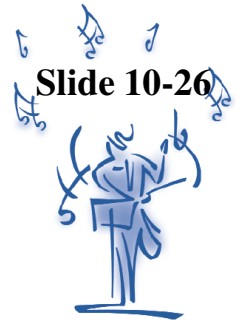


# Banker's Algorithm



- Copy the  $\text{alloc}[i,j]$  table to  $\text{alloc}'[i,j]$
- Given  $C$ ,  $\text{maxc}$  and  $\text{alloc}'$ , compute avail vector
- Find  $p_i$ :  $\text{maxc}[i,j] - \text{alloc}'[i,j] \leq \text{avail}[j]$  for  $0 \leq j < m$  and  $0 \leq i < n$ .
  - If no such  $p_i$  exists, the state is unsafe
  - If  $\text{alloc}'[i,j]$  is 0 for all  $i$  and  $j$ , the state is safe
- Set  $\text{alloc}'[i,j]$  to 0; deallocate all resources held by  $p_i$ ; go to Step 2

# Example

## Maximum Claim

Process	R <sub>0</sub>	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
p <sub>0</sub>	3	2	1	4
p <sub>1</sub>	0	2	5	2
p <sub>2</sub>	5	1	0	5
p <sub>3</sub>	1	5	3	0
p <sub>4</sub>	3	0	3	3

## Allocated Resources

Process	R <sub>0</sub>	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
p <sub>0</sub>	2	0	1	1
p <sub>1</sub>	0	1	2	1
p <sub>2</sub>	4	0	0	3
p <sub>3</sub>	0	2	1	0
p <sub>4</sub>	1	0	3	0
Sum	7	3	7	5

$$C = \langle 8, 5, 9, 7 \rangle$$

- Compute total allocated
- Determine available units

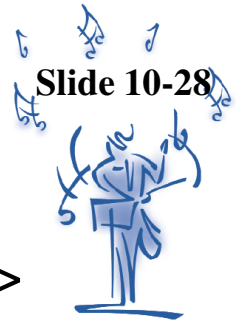
$$\begin{aligned} \text{avail} &= \langle 8-7, 5-3, 9-7, 7-5 \rangle \\ &= \langle 1, 2, 2, 2 \rangle \end{aligned}$$

- Can anyone's maxc be met?

$$\begin{aligned} \text{maxc}[2,0] - \text{alloc}'[2,0] &= 5-4 = 1 \leq 1 = \text{avail}[0] \\ \text{maxc}[2,1] - \text{alloc}'[2,1] &= 1-0 = 1 \leq 2 = \text{avail}[1] \\ \text{maxc}[2,2] - \text{alloc}'[2,2] &= 0-0 = 0 \leq 2 = \text{avail}[2] \\ \text{maxc}[2,3] - \text{alloc}'[2,3] &= 5-3 = 2 \leq 2 = \text{avail}[3] \end{aligned}$$

- P<sub>2</sub> can exercise max claim

$$\begin{aligned} \text{avail}[0] &= \text{avail}[0] + \text{alloc}'[2,0] = 1+4 = 5 \\ \text{avail}[1] &= \text{avail}[1] + \text{alloc}'[2,1] = 2+0 = 2 \\ \text{avail}[2] &= \text{avail}[2] + \text{alloc}'[2,2] = 2+0 = 2 \\ \text{avail}[3] &= \text{avail}[3] + \text{alloc}'[2,3] = 2+3 = 5 \end{aligned}$$



# Example

$$C = \langle 8, 5, 9, 7 \rangle$$

## Maximum Claim

Process	R <sub>0</sub>	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
p <sub>0</sub>	3	2	1	4
p <sub>1</sub>	0	2	5	2
p <sub>2</sub>	5	1	0	5
p <sub>3</sub>	1	5	3	0
p <sub>4</sub>	3	0	3	3

- Compute total allocated
- Determine available units

$$\begin{aligned} \text{avail} &= \langle 8-7, 5-3, 9-7, 7-5 \rangle \\ &= \langle 5, 2, 2, 5 \rangle \end{aligned}$$

- Can anyone's maxc be met?

## Allocated Resources

Process	R <sub>0</sub>	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
p <sub>0</sub>	2	0	1	1
p <sub>1</sub>	0	1	2	1
p <sub>2</sub>	0	0	0	0
p <sub>3</sub>	0	2	1	0
p <sub>4</sub>	1	0	3	0
Sum	3	3	7	2

$$\begin{aligned} \text{maxc}[4,0] - \text{alloc}'[4,0] &= 5-1 = 4 \leq 5 = \text{avail}[0] \\ \text{maxc}[4,1] - \text{alloc}'[4,1] &= 0-0 = 0 \leq 2 = \text{avail}[1] \\ \text{maxc}[4,2] - \text{alloc}'[4,2] &= 3-3 = 0 \leq 2 = \text{avail}[2] \\ \text{maxc}[4,3] - \text{alloc}'[4,3] &= 3-0 = 3 \leq 5 = \text{avail}[3] \end{aligned}$$

- P<sub>4</sub> can exercise max claim

$$\begin{aligned} \text{avail}[0] &= \text{avail}[0] + \text{alloc}'[4,0] = 5+1 = 6 \\ \text{avail}[1] &= \text{avail}[1] + \text{alloc}'[4,1] = 2+0 = 2 \\ \text{avail}[2] &= \text{avail}[2] + \text{alloc}'[4,2] = 2+3 = 5 \\ \text{avail}[3] &= \text{avail}[3] + \text{alloc}'[4,3] = 5+0 = 5 \end{aligned}$$



# Example

$$C = \langle 8, 5, 9, 7 \rangle$$

## Maximum Claim

Process	R <sub>0</sub>	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
p <sub>0</sub>	3	2	1	4
p <sub>1</sub>	0	2	5	2
p <sub>2</sub>	5	1	0	5
p <sub>3</sub>	1	5	3	0
p <sub>4</sub>	3	0	3	3

- Compute total allocated
- Determine available units

$$\begin{aligned} \text{avail} &= \langle 8-7, 5-3, 9-7, 7-5 \rangle \\ &= \langle 6, 2, 5, 5 \rangle \end{aligned}$$

- Can anyone's maxc be met?  
(Yes, any of them can)

## Allocated Resources

Process	R <sub>0</sub>	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
p <sub>0</sub>	2	0	1	1
p <sub>1</sub>	0	1	2	1
p <sub>2</sub>	0	0	0	0
p <sub>3</sub>	0	2	1	0
p <sub>4</sub>	0	0	0	0
Sum	2	1	4	2