

Government Engineering College Thrissur

System Software Lab

Navaneeth D

TCR18CS043

S5, CSE

Bankers Algorithm

AIM

Implement the banker's algorithm for deadlock avoidance.

THEORY

The banker's algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation for predetermined maximum possible amounts of all resources, then makes an "s-state" check to test for possible activities, before deciding whether allocation should be allowed to continue.

For the Banker's algorithm to work, it needs to know three things:

- How much of each resource each process could possibly request[**MAX**]
- How much of each resource each process is currently holding[**ALLOCATED**]
- How much of each resource the system currently has available[**AVAILABLE**]

Resources may be allocated to a process only if the amount of resources requested is less than or equal to the amount available; otherwise, the process waits until resources are available.

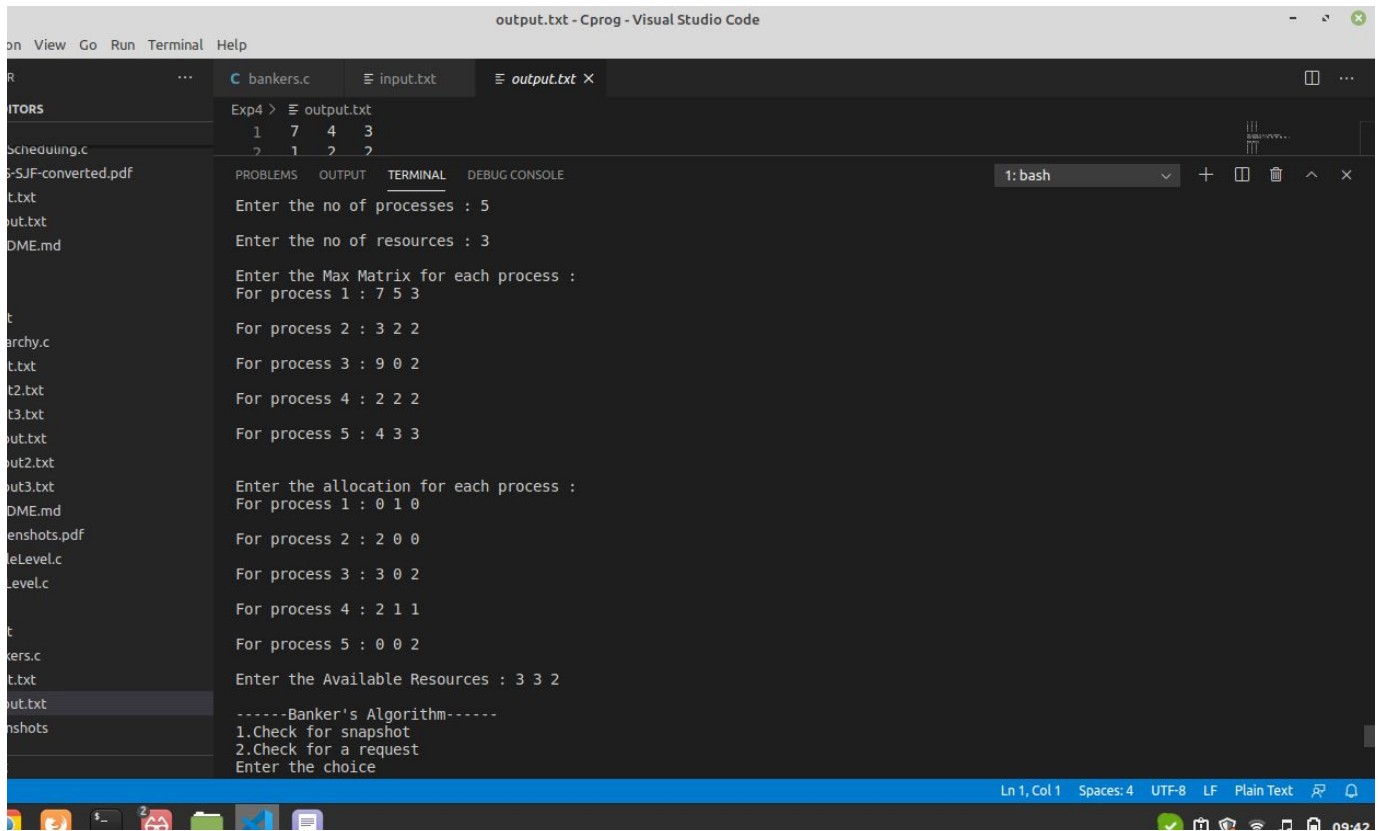
Some of the resources that are tracked in real systems are **memory**, **semaphores** and **interface** access.

