CO-562 OS Problem Sheet 5

Shahin Mammadov

October 12, 2020

1 Safe States

A system has n = 4 processes, m = 5 resource types, and the number of resources for each resource type is given by t = (6, 15, 8, 10, 9). The system is in the following state:

$$M = \begin{bmatrix} 3 & 5 & 8 & 10 & 1 \\ 2 & 5 & 3 & 3 & 2 \\ 4 & 12 & 4 & 9 & 2 \\ 6 & 1 & 4 & 5 & 5 \end{bmatrix} A = \begin{bmatrix} 0 & 2 & 1 & 1 & 1 \\ 0 & 5 & 3 & 1 & 1 \\ 0 & 7 & 1 & 2 & 1 \\ 3 & 1 & 1 & 1 & 0 \end{bmatrix}$$

First we need to find the remaining maximum need $\boldsymbol{N}$ for all processes:

$$N = M - A = \begin{bmatrix} 3 & 5 & 8 & 10 & 1 \\ 2 & 5 & 3 & 3 & 2 \\ 4 & 12 & 4 & 9 & 2 \\ 6 & 1 & 4 & 5 & 5 \end{bmatrix} - \begin{bmatrix} 0 & 2 & 1 & 1 & 1 \\ 0 & 5 & 3 & 1 & 1 \\ 0 & 7 & 1 & 2 & 1 \\ 3 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 7 & 9 & 0 \\ 2 & 0 & 0 & 2 & 1 \\ 4 & 5 & 3 & 7 & 1 \\ 3 & 0 & 3 & 4 & 5 \end{bmatrix}$$

Now we find available resource instances $\boldsymbol{a}$:

$$a = t - colsum(A) = \begin{bmatrix} 6 & 15 & 8 & 10 & 9 \end{bmatrix} - \begin{bmatrix} 3 & 15 & 6 & 5 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 2 & 5 & 6 \end{bmatrix}$$

$$\boldsymbol{a} = \begin{bmatrix} 3 & 0 & 2 & 5 & 6 \end{bmatrix}$$

Now we have to find a sequence such that all processes can obtain their needed resources and then terminate. We start comparing each process to $\boldsymbol{a}$. Lets once again remember the remaining maximum need N:

$$N = \left[\begin{array}{c|ccccc} o & \mathrm{R}_1 & \mathrm{R}_2 & \mathrm{R}_3 & \mathrm{R}_4 & \mathrm{R}_5 \\ \hline \mathrm{P}_1 & 3 & 3 & 7 & 9 & 0 \\ \mathrm{P}_2 & 2 & 0 & 0 & 2 & 1 \\ \mathrm{P}_3 & 4 & 5 & 3 & 7 & 1 \\ \mathrm{P}_4 & 3 & 0 & 3 & 4 & 5 \end{array} \right]$$

- P_1 can not be allocated - P_1 is bigger than $\boldsymbol{a}$.
- P_2 can be allocated - a = [3 5 5 6 7].
- P_3 can not be allocated - P_3 is bigger than$\boldsymbol{a}$.
- P_4 can be allocated - a = [6 6 6 7 7].
- P_1 can not be allocated - P_1 is bigger than $\boldsymbol{a}$.
- P_3 can be allocated - a = [6 13 7 9 8].
- P_1 can be allocated - a = [6 15 8 10 9].

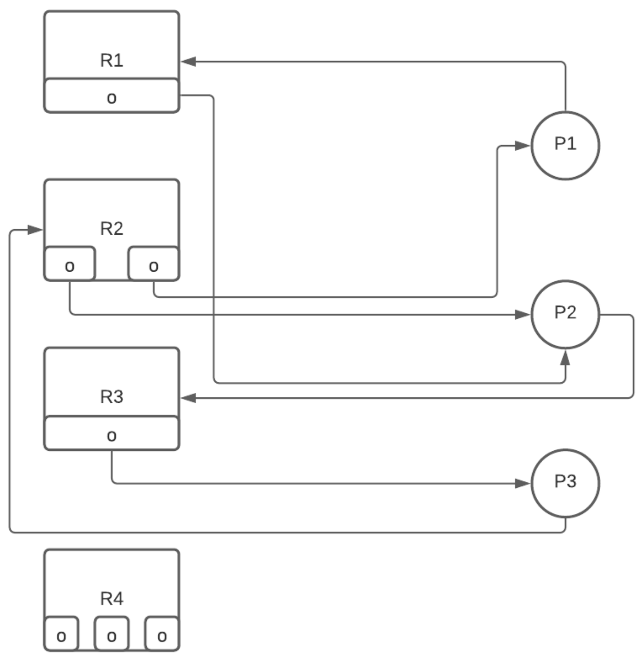
Now we know that safe sequence is P_2 $\rightarrow$P$_4$ $\rightarrow$P$_3$ $\rightarrow$P$_1$.

2 Deadlock Detection

A system has n = 3 processes, m = 4 resource types, and the number of resources for each resource type is given by t = (1, 2, 1, 3). The system is in the following state:

$$A = \begin{array}{c|cccc} o & R_1 & R_2 & R_3 & R_4 \\ \hline P_1 & 0 & 1 & 0 & 0 \\ P_2 & 1 & 1 & 0 & 0 \\ P_3 & 0 & 0 & 1 & 0 \end{array} \quad N = \begin{array}{c|cccc} o & R_1 & R_2 & R_3 & R_4 \\ \hline P_1 & 1 & 0 & 0 & 0 \\ P_2 & 0 & 0 & 1 & 0 \\ P_3 & 0 & 1 & 0 & 0 \end{array}$$

Lets draw resource allocation graph:



- R_2 is allocated to process P_1.
- R_2 and R_1 is allocated to process P_2.
- R_3 is allocated to process P_3.
- P_1 needs R_1 resources.
- P_2 needs R_3 resources.
- P_3 needs R_2 resources.

