OS | Assignment 2

operating systems

by Soravis Varayuththasawad 5913094

Banker’s algorithm

Banker's algorithm

Process	Allocation	Max	Available
P0	0 1 0	7 5 3	3 3 2
P1	2 0 0	3 2 2	
P2	3 0 2	9 0 2	
P3	2 1 1	2 2 2	
P4	0 0 2	4 3 3	

Banker's algorithm

Process	Allocation	Max	Need
P0	0 1 0	7 5 3	7 4 3
P1	2 0 0	3 2 2	1 2 2
P2	3 0 2	9 0 2	6 0 0
P3	2 1 1	2 2 2	0 1 1
P4	0 0 2	4 3 3	4 3 1

Banker's algorithm

Process	Allocation	Max	Available	Need
P0	0 1 0	7 5 3	3 3 2	7 4 3
P1	2 0 0	3 2 2	5 3 2	1 2 2
P2	3 0 2	9 0 2	7 4 3	6 0 0
P3	2 1 1	2 2 2	7 4 5	0 1 1
P4	0 0 2	4 3 3	7 5 5	4 3 1
			10 5 7	

Banker’s algorithm

Safe State

P1 P3 P4 P0 P2

Init value

```
processes = input()
processes = eval(processes)

types = input()
types = eval(types)

allocation = []
mx = []
need = [[0 for i in range(types)]for j in range(processes)]
available = [[0 for i in range(types)] for j in range(processes + 1)]
available_top = 0
safty = [0 for i in range(processes)]
```

Use the init value and calculate the need.

```
for i in range(processes):
    temp = input().split()
    for j in range(types):
        temp[j] = eval(temp[j])
    allocation.append(temp)

for i in range(processes):
    temp = input().split()
    for j in range(types):
        temp[j] = eval(temp[j])
    mx.append(temp)

temp = input().split()
for i in range(types):
    temp[i] = eval(temp[i])
    available[0][i] = temp[i]

print("Allocation : " + str(allocation))
print("Max : " + str(mx))
print("Available : " + str(available))

for i in range(processes):
    for j in range(types):
        need[i][j] = mx[i][j] - allocation[i][j]

print("Need : " + str(need))
```

Calculate the safe state.

```
#process
counter = processes
finish = True

while(finish):
    counter_temp = counter
    for i in range(processes):

        available_temp = ""
        need_temp = ""
        allocation_temp = ""

        for j in range(types):
            available_temp += str(available[available_top][j])
            need_temp += str(need[i][j])

        available_temp = int(available_temp)
        need_temp = int(need_temp)

        if available_temp >= need_temp:
            counter = counter - 1

            for j in range(types):
                need[i][j] = 9

            safty[available_top] = int(i)
            available_top = available_top + 1

            for j in range(types):
                allocation_temp += str(allocation[i][j])
```

```
            allocation_temp = int(allocation_temp)
            available_temp = available_temp + allocation_temp
            available_temp = str(available_temp).zfill(types)
            available_temp = " ".join(available_temp)
            print(available_temp)
            temp = available_temp.split()
            for j in range(types):
                temp[j] = eval(temp[j])
                available[available_top][j] = temp[j]

    if counter == 0:
        finish = False

    if counter == counter_temp:
        finish = False
        print("The request cannot be granted")

print("------------------------------------------------------------")
print("Safty : " + str(safty))
```

Input

```
5
3
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
3 3 2
```

output

```
C:\Users\Elk\Desktop>os_assignment2.py < banker_test.in
Allocation : [[0, 1, 0], [2, 0, 0], [3, 0, 2], [2, 1, 1], [0, 0, 2]]
Max : [[7, 5, 3], [3, 2, 2], [9, 0, 2], [2, 2, 2], [4, 3, 3]]
Available : [[3, 3, 2], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]]
Need : [[7, 4, 3], [1, 2, 2], [6, 0, 0], [0, 1, 1], [4, 3, 1]]
--------------------------------------------------------------------------------
5 3 2
7 4 3
7 4 5
7 5 5
1 0 5 7
--------------------------------------------------------------------------------
Safty : [1, 3, 4, 0, 2]
```

OS

operating systems

by Soravis Varayuththasawad 5913094