The Banker's Algorithm derives its name from the fact that this algorithm could be used in a banking system to ensure that the bank does not run out of resources, because the bank would never allocate its money in such a way that it can no longer satisfy the needs of all its customers^[2]. By using the Banker's algorithm, the bank ensures that when customers request money the bank never leaves a safe state. If the customer's request does not cause the bank to leave a safe state, the cash will be allocated, otherwise the customer must wait until some other customer deposits enough.

RESULT

Banker's algorithm was implemented for m process and n resources and the output were written to the output file.

Output Screenshots



The screenshot shows a Visual Studio Code window with a terminal running a C program. The terminal output is as follows:

```
Exp4 > output.txt
1 7 4 3
2 1 2 2

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: bash

Enter the no of processes : 5

Enter the no of resources : 3

Enter the Max Matrix for each process :
For process 1 : 7 5 3

For process 2 : 3 2 2

For process 3 : 9 0 2

For process 4 : 2 2 2

For process 5 : 4 3 3

Enter the allocation for each process :
For process 1 : 0 1 0

For process 2 : 2 0 0

For process 3 : 3 0 2

For process 4 : 2 1 1

For process 5 : 0 0 2

Enter the Available Resources : 3 3 2

-----Banker's Algorithm-----
1.Check for snapshot
2.Check for a request
Enter the choice
```

The status bar at the bottom indicates the current position is Ln 1, Col 1, with 4 spaces, using UTF-8 encoding, LF line endings, and Plain Text format. The system clock shows 09:42.

output.txt - Cprog - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

CPROG

CPUScheduling.c

FCFS-SJF-converted.pdf

input.txt

output.txt

README.md

Exp2

Exp3

a.out

heirarchy.c

input.txt

input2.txt

input3.txt

output.txt

output2.txt

output3.txt

README.md

Screenshots.pdf

singleLevel.c

twoLevel.c

Exp4

a.out

bankers.c

input.txt

output.txt

screenshots

a.out

OUTLINE

Exp4 > output.txt

```
1 7 4 3
2 1 2 2
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

1: bash

-----Banker's Algorithm-----

1.Check for snapshot

2.Check for a request

Enter the choice

1

Need Matrix

7	4	3
1	2	2
6	0	0
0	1	1
4	3	1

The system is in a safe state!!

Safe Sequence : < P1 P3 P4 P0 P2 >

Do you want to continue? :1

-----Banker's Algorithm-----

1.Check for snapshot

2.Check for a request

Enter the choice

2

Enter the process for request: 1

Enter the request: 1 0 2

Need Matrix

7	4	3
0	2	0
6	0	0
0	1	1
4	3	1

The system is in a safe state!!

Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text 09:42

output.txt - Cprog - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

CPROG

CPUScheduling.c

FCFS-SJF-converted.pdf

input.txt

output.txt

README.md

Exp2

Exp3

a.out

heirarchy.c

input.txt

input2.txt

input3.txt

output.txt

output2.txt

output3.txt

README.md

Screenshots.pdf

singleLevel.c

twoLevel.c

Exp4

a.out

bankers.c

input.txt

output.txt

screenshots

a.out

OUTLINE

Exp4 > output.txt

```
1 7 4 3
2 1 2 2
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

1: bash

2

Enter the process for request: 1

Enter the request: 1 0 2

Need Matrix

7	4	3
0	2	0
6	0	0
0	1	1
4	3	1

The system is in a safe state!!

Safe Sequence : < P1 P3 P4 P0 P2 >

Do you want to continue? :1

-----Banker's Algorithm-----

1.Check for snapshot

2.Check for a request

Enter the choice

2

Enter the process for request: 4

Enter the request: 3 3 0

Need Matrix

7	4	3
1	2	2
6	0	0
0	1	1
1	0	1

The system is in an unsafe state

Do you want to continue? :0

navaneeth@navaneeth-Lap:~/Documents/NAV/Cprog/Exp4\$

Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text 09:42