Lab 5: Resource Management

Use the banker's algorithm to determine if the following situations are safe or unsafe. You should include the order in which processes will run to create a safe state, or to arrive at an unsafe state.

1)

Task 1: 3 3 2 + 1 2 0 = 4 5 2 (Need: 4 4 2)

Task 2: 1 0 1 + 4 5 2 = 5 5 3 (Need: 2 2 1)

Task 3: 1 2 1 + 5 5 3 = 6 7 4 (Need: 5 1 2)

Task 4: 4 0 1 + 6 7 4 = 10 7 5 (Need: 4 5 5)

The order is 1, 2, 3, 4, which will create a safe state.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Resource A |  | Resource B |  | Resource C |  |
| Available resources: | 1 |  | 2 |  | 0 |  |
| Process | Has | Needs | Has | Needs | Has | Needs |
| Task 1 | 3 | 4 | 3 | 4 | 2 | 2 |
| Task 2 | 1 | 2 | 0 | 2 | 1 | 1 |
| Task 3 | 1 | 5 | 2 | 1 | 1 | 2 |
| Task 4 | 4 | 4 | 0 | 5 | 1 | 5 |

2)

This situation will not result in a safe state

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Resource A |  | Resource B |  | Resource C |  |
| Available resources: | 4 |  | 0 |  | 0 |  |
| Process | Has | Needs | Has | Needs | Has | Needs |
| Task 1 | 3 | 4 | 3 | 4 | 2 | 2 |
| Task 2 | 1 | 2 | 0 | 2 | 2 | 3 |
| Task 3 | 2 | 5 | 2 | 3 | 1 | 2 |
| Task 4 | 4 | 4 | 0 | 6 | 1 | 5 |

3)

This situation will not result in a safe state.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Resource A |  | Resource B |  | Resource C |  |
| Available resources: | 1 |  | 0 |  | 0 |  |
| Process | Has | Needs | Has | Needs | Has | Needs |
| Task 1 | 3 | 4 | 3 | 4 | 2 | 2 |
| Task 2 | 1 | 2 | 2 | 2 | 2 | 3 |
| Task 3 | 2 | 5 | 2 | 3 | 1 | 2 |
| Task 4 | 4 | 5 | 2 | 2 | 1 | 5 |

4) Based on the banker's algorithm, what are heuristics that could simplify the determination of a safe or unsafe state?

A heuristic that would simply the determination of a safe or unsafe state is to look at the available resources and compare them to what each task needs. If the available resources cannot satisfy any task, it’s easy to determine it’s unsafe. If the resources satisfy at least 1 task, then you know it COULD be safe, but further logic is required.

5) Draw the following table in resource chart form:

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Allocated  A B C | Max Allocated  A B C | Available  A B C |
| Task 1 | 1 3 2 | 2 4 2 |  |
| Task 2 | 1 1 2 | 2 2 3 | 2 1 2 |
| Task 3 | 2 2 1 | 5 4 2 |  |

The satisfying order is 1, 2, and 3.

Task 1: (1 3 2) + (2 1 2) = (3 4 4) (Needs 2 4 2)

Task 2: (1 1 2) + (3 4 4) = (4 5 6) (Needs 2 2 3)

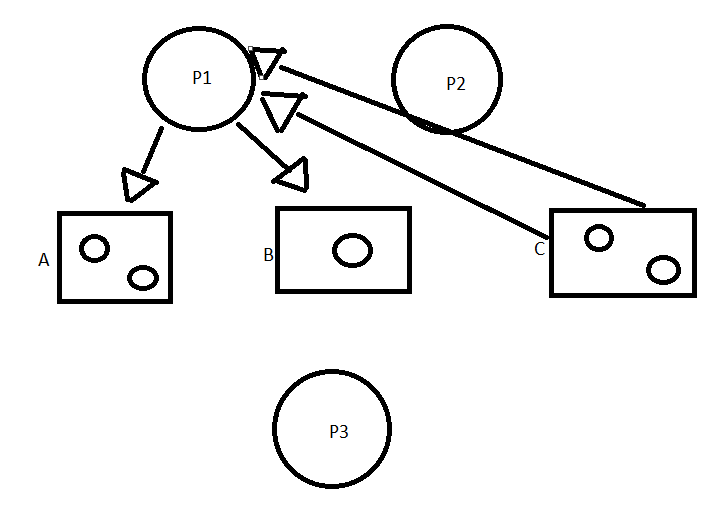
Task 3: (2 2 1) + (4 5 6) = (6 7 7) (Needs 5 4 2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Resource A |  | Resource B |  | Resource C |  |
| Available resources: | 2 |  | 1 |  | 2 |  |
| Process | Has | Needs | Has | Needs | Has | Needs |
| Task 1 | 1 | 2 | 3 | 4 | 2 | 2 |
| Task 2 | 1 | 2 | 1 | 2 | 2 | 3 |
| Task 3 | 2 | 5 | 2 | 4 | 1 | 2 |

