Memory Allocation using Segmentation

1601626	نيفين نبيل شكرى زكى
1700078	أحمد سيد عابد أحمد
1300151	أحمد عصام الدين أحمد الموجى
1700253	اسلام محمد عبد العزيز عبد العال
1700258	اسلام هشام محمد ابو الفضل
1200427	باسم حسين فوزى ياقوت

Contents

Preface

This report demonstrates the technique of memory allocation using segmentation through various examples done on the presented software.

The following examples illustrate the different styles of segmentation allocation (first fit, best fit, worst fit), two examples for each algorithm, as well as using compaction. In addition to de-allocation of processes and combining adjacent holes.

The report also covers the different error messages that emerges on different occasions.

The Link to the executable file

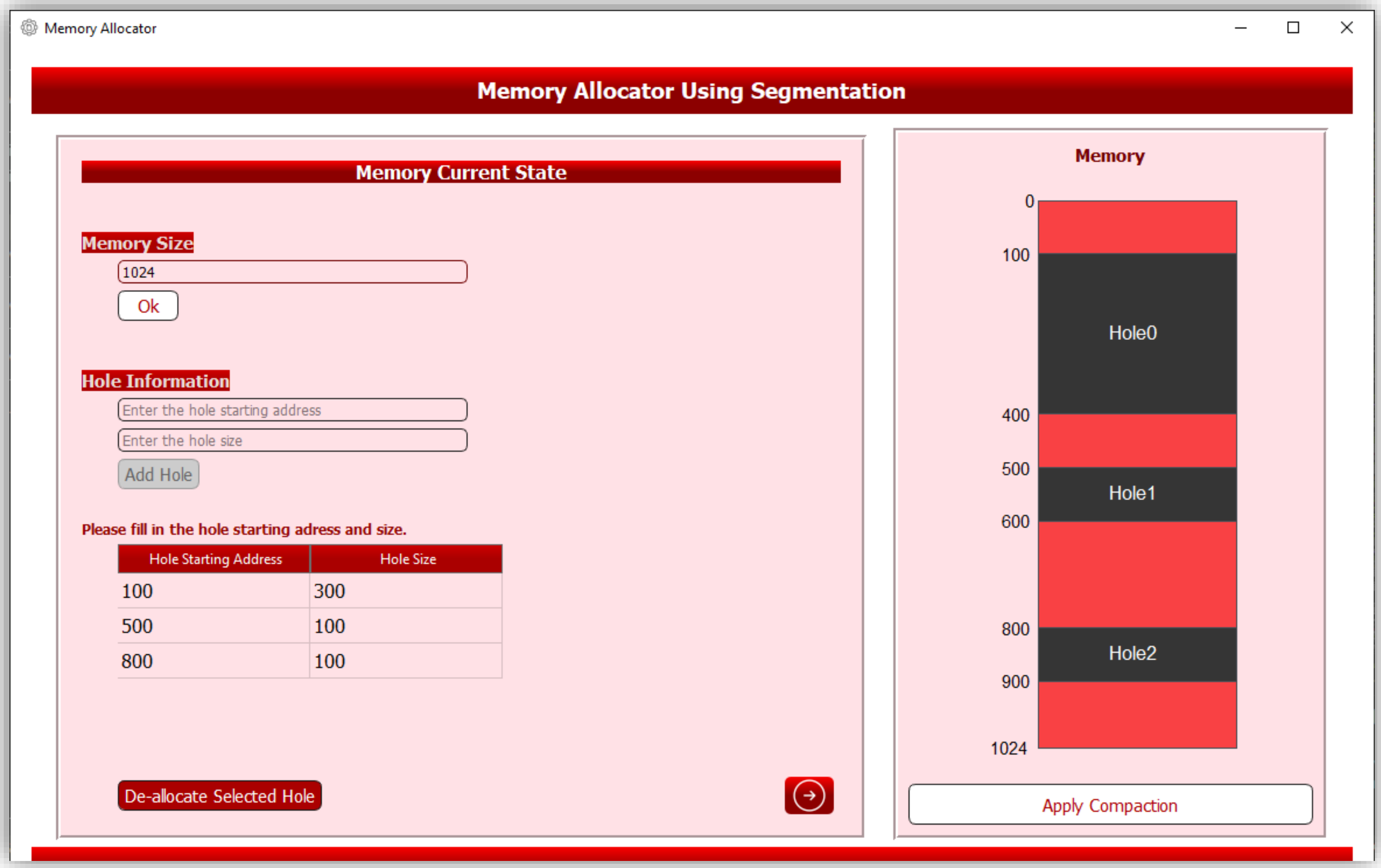
https://github.com/NevenNabil/Memory-Allocation-By-Segmentation/tree/master

