Usman Institute of Technology

Department of Computer Science Fall 2022

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

Course Instructor: Ma'am Shabina Mushtaq

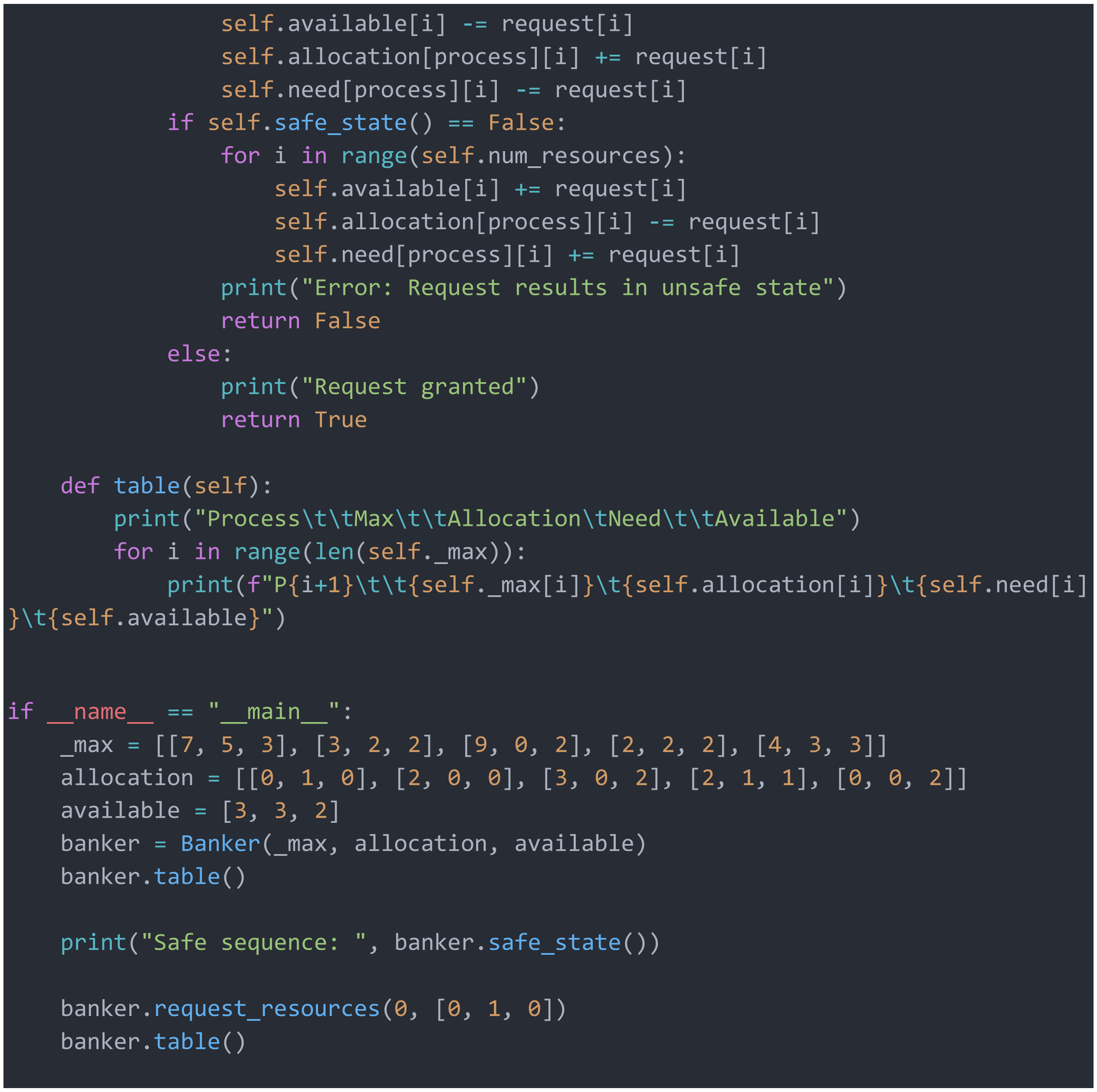
Date: 14-Jan-2023

Muhammad Waleed
20b-115-se
SE-B
OS Lab#13
Ma’am Shabina Mushtaq

Banker’s Algorithm (With Resource Request method):

```
class Banker:
    def __init__(self, _max, allocation, available):
        self._max = _max
        self.allocation = allocation
        self.available = available
        self.need = []
        self.num_resources = len(_max[0])
        self.calculate_need()

    def calculate_need(self):
        for i in range(len(self._max)):
            for j in range(self.num_resources):
                self.need.append(self._max[i][j]-self.allocation[i][j])
        self.need = [self.need[i:i+self.num_resources] for i in range(0,
len(self.need), self.num_resources)]

    def safe_state(self):
        work = self.available
        finish = [False for i in range(len(self.need))]
        sequence = []
        while False in finish:
            for i in range(len(self.need)):
                for j in range(len(self.need[i])):
                    if self.need[i][j] <= work[j] and finish[i] == False:
                        work[j] += self.allocation[i][j]
                        finish[i] = True
                        sequence.append(f"p{i+1}")
        return sequence

    def request_resources(self, process, request):
        if request[0] > self.need[process][0] or request[1] >
self.need[process][1] or request[2] > self.need[process][2]:
            print("Error: Request exceeds need")
            return False
        elif request[0] > self.available[0] or request[1] > self.available[1] or
request[2] > self.available[2]:
            print("Error: Request exceeds available")
            return False
        else:
            for i in range(self.num_resources):
```

