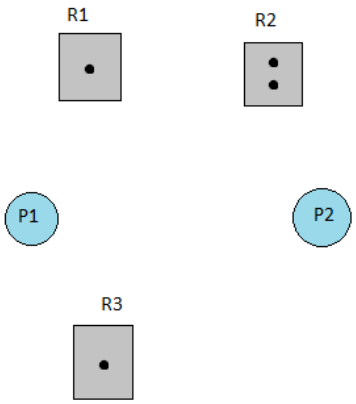
Operating Systems
Project 18

Haukur Óskar Þorgeirsson
hth152@hi.is

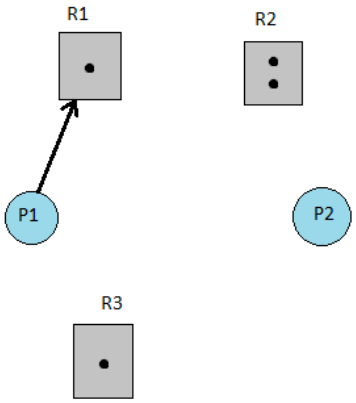
Matthías Páll Gissurarson
mpg3@hi.is

14. mars 2013

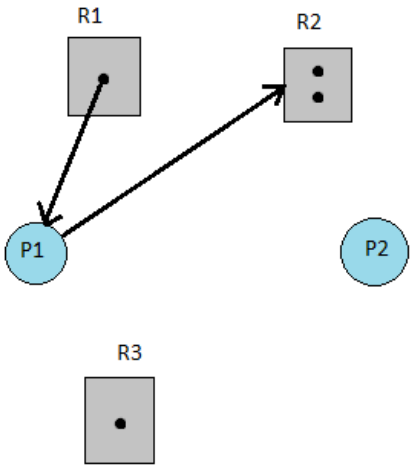
1



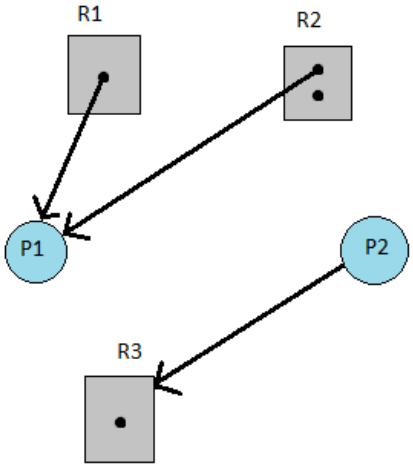
Mynd 1: At the beginning



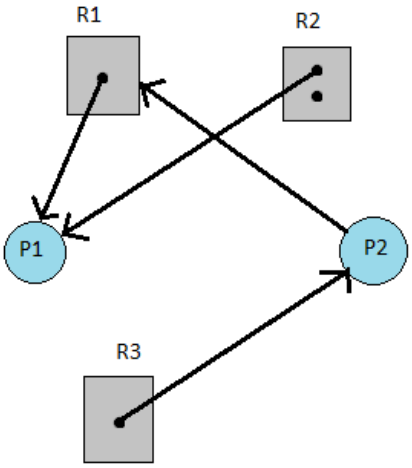
Mynd 2: P_1 requests R_1



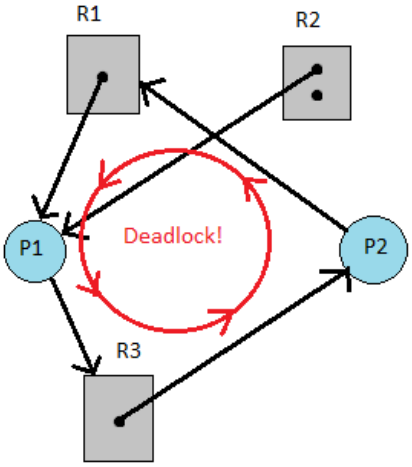
Mynd 3: P_1 is assigned R_1 and requests R_2



Mynd 4: P_1 is assigned an instance of R_2 and P_2 requests R_3



Mynd 5: P_2 is assigned R_3 and requests R_1



Mynd 6: P_2 is not assigned R_1 as P_1 has it. P_1 requests R_3 but will not get it because P_2 has it. Neither process can continue as they are both waiting for a resource the other one has but will not release until it has finished. We can also see the cycle clearly, it is a deadlock.

2

At the beginning, the state is such:

$$N = M - C = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(C is all zeroes as this is the initial state and no resources have been requested or allocated. M describes how each process needs one of each resource.)

$$A = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Now,

$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \leq \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$ (pointwise, the former being the first row of M)
So we mark P_1 and $A := A + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ (The latter being the first row of C). Also,

$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \leq \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$ (pointwise, the former being the second row of M)
So we mark P_2 and $A := A + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ (The latter being the second row of C).

Both processes have been marked, so this is a safe state. an example og a schedule that would be without deadlocks is:

P_1: R_1; P_1: R_2; P_1: R_3; P_2: R_3; P_2: R_1; P_2: R_2.