Lets see if the system is deadlocked. We need to find number of available resource instances ***a*** first.

$$a = t - colsum(A) = \begin{bmatrix}1 & 2 & 1 & 3\end{bmatrix} - \begin{bmatrix}1 & 2 & 1 & 0\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 3\end{bmatrix}$$

$$a = \begin{bmatrix}0 & 0 & 0 & 3\end{bmatrix}$$

Now lets see if processors can be allocated and if we can get safe sequence:

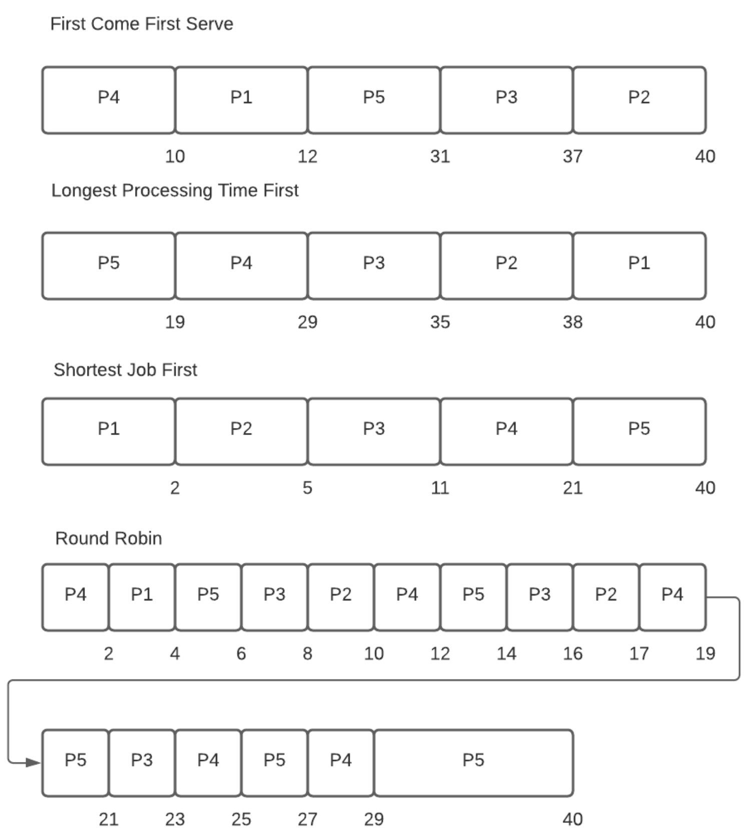
$$N = \begin{array}{c|cccc} o & R_1 & R_2 & R_3 & R_4 \\ \hline P_1 & 1 & 0 & 0 & 0 \\ P_2 & 0 & 0 & 1 & 0 \\ P_3 & 0 & 1 & 0 & 0 \end{array}$$

- P_1 can not be allocated - P_1 is bigger than $\boldsymbol{a}$.
- P_2 can not be allocated - P_2 is bigger than $\boldsymbol{a}$.
- P_3 can not be allocated - P_3 is bigger than $\boldsymbol{a}$.
- P_4 can not be allocated - P_4 is bigger than $\boldsymbol{a}$.

We see that no processes can be allocated as the processes are not satisfied with the available resources so system gets deadlocked.

3 Scheduling

Five proceses arrived in the order P4 - P1 - P5 - P3 - P2 and they are all ready at time t = 0. Process P1 needs 2 time units, P2 needs 3 time units, P3 needs 6 time units, P4 needs 10 time units, and P5 needs 19 time units. There is only one CPU in the system.



Here are the resulting schedule for the scheduling strategies with time slice of 2. Properties are:

- P1 have 2 time units.
- P2 needs 3 time units.
- P3 needs 6 time units.
- P4 needs 10 time units.
- P5 needs 19 time units.

.
So now we will calculate average completion time for each schedule. Here:

	FCFS	LPTF	SJF	RR
P4	10	29	21	29
P1	12	40	2	4
P5	31	19	40	40
P3	37	35	11	23
P2	40	38	5	17
TOTAL	130	161	79	113

$$AvgFCFS = 130 \div 5 = 26$$

$$AvgLPTF = 161 \div 5 = 32.2$$

$$AvgSJF = 79 \div 5 = 15.8$$

$$AvgRR = 113 \div 5 = 22.6$$

4 Linking

file	symbol	internal	external	weak symbol	strong symbol
a.c	x		✓	✓	
a.c	y		✓	✓	
a.c	f	✓			✓
a.c	g		✓		✓
b.c	x		✓		✓
b.c	y	✓			✓
b.c	f		✓		✓
b.c	g		✓	✓	

1. The Internal symbols are not accessible to the external translation when it is declared
2. The external symbols are accessible to the external translation when it is declared
3. The weak symbols are not initialized variable or not undefined functions in globally
4. The strong symbols are those in which variable and functions are initialized and declared globally

b) The output is:

```
main.c : (.text + 0xa ) : undefined reference to ' f '

collect2: error: ld returned 1 exit status
```

The function f() is not defined. When the main function is compiled, compiler sees there is no function definition in the main function so linker exists with the error code ***'undefined reference'***.