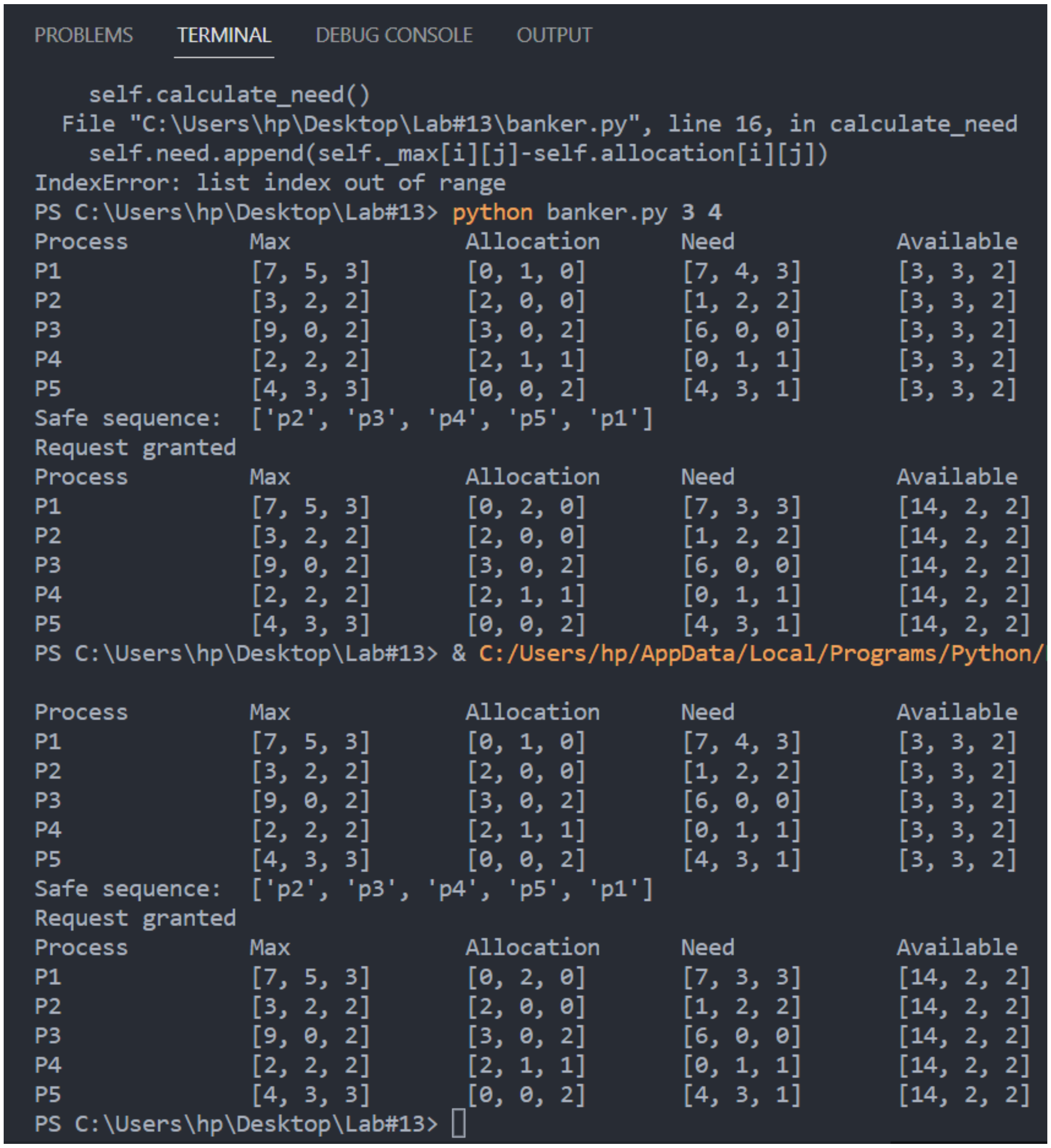
Muhammad Waleed
20b-115-se
SE-B
OS Lab#13
Ma’am Shabina Mushtaq



```
                self.available[i] -= request[i]
                self.allocation[process][i] += request[i]
                self.need[process][i] -= request[i]
            if self.safe_state() == False:
                for i in range(self.num_resources):
                    self.available[i] += request[i]
                    self.allocation[process][i] -= request[i]
                    self.need[process][i] += request[i]
                print("Error: Request results in unsafe state")
                return False
            else:
                print("Request granted")
                return True

    def table(self):
        print("Process\t\tMax\t\tAllocation\tNeed\t\tAvailable")
        for i in range(len(self._max)):
            print(f"P{i+1}\t\t{self._max[i]}\t{self.allocation[i]}\t{self.need[i]
}\t{self.available}")


if __name__ == "__main__":
    _max = [[7, 5, 3], [3, 2, 2], [9, 0, 2], [2, 2, 2], [4, 3, 3]]
    allocation = [[0, 1, 0], [2, 0, 0], [3, 0, 2], [2, 1, 1], [0, 0, 2]]
    available = [3, 3, 2]
    banker = Banker(_max, allocation, available)
    banker.table()

    print("Safe sequence: ", banker.safe_state())

    banker.request_resources(0, [0, 1, 0])
    banker.table()
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#13
