

## Algoritmo do Banqueiro

Detecção de Deadlock com várias recursos de cada tipo

Algoritmo baseado numa matriz para detecção de deadlock com  $n$  processos e  $m$  classes de recursos diferentes, com  $E_i$  recursos da classe  $i$ , sendo  $(1 \leq i \leq m)$ .

Estrutura de dados:

$E = (E_1, E_2, E_3, \dots, E_m)$ : vetor de recursos existentes

$A = (A_1, A_2, A_3, \dots, A_m)$ : vetor de recursos disponíveis

$C_{nm}$  = matriz de alocação corrente

$C_{ij}$  = número de instâncias de um recurso  $j$  entregue ao processo  $i$

$R_{nm}$  = matriz de requisições

$R_{ij}$  = número de instâncias do recurso  $j$  de que o processo  $P_i$  precisa

$$\sum_{i=1}^m C_{ij} + A_j = E_j$$

Algoritmo:

1. Procure por um processo desmarcado  $P_i$ , para o qual a  $i$ -ésima linha de  $R$  é menor do que a correspondente de  $A$ ;
2. Se um processo com tais característica for encontrado, adicione a  $i$ -ésima linha de  $C$  a  $A$ , marque o processo e volte para o passo 1;
3. Se não houver nenhum processo nesta situação, o algoritmo termina.

Exemplo:

Recursos existentes:

$E = (4 \ 2 \ 3 \ 1) \ 4$  (dvd rw) 2 (plotters) 3 (impressoras) 1 (blue-ray)

Recursos disponíveis:  $A = (2 \ 1 \ 0 \ 0)$

Matriz de alocação corrente

$C =$

0	0	1	0	P1
2	0	0	1	P2
0	1	2	0	P3

## Matriz de requisições

R =

2	0	0	1	P1
1	0	1	0	P2
2	1	0	0	P3

P3 (rodar) A = (2 2 2 0)

P2 (rodar) A = (4 2 2 1)

P1 (rodar) não há deadlock

Obs: Se R[3,4] fosse igual a 1, todo o processo estaria em deadlock.

## Tentativa de evitar o Deadlock

### Estados seguros e inseguros

Um estado é seguro se não provocar deadlock, e houver uma maneira de satisfazer todas as requisições pendentes partindo dos processos em execução.

1 - recurso, 10 - instâncias, 7 - alocados, 3 - livres, P - possui, N - necessita

	P	N			P	N			P	N			P	N			P	N	
A	3	9			A	3	9			A	3	9					A	3	9
B	2	4			B	4	4			B	0	-					B	0	-
C	2	7			C	2	7			C	2	7					C	0	-
Livre = 3					Livre = 1					Livre = 5							Livre = 7		

Este exemplo mostra um estado seguro pois o sistema através de seu escalamento cuidadoso pode evitar a ocorrência de deadlocks.

A diferença de um estado seguro e inseguro é que, de um estado seguro, o sistema pode garantir que todos os processos vão terminar, enquanto que a parti de um inseguro esta garantia não pode ser dada.

Obs: Se A[1] tivesse o valor 4, em vez de 03 (círculo vermelho), os processos estariam em um estado inseguro.