**Homework Assignment #3**

1. **Briefly explain the starvation of a process in priority scheduling and how to prevent starvation?**

- Priority scheduling can suffer from a major problem known as indefinite blocking, or starvation. The solution to this problem is called aging, which means that as time progresses, the priority of the process increases. So, process priorities are updated dynamically, and scheduling happens at every moment that priorities are updated.

1. **Suppose that there are 12 resources available to three processes. At time 0, the following data is collected. The table indicates the process, the maximum number of resources needed by the process, and the number of resources currently owned by each process. Is the system in a safe state? Why?**

- It is safe because each process can safely request can be satisfied by currently available resources + resources held by prior process. Thus, avoiding deadlock in the system as the number of free resources are available for each process before running.

1. **Answer the following questions using the banker’s algorithm:**
   1. **How many instances of each resource type does the system have?**

- A (3 Instances), B (14 Instances), C (12 Instances), and D (12 Instances)

* 1. **What is the content of the matrix Need? Draw the table.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 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 |

* 1. **Is the system in a safe state? Why?**

Available = 1 | 5 | 2 | 0

|  |  |
| --- | --- |
|  | Work (A | B | C | D) |
| P0 | 1|5|2|0 |
| P1 | 1|12|7|0 |
| P2 | 2|12|7|2 |
| P3 | 2|12|9|2 |
| P4 | 2|18|13|2 |

This system is a safe state when finished in the sequence (P0 , P1, P2, P3, P4).

* 1. **If a request from P1 arrives for (0,4,2,0), can the request be granted immediately? Explain why. (Hint: Suppose the request is granted, then is the system safe? If yes, then the request can be immediately; else, it cannot be granted. So, for this question, you need to inspect whether the system is safe rather after the request is granted)**

- The system is safe becomes the need at P1 (0,7,5,0) is greater than the request that arrives (0,4,2,0).