Ma'am Shabina Mushtaq

Output:


PROBLEMS   TERMINAL   DEBUG CONSOLE   OUTPUT

```
    self.calculate_need()
  File "C:\Users\hp\Desktop\Lab#13\banker.py", line 16, in calculate_need
    self.need.append(self._max[i][j]-self.allocation[i][j])
IndexError: list index out of range
PS C:\Users\hp\Desktop\Lab#13> python banker.py 3 4
Process         Max             Allocation      Need            Available
P1              [7, 5, 3]       [0, 1, 0]       [7, 4, 3]       [3, 3, 2]
P2              [3, 2, 2]       [2, 0, 0]       [1, 2, 2]       [3, 3, 2]
P3              [9, 0, 2]       [3, 0, 2]       [6, 0, 0]       [3, 3, 2]
P4              [2, 2, 2]       [2, 1, 1]       [0, 1, 1]       [3, 3, 2]
P5              [4, 3, 3]       [0, 0, 2]       [4, 3, 1]       [3, 3, 2]
Safe sequence:  ['p2', 'p3', 'p4', 'p5', 'p1']
Request granted
Process         Max             Allocation      Need            Available
P1              [7, 5, 3]       [0, 2, 0]       [7, 3, 3]       [14, 2, 2]
P2              [3, 2, 2]       [2, 0, 0]       [1, 2, 2]       [14, 2, 2]
P3              [9, 0, 2]       [3, 0, 2]       [6, 0, 0]       [14, 2, 2]
P4              [2, 2, 2]       [2, 1, 1]       [0, 1, 1]       [14, 2, 2]
P5              [4, 3, 3]       [0, 0, 2]       [4, 3, 1]       [14, 2, 2]
PS C:\Users\hp\Desktop\Lab#13> & C:/Users/hp/AppData/Local/Programs/Python/