Download the code files then click on 'Memory_alloc.exe'

1. Entering the current state of the memory

1.1 Memory size and holes as inputs

The user starts by entering the memory size, and the current state of the memory using holes' starting addresses and sizes as inputs.



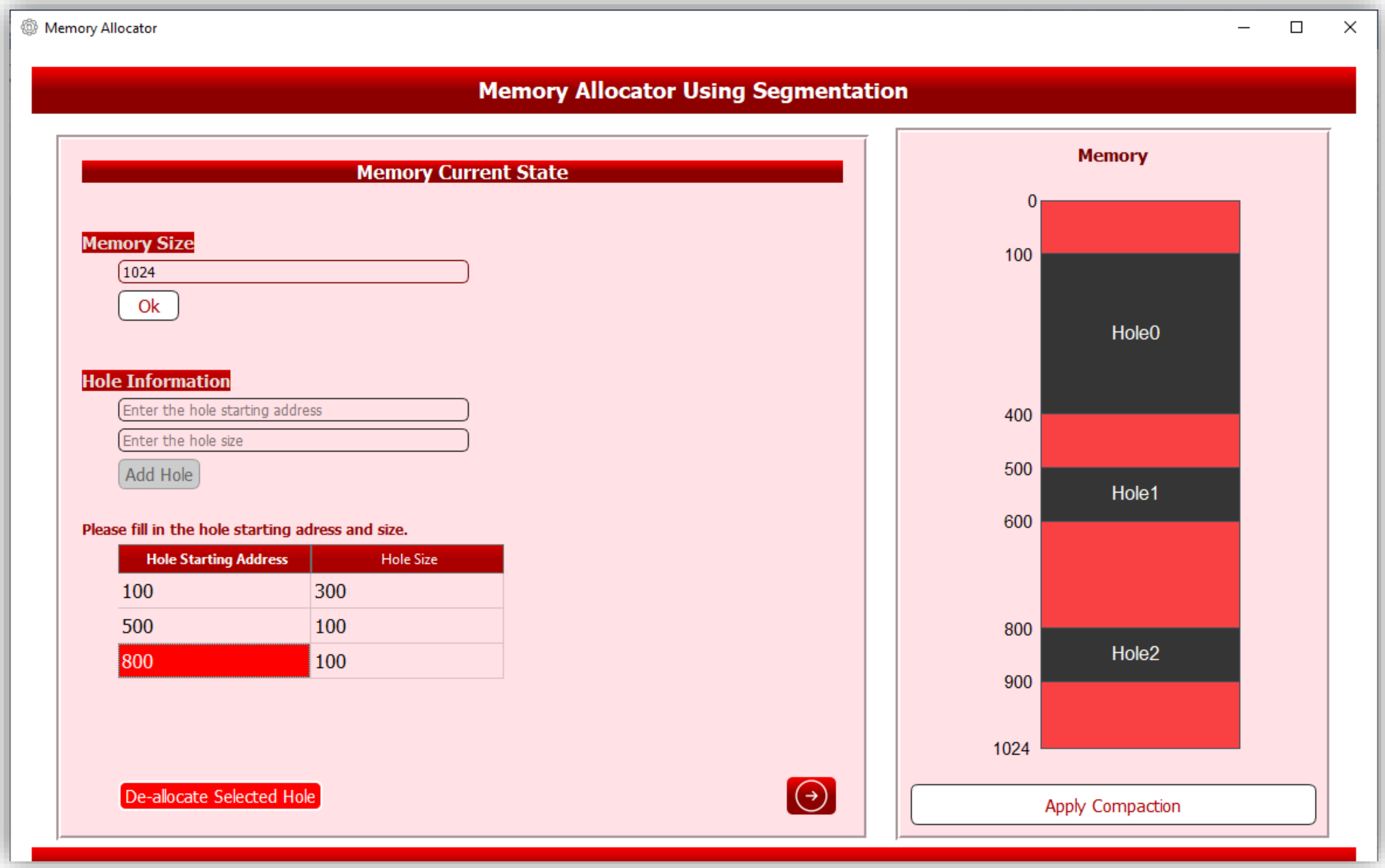
Please fill in the hole starting adress and size.