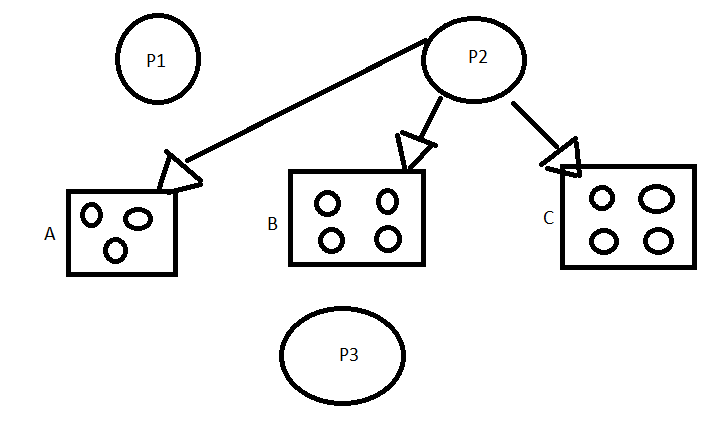
6) Draw the resolution of the chart for number 5: (a new chart for each step should be drawn)

Step 1

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Holding A B C | Waiting A B C | Available A B C |
| Task 1 | 0 0 2 | 1 1 0 | 2 1 2 |

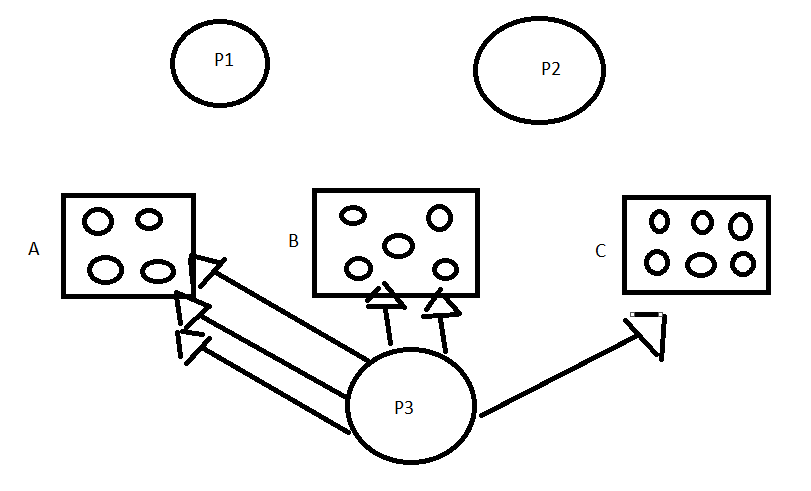
  
  
Step 2

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Holding A B C | Waiting A B C | Available A B C |
| Task 2 | 0 0 0 | 1 1 1 | 3 4 4 |



Step 3

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Holding A B C | Waiting A B C | Available A B C |
| Task 3 | 0 0 0 | 3 2 1 | 4 5 6 |



7) List 4 examples of limited resources. Cite any sources  
  
- Energy (https://ubiquity.acm.org/article.cfm?id=1386854)

- Physical Space

- Memory  
- Process Management

8) Explain the difference between a binary semaphore and a counting semaphore.

A binary semaphore has the possible values of 0 and 1. This means that one task at a time is enforced. A counting semaphore have the values from 0-N, which means it can allow up to N tasks to be enforced at a time.

Extra Credit: Draw the process chart for #3 (5 pts ec)