Process         Max             Allocation      Need            Available
P1              [7, 5, 3]       [0, 1, 0]       [7, 4, 3]       [3, 3, 2]
P2              [3, 2, 2]       [2, 0, 0]       [1, 2, 2]       [3, 3, 2]
P3              [9, 0, 2]       [3, 0, 2]       [6, 0, 0]       [3, 3, 2]
P4              [2, 2, 2]       [2, 1, 1]       [0, 1, 1]       [3, 3, 2]
P5              [4, 3, 3]       [0, 0, 2]       [4, 3, 1]       [3, 3, 2]
Safe sequence:  ['p2', 'p3', 'p4', 'p5', 'p1']
Request granted
Process         Max             Allocation      Need            Available
P1              [7, 5, 3]       [0, 2, 0]       [7, 3, 3]       [14, 2, 2]
P2              [3, 2, 2]       [2, 0, 0]       [1, 2, 2]       [14, 2, 2]
P3              [9, 0, 2]       [3, 0, 2]       [6, 0, 0]       [14, 2, 2]
P4              [2, 2, 2]       [2, 1, 1]       [0, 1, 1]       [14, 2, 2]
P5              [4, 3, 3]       [0, 0, 2]       [4, 3, 1]       [14, 2, 2]
PS C:\Users\hp\Desktop\Lab#13> 
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#13
Ma'am Shabina Mushtaq

Bash Script:

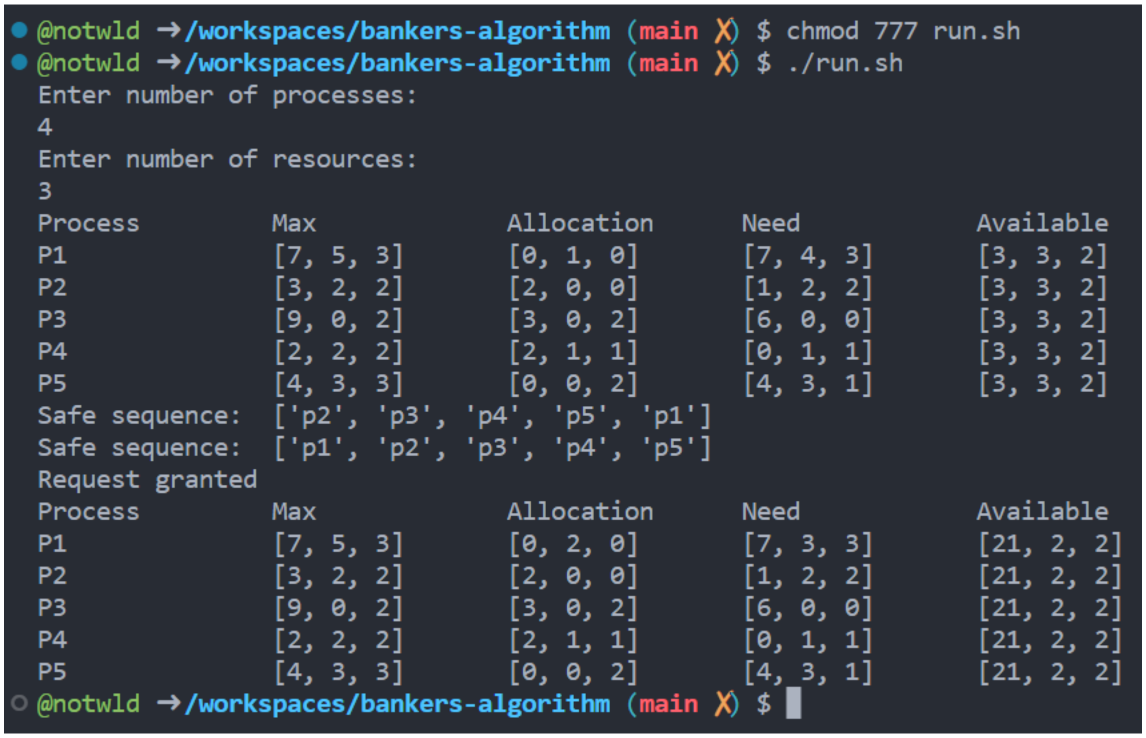
```
echo "Enter number of processes:"
read num_processes
echo "Enter number of resources:"
read num_resources

python3 modified_banker.py $num_processes $num_resources
```