Hole Starting Address	Hole Size
100	300
500	100
800	100

Figure 1, Memory size and holes as inputs

1.2 De-allocate selected holes

The user has the option at this point to de-allocate selected holes to easily shape the memory before entering processes.



Memory Allocator

Memory Allocator Using Segmentation

Memory Current State

Memory Size

1024

Ok

Hole Information

Enter the hole starting address

Enter the hole size

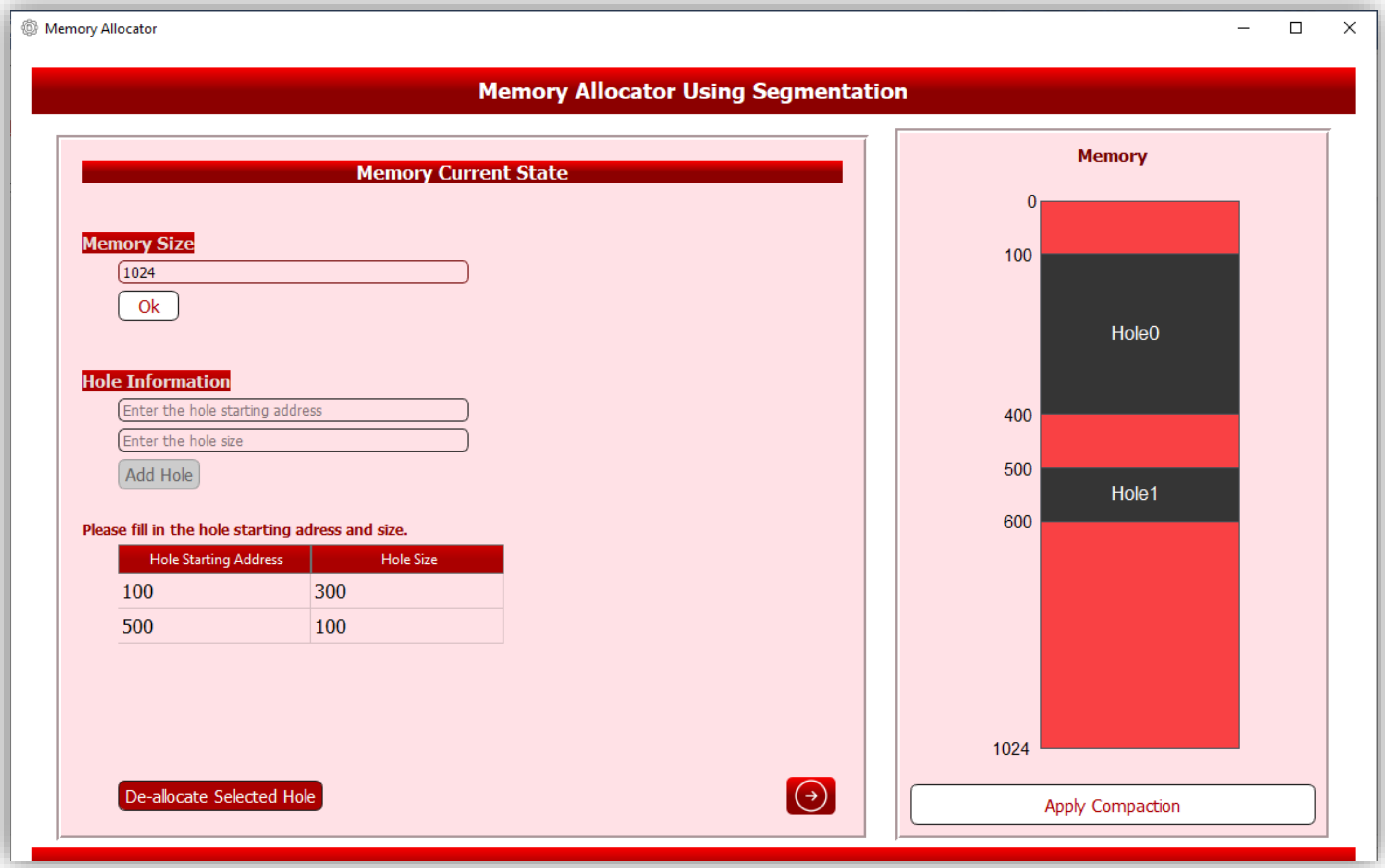
Add Hole

Please fill in the hole starting adress and size.

