King Saud University

College of Computer and Information Sciences

Information Technology Department

Operating Systems | CSC 227

Memory Fragmentation Simulation

Semester-2, 1446H
Section #44075

Name student	id
Rana alotebe	444201191
Lena Alsabi	444200854
Zinah Anbar	443204590
Areen Aljarbou	444200820
Hessa Alhozaimy	44200799

Instructor: Abeer Ibrahim Alshaya

Table of Contents

Task Distribution

Student Name	Task
Areen Aljarbou	Memory Initialization (Prompt user, create blocks, set attributes)
Lena Alsabi	Allocation Algorithms (First-Fit, Best-Fit, Worst-Fit)
Hessa Alhozaimy	Menu System & User Interaction (Allocate, Deallocate, Exit)
Zinah Anbar	Internal Fragmentation Handling (Calculate & Display)
Rana alotebe	Report, Testing, Screenshots, ReadMe PDF

Description

In this project, a simple memory management simulator was built using three allocation strategies: **First Fit**, **Best Fit**, and **Worst Fit**.

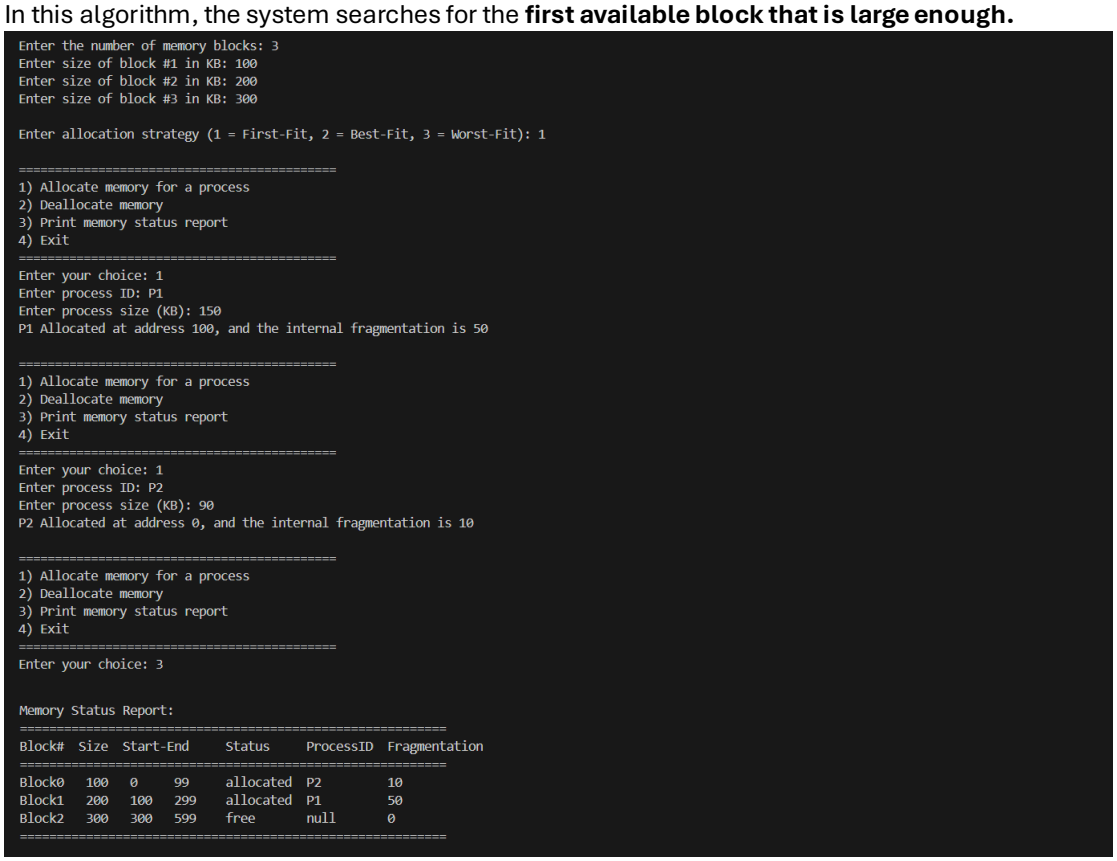
Memory Allocation Strategies:

- **First Fit**: Memory allocation is based on the first available partition (or block) that is larger than or equal to the process size. This strategy is the fastest among the others because it selects and allocates the first suitable block immediately.
- **Best Fit:** Memory allocation involves scanning all the empty partitions and selecting the one with the least fragmentation (i.e., the smallest partition that can accommodate the process). This strategy does not waste free space in memory, but it is time-consuming because it requires checking all available blocks.
- **Worst Fit**: Similar to the First Fit strategy, but instead of selecting the first suitable block, the system searches for the largest available block that can accommodate the process. This strategy is also time-consuming due to the need to search for the largest block.

Screenshot of The Input and Output

First Fit algorithm:

In this algorithm, the system searches for the **first available block that is large enough.**

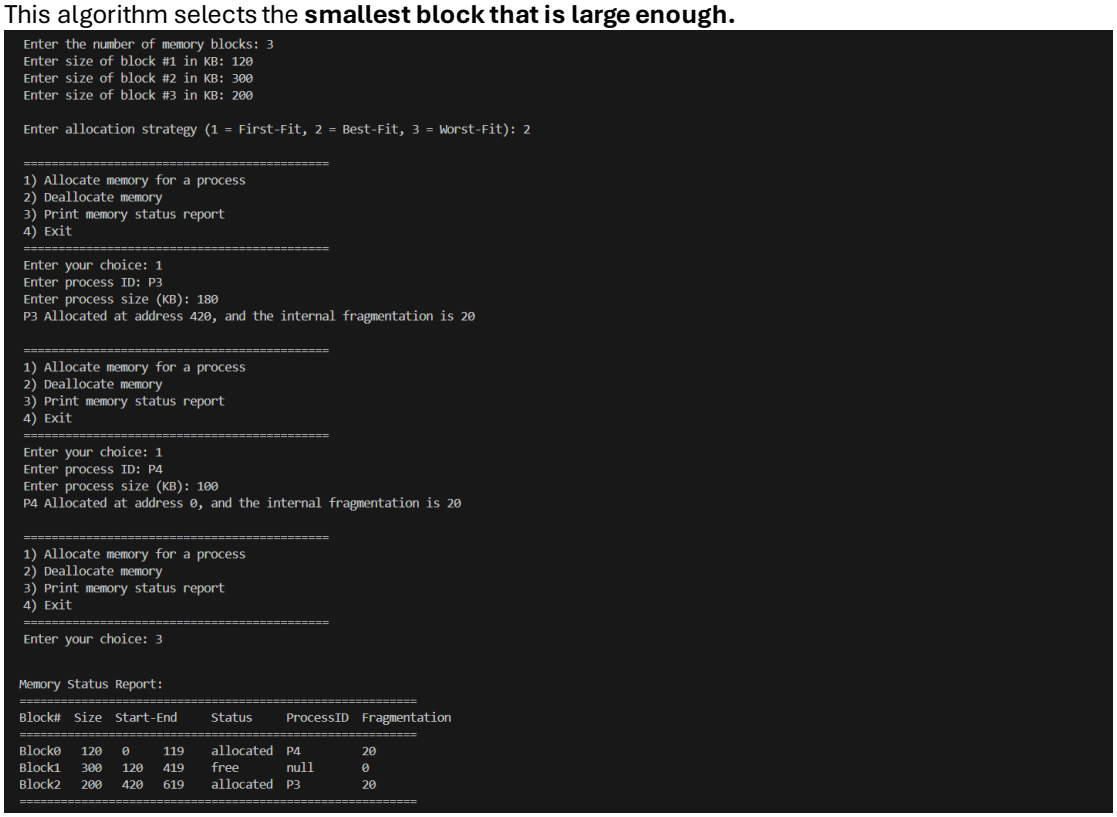


First-Fit memory allocation strategy with 3 initialized blocks:

- Block 0: *100 KB*
- Block 1: *200 KB*
- Block 2: *300 KB.*
- *Process P1 (150 KB) was allocated to the second block (200 KB) with 50 KB internal fragmentation.*
- *Process p2 (90 KB) was allocated to the first block (100 KB) with 10 KB internal fragmentation.*
- *The memory status report shows updated block states:*
    - *Block 0 → allocated to p2*
    - *Block 1 → allocated to P1*
    - *Block 2 → still free and available*

Best Fit algorithm:

This algorithm selects the **smallest block that is large enough.**



```
Enter the number of memory blocks: 3
Enter size of block #1 in KB: 120
Enter size of block #2 in KB: 300
Enter size of block #3 in KB: 200

Enter allocation strategy (1 = First-Fit, 2 = Best-Fit, 3 = Worst-Fit): 2

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 1
Enter process ID: P3
Enter process size (KB): 180
P3 Allocated at address 420, and the internal fragmentation is 20

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 1
Enter process ID: P4
Enter process size (KB): 100
P4 Allocated at address 0, and the internal fragmentation is 20

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 3


Memory Status Report:
=====================================================
Block#  Size  Start-End     Status     ProcessID  Fragmentation
=====================================================
Block0   120   0     119    allocated  P4         20
Block1   300   120   419    free       null       0
Block2   200   420   619    allocated  P3         20
=====================================================
```

Best-Fit memory allocation strategy with 3 initialized blocks:

- Block 0: 120 KB
- Block 1: 300 KB
- Block 2: 200 KB

Processes were allocated as follows using the Best-Fit approach:

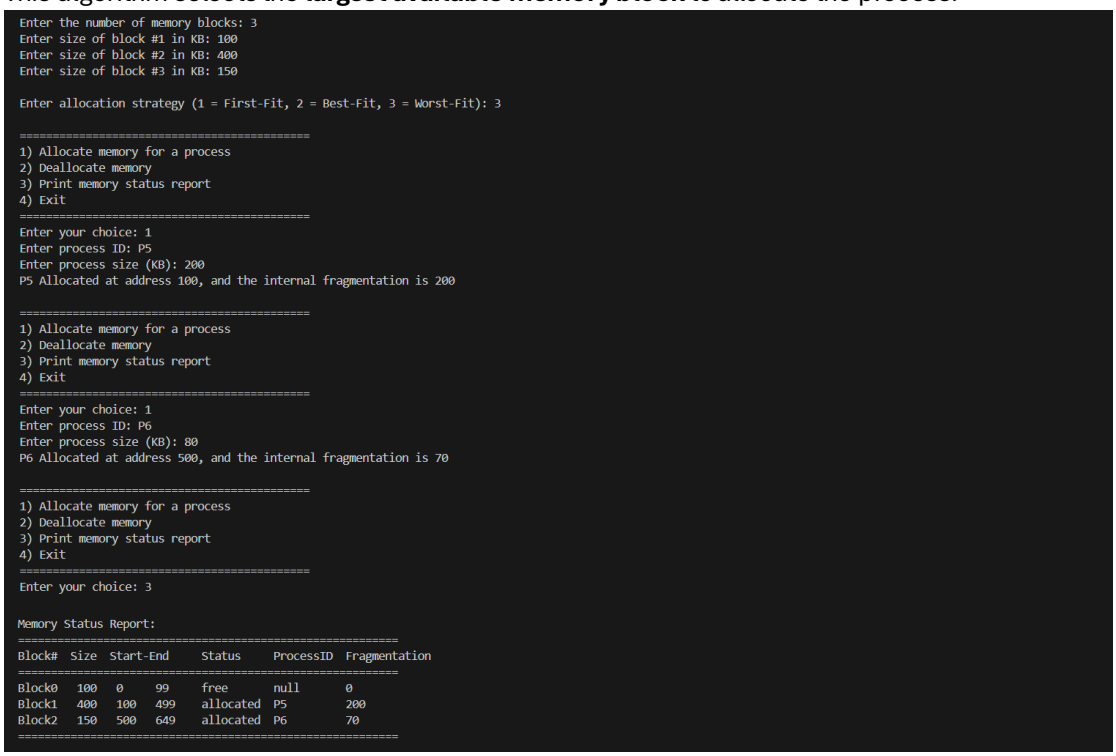
- Process P3 (180 KB) was allocated to Block 2 (200 KB), resulting in 20 KB internal fragmentation.
- Process P4 (100 KB) was allocated to Block 0 (120 KB), resulting in 20 KB internal fragmentation.