Changes in banker.py

```
import sys

class Banker:
    def __init__(self, num_resources, num_processes, _max, allocation,
available):
        self._max = _max
        self.allocation = allocation
        self.available = available
        self.need = []
        self.num_resources = num_resources
        self.num_processes = num_processes
        self.calculate_need()

        ... rest of code


if __name__ == "__main__":
    num_resources = int(sys.argv[1])
    num_processes = int(sys.argv[2])
    _max = [[7, 5, 3], [3, 2, 2], [9, 0, 2], [2, 2, 2], [4, 3, 3]]
    allocation = [[0, 1, 0], [2, 0, 0], [3, 0, 2], [2, 1, 1], [0, 0, 2]]
    available = [3, 3, 2]
    banker = Banker(num_resources, num_processes,_max, allocation, available)
    banker.table()

    print("Safe sequence: ", banker.safe_state())

    print("Safe sequence: ", banker.safe_state())

    banker.request_resources(0, [0, 1, 0])
    banker.table()
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#13
Ma’am Shabina Mushtaq

Output:



```
● @notwld →/workspaces/bankers-algorithm (main ✗) $ chmod 777 run.sh
● @notwld →/workspaces/bankers-algorithm (main ✗) $ ./run.sh
  Enter number of processes:
  4
  Enter number of resources:
  3
  Process         Max             Allocation      Need            Available
  P1              [7, 5, 3]       [0, 1, 0]       [7, 4, 3]       [3, 3, 2]
  P2              [3, 2, 2]       [2, 0, 0]       [1, 2, 2]       [3, 3, 2]
  P3              [9, 0, 2]       [3, 0, 2]       [6, 0, 0]       [3, 3, 2]
  P4              [2, 2, 2]       [2, 1, 1]       [0, 1, 1]       [3, 3, 2]
  P5              [4, 3, 3]       [0, 0, 2]       [4, 3, 1]       [3, 3, 2]
  Safe sequence:  ['p2', 'p3', 'p4', 'p5', 'p1']
  Safe sequence:  ['p1', 'p2', 'p3', 'p4', 'p5']
  Request granted
  Process         Max             Allocation      Need            Available
  P1              [7, 5, 3]       [0, 2, 0]       [7, 3, 3]       [21, 2, 2]
  P2              [3, 2, 2]       [2, 0, 0]       [1, 2, 2]       [21, 2, 2]
  P3              [9, 0, 2]       [3, 0, 2]       [6, 0, 0]       [21, 2, 2]
  P4              [2, 2, 2]       [2, 1, 1]       [0, 1, 1]       [21, 2, 2]
  P5              [4, 3, 3]       [0, 0, 2]       [4, 3, 1]       [21, 2, 2]
○ @notwld →/workspaces/bankers-algorithm (main ✗) $
```