Hole Starting Address	Hole Size
100	300
500	100
800	100

De-allocate Selected Hole

Figure 2, First, select the hole



Please fill in the hole starting adress and size.

Hole Starting Address	Hole Size
100	300
500	100

Figure 3, Then, click on "De-allocate Selected Hole"

1.3 Recognizing old processes

The software then recognizes old processes based on holes' locations when the user navigates to the next page; where he/she is meant to enter processes' information and algorithm to be used.

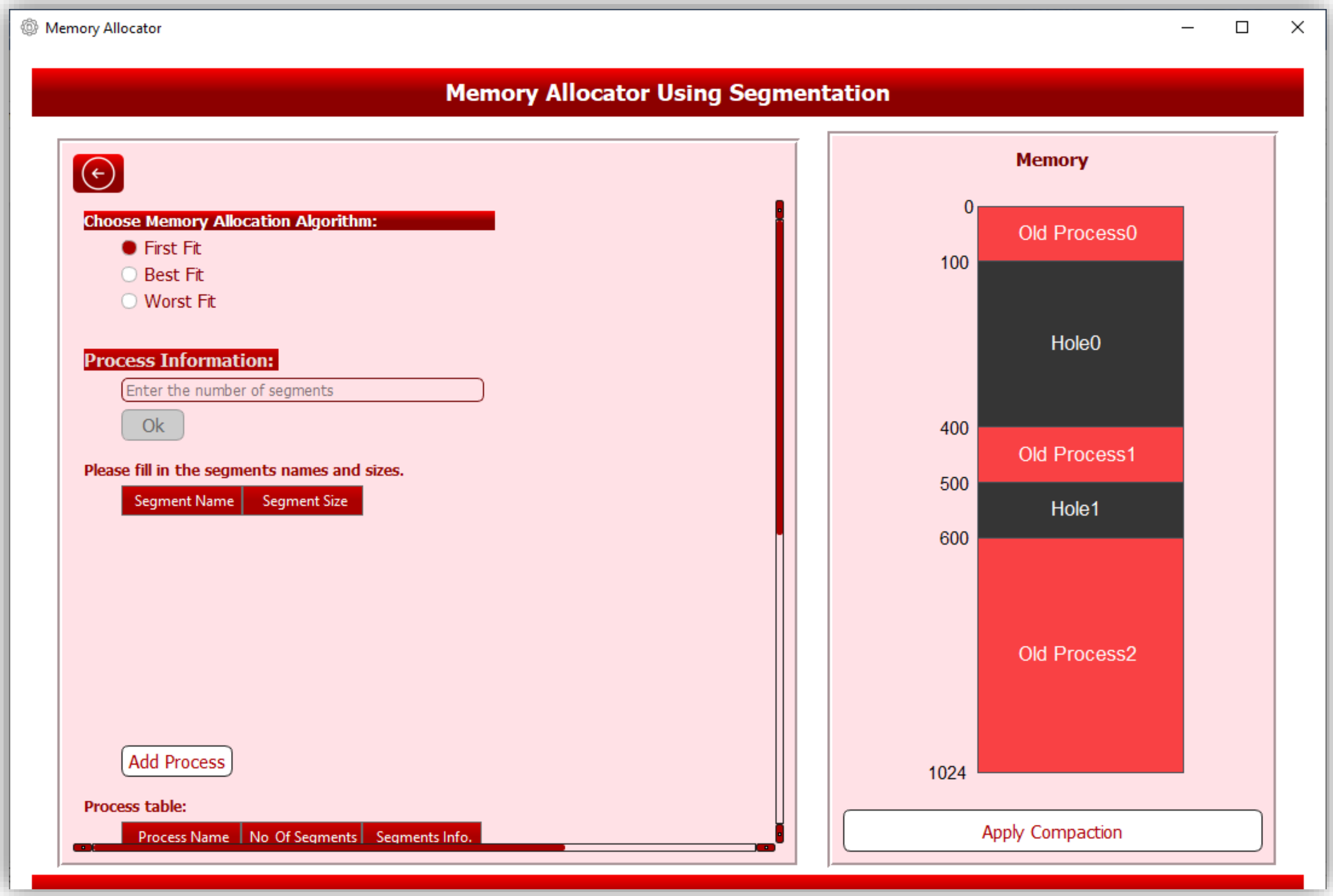


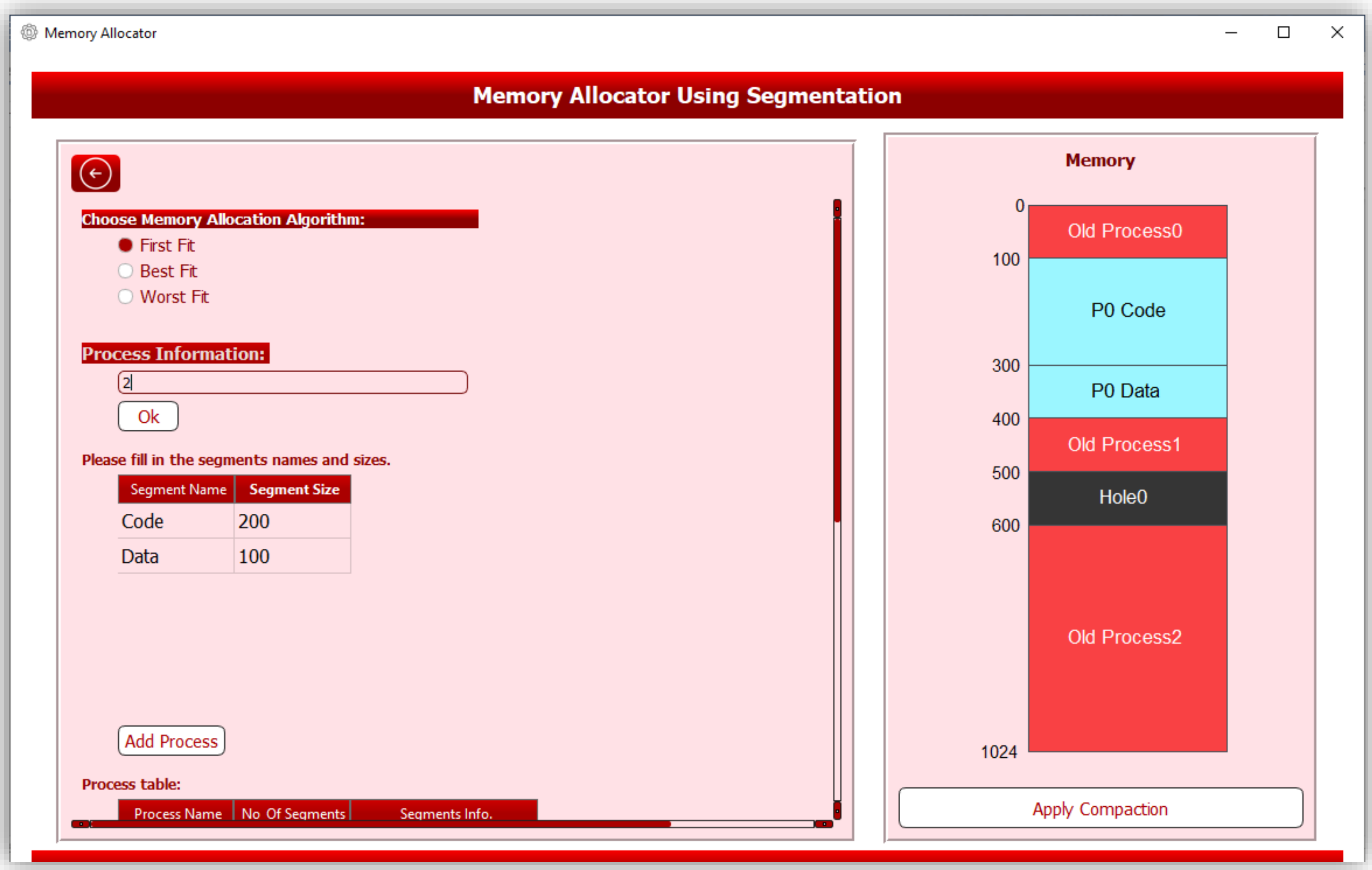
Figure 4, Next page, and old processes are recognized

2. Segmentation algorithms examples

The user chooses the allocation algorithm, enters the process number of segments along with each segment's name and size.

