Theodore Jagodits / CS 492 / HW#2 /

I pledge my honor that I have abided by the Stevens Honor System

1. It does solve the mutual exclusion problem.

2.

a. For round robin, each process gets an equal share of CPU for the first 10 minutes (1/5). Then since C is finished every process gets ¼ of the CPU for the next 8 minutes. Then D will finish, so then 1/3 of the CPU Is shared for the next 6 minutes. Then B finishes, so A and E each have half of the CPU for 4 minutes. E finishes, leaving A alone on the CPU for another 2 minutes.

C = 10min.

D = 10+8 = 18min.

B = 10+8+6 = 24min.

E = 10+8+6+4 = 28min.

A = 10+8+6+4+2 = 30min.

Avg. = (10+18+24+28+30)/5 = 22min.

b. B has the highest priority and runs for 6min. Then E runs in 8min. then A in 10min. Then C in 2min. and D in 4min..

B = 6min, E = 14min, A = 24min, C = 26min, 30min.

Avg. (6+14+24+26+30)/5 = 20min .

c.

A = 10min, B = 16min, C = 18min, D = 22min, E = 30min.

Avg. = (10+16+18+22+30)/5 = 19.2min

d.

C = 2min, D = 2+4 = 6min, B = 2+4+6 = 12min, 2+4+6+8 = 20min, A = 2+4+6+8+10 = 30min.

Avg. = (2+6+12+20+30)/5 = 14min.

3.

For X to fit, 35/50 + 20/100 + 10/200 + X/250 < 1. This equals to 19/20 + X/250 < 1. So X < 12.5 msec to be schedulable on the CPU.

4. in A.

| Process/resource | Rs1 | Rs2 | Rs3 | Rs4 | Rs5 |
|---|---|---|---|---|---|
| P1 | | | | | |
| P2 | 0 | 2 | 0 | 3 | 1 |
| P3 | 0 | 2 | 0 | 3 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| P4 | | | | | |

The table is constructed from p2, since it has the only row that contains R<=A. The eligible values that are added are (0 1 0 2 1)+ (0 1 0 1 0) = (0 2 0 3 1). The only row that is less or equal is P3. So we add P3. (0 2 0 3 1) + (0 0 0 0 1) = (0 2 0 3 2). Then there are no rows less than that so we arrive at deadlock with P1 and P4.

5.

The constructed Request table is:

| | | | | | |
|---|---|---|---|---|---|
| P1 | 0 | 1 | 0 | 0 | 1 |
| P2 | 0 | 2 | 1 | 0 | 0 |
| P3 | 1 | 0 | 3 | 0 | 0 |
| P3 | 0 | 0 | 1 | 1 | 1 |

So the smallest value X can be is 1. Making A (0 0 1 1 1). P4 runs, turning A into (1 1 1 1 0)+(0 0 1 1 1) = (1 1 2 2 1). Then P1 runs since it can turning A into (2 1 4 3 2). After that, P3 runs so A = (3 2 4 4 2), then P2 leaving A = (5 2 5 5 2).