

Operating System Labs July-Dec-2017

Assignment 5

Exercise 1: Write a C/C++ or Java code to implement a deadlock detection algorithm. The input to your program is a 7x7 array of resource allocation and demand requests. This array is filled with w, x, and n, where w = waiting for resource, x = exclusively holding a resource, and n = no need for the resource. The rows of this array represent jobs or processes and the columns represent resources. Following is a test case array:

```
String [][] grid = { {"w","n","x","x","x","n","n"},
                     {"x","w","n","n","n","x","n"},
                     {"n","x","w","n","n","n","n"},
                     {"n","n","n","n","n","n","x"},
                     {"n","n","n","n","n","n","n"},
                     {"n","n","w","n","n","n","w"},
                     {"w","w","w","w","w","w","w"} };
```

As you can see the deadlock is among Row 0, 1 and 2. P0 (process 1) is waiting for C0 (resource 1), but P1 (process 2) is holding it. P1 is waiting for C1 (resource 2) but P2 (process 3) is holding it. P2 is waiting for C2 (resource 3), but P0 is holding it.

The output of this program should display that

“Deadlock is detected involving:”

Processes P0, P1, P2

Resources C0, C1, C2

Also write two more test case arrays involving 6 processes and 6 resources, in this first new test case all six processes should be in deadlock and the second new test case there should be no deadlock.