2.1 First fit

Example 1, The previous memory is populated with new processes' segments based on the first-fit algorithm.



Memory Allocator

Memory Allocator Using Segmentation

Choose Memory Allocation Algorithm:

[✓ First Fit] Best Fit Worst Fit

Process Information:

2

Please fill in the segments names and sizes.

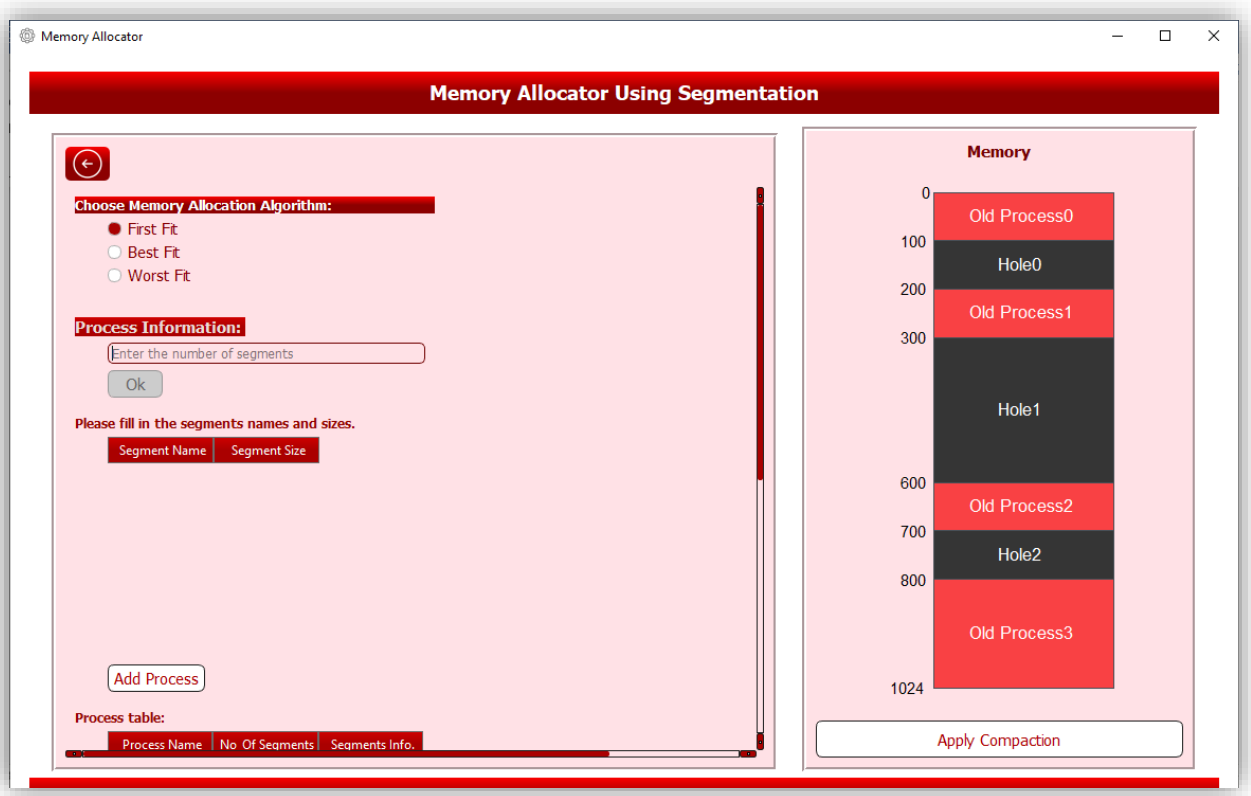
Segment Name	Segment Size
Code	200
Data	100

Process table:

Process Name	No Of Segments	Segments Info.

