4 |  | 0 | 1 | 1 | 2 |  |  |  |  |  |
| P4 |  | 1 | 4 | 5 | 2 |  | 3 | 6 | 6 | 5 |  | 2 | 2 | 1 | 3 |  |  |  |  |  |

At this point the system would be unsafe because none of the processes would be able to complete. Hence, if such request arrives it would **not** be granted immediately because the system would **reject the request**.

1. A single-lane bridge connects the two Vermont villages of north Tunbridge and south Tunbridge. Farmers in the two villages use this bridge to deliver their produce to the neighboring town. The bride can become deadlocked if a north bound and a south bound farmer get to the bridge at the same time. (Vermont farmers are stubborn and are unable to back up.) Using semaphores and/or mutex locks, design an algorithm in pseudocode that prevents deadlock. Initially, do not be concerned about starvation (the situation in which north bound farmers prevent southbound farmers from using the bridge, or vice versa).

Semaphore okCross = 1

Void beginCrossing(){

okCross.wait();

}

Void doneCrossing(){

okCross.signal();

}

References:

1. Operating System Concepts, Ninth Edition, Abraham Silberschartz, Peter bear Galvin, Greg Gagne. Entire Chapter 7 for each question.
2. Dr. Raisa Szabo’s ch7\_Deadlock\_mod.ppx.
3. <http://web.cs.wpi.edu/~cs3013/c07/lectures/Section07-Deadlocks.pdf>.
4. http://www.cs.rpi.edu/academics/courses/fall04/os/c10/.

In this chapter I learned about deadlocks in greater detail. I clarified that in a multiprogramming environment, several process may compete for a finite number of resources. If the requested resources are not available when a process request it, the process enters into a waiting state. Sometimes, a waiting process is never again able to change state, because the resources it has requested are held by other waiting processes. And this is what we know as deadlock. I explored methods that an operating system can use to prevent or deal with deadlocks. I now understand that although some application can identify program s that may deadlock, operating systems typically do not provide deadlock-preventing facilities, and it remains the responsibility of programmers to ensure that they design deadlock-free programs. I derived that truly understanding deadlock problems is a necessity for future programmers as these problems an only become more common, given the current trends, including larger numbers of processes, multithread program many more resources within a system, and an emphasis on long-lived file and database servers rather than batch systems.