The final memory report confirms:

- Block 0 → allocated to P4
- Block 1 → still free
- Block 2 → allocated to P3

Worst fit algorithm:

This algorithm selects the **largest available memory block** to allocate the process.



```
Enter the number of memory blocks: 3
Enter size of block #1 in KB: 100
Enter size of block #2 in KB: 400
Enter size of block #3 in KB: 150

Enter allocation strategy (1 = First-Fit, 2 = Best-Fit, 3 = Worst-Fit): 3

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 1
Enter process ID: P5
Enter process size (KB): 200
P5 Allocated at address 100, and the internal fragmentation is 200

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 1
Enter process ID: P6
Enter process size (KB): 80
P6 Allocated at address 500, and the internal fragmentation is 70

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 3

Memory Status Report:
=============================================================
Block#  Size  Start-End     Status     ProcessID  Fragmentation
=============================================================
Block0   100   0     99     free       null       0
Block1   400   100   499    allocated  P5         200
Block2   150   500   649    allocated  P6         70
=============================================================
```

Worst-Fit memory allocation strategy with 3 memory blocks:

- Block 0: 100 KB
- Block 1: 400 KB
- Block 2: 150 KB

Allocation results using Worst-Fit:

- Process P5 (200 KB) was allocated to the largest available block (400 KB → Block 1), causing 200 KB internal fragmentation.
- Process P6 (80 KB) was allocated to the next largest block (150 KB → Block 2), with 70 KB internal fragmentation.

The final memory status report shows:

- Block 0 → still free
- Block 1 → allocated to P5
- Block 2 → allocated to P6

Successful Memory Deallocation:

```
Memory Status Report:
=======================================================
Block#  Size  Start-End     Status     ProcessID  Fragmentation
=======================================================
Block0   100   0     99     free       null       0
Block1   400   100   499    allocated  P5         200
Block2   150   500   649    allocated  P6         70
=======================================================

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 2
Enter process ID to deallocate: P5
Process P5 has been deallocated.

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 3

Memory Status Report:
=======================================================
Block#  Size  Start-End     Status     ProcessID  Fragmentation
=======================================================
Block0   100   0     99     free       null       0
Block1   400   100   499    free       null       0
Block2   150   500   649    allocated  P6         70
=======================================================
```

Process P5 (200 KB) was deallocated from Block 1.
Final memory report shows Block 1 is now free,
while Block 2 remains allocated to P6, and Block 0 is also free.

Attempt to Deallocate Non-Existent Process:

```
Memory Status Report:
=======================================================
Block#  Size  Start-End     Status     ProcessID  Fragmentation
=======================================================
Block0   100   0     99     free       null       0
Block1   400   100   499    free       null       0
Block2   150   500   649    allocated  P6         70
=======================================================

========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
========================================
Enter your choice: 2
Enter process ID to deallocate: P5
Error: Process ID not found.

========================================
```

Memory Allocation Failed – Process Exceeds All Available Blocks:

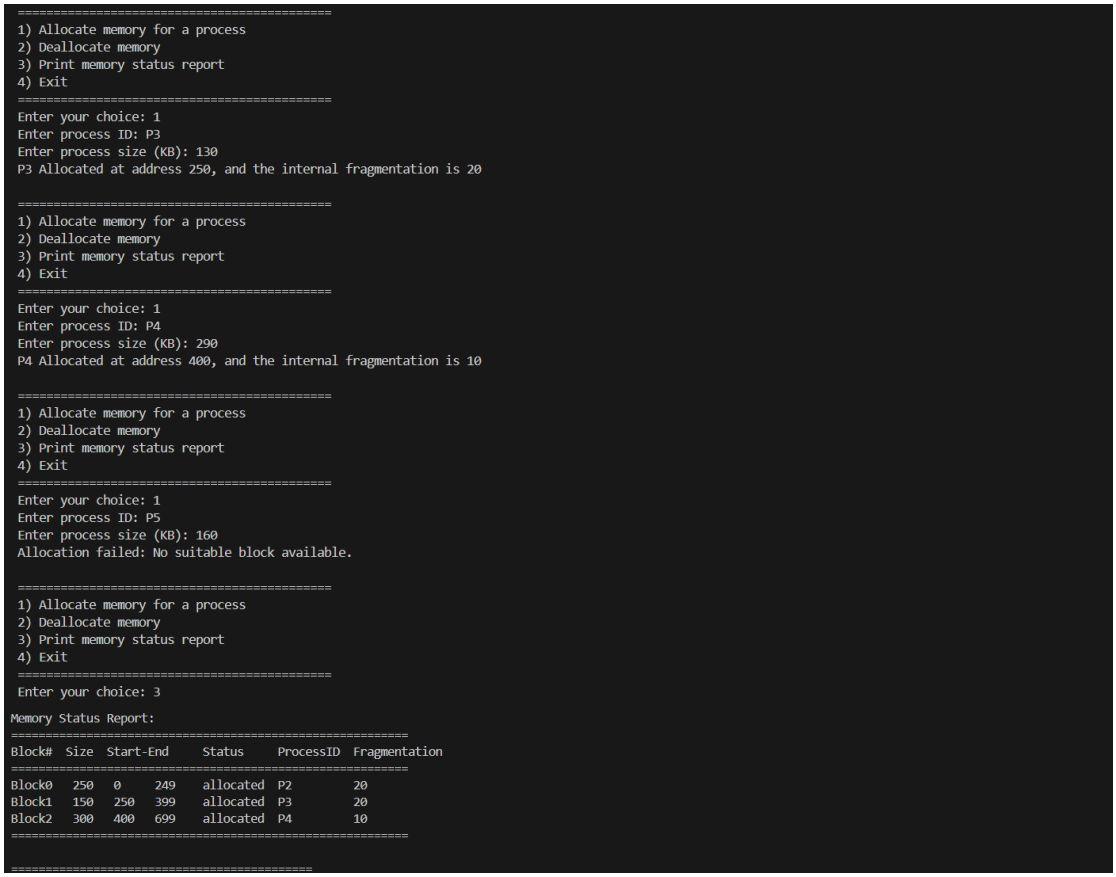
```
==========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
==========================================
Enter your choice: 1
Enter process ID: P7
Enter process size (KB): 600
Allocation failed: No suitable block available.

==========================================
```

All available blocks are too small to accommodate the request.

Reallocation and Allocation Failure (Best-Fit):

This test case shows how the system can reuse a free memory block and handle failed allocation when no block is large enough.

```
Enter the number of memory blocks: 3
Enter size of block #1 in KB: 250
Enter size of block #2 in KB: 150
Enter size of block #3 in KB: 300

Enter allocation strategy (1 = First-Fit, 2 = Best-Fit, 3 = Worst-Fit): 2

==========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
==========================================
Enter your choice: 1
Enter process ID: P1
Enter process size (KB): 140
P1 Allocated at address 250, and the internal fragmentation is 10

==========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
==========================================
Enter your choice: 1
Enter process ID: P2
Enter process size (KB): 230
P2 Allocated at address 0, and the internal fragmentation is 20

==========================================
1) Allocate memory for a process
2) Deallocate memory
3) Print memory status report
4) Exit
==========================================
Enter your choice: 2
Enter process ID to deallocate: P1
Process P1 has been deallocated.

==========================================
1) Allocate memory for a process
```



Best-Fit failed to allocate **P5 (160 KB)** because **no free block** was large enough.
All blocks were already allocated, so the algorithm had **no options to choose from**.

Evaluation

Functional Requirements			
	Criteria	**Evaluation**	**All team members**
Implementation	Overall quality of the code implementation (organization, clearness, design…)	.5	
	Create and initialize appropriate Memory and blocks.	1	
	Logic for first-fit.	1	
	Logic for best-fit.	1	
	Logic for worst-fit.	1	
	Logic for computing fragmentations.	1	
	Displays the desired output on the console correctly and completely.	1	
Deliverables	1. Program code	.5	
	2. A Read Me file in PDF	.5	
	Total	**7.5**	