Figure 5, Population of memory based on the first-fit algorithm

Example 2, before and after populating memory using first fit:


Memory Allocator

Memory Allocator Using Segmentation

Choose Memory Allocation Algorithm:

[✓ First Fit] Best Fit Worst Fit

Process Information:

Enter the number of segments

Ok

Please fill in the segments names and sizes.

Segment Name	Segment Size

Add Process

Process table:

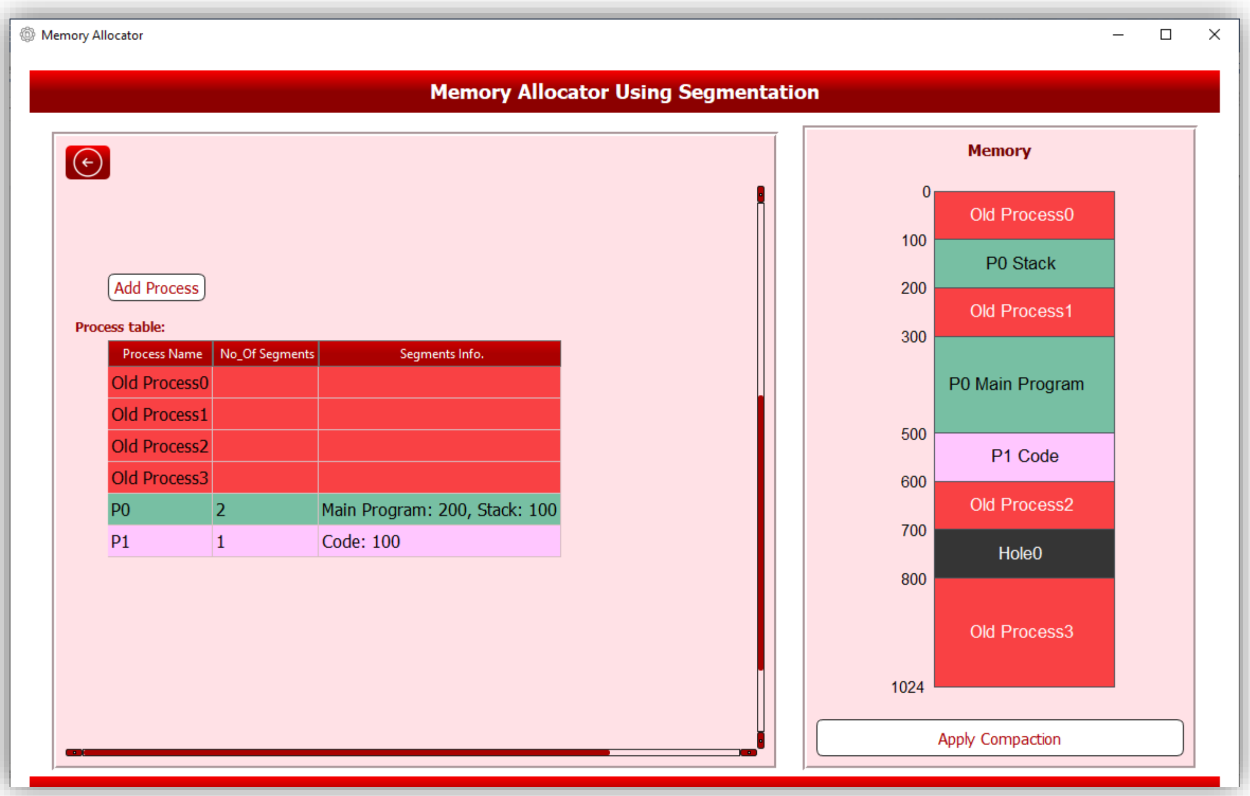
Process Name	No_Of Segments	Segments Info.

Memory

0	Old Process0
100	Hole0
200	Old Process1
300	Hole1
600	Old Process2
700	Hole2
800	Old Process3
1024	

Figure 6, Example 2 old processes

And, when allocating segments:


Memory Allocator

Memory Allocator Using Segmentation

Add Process

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
P0	2	Main Program: 200, Stack: 100
P1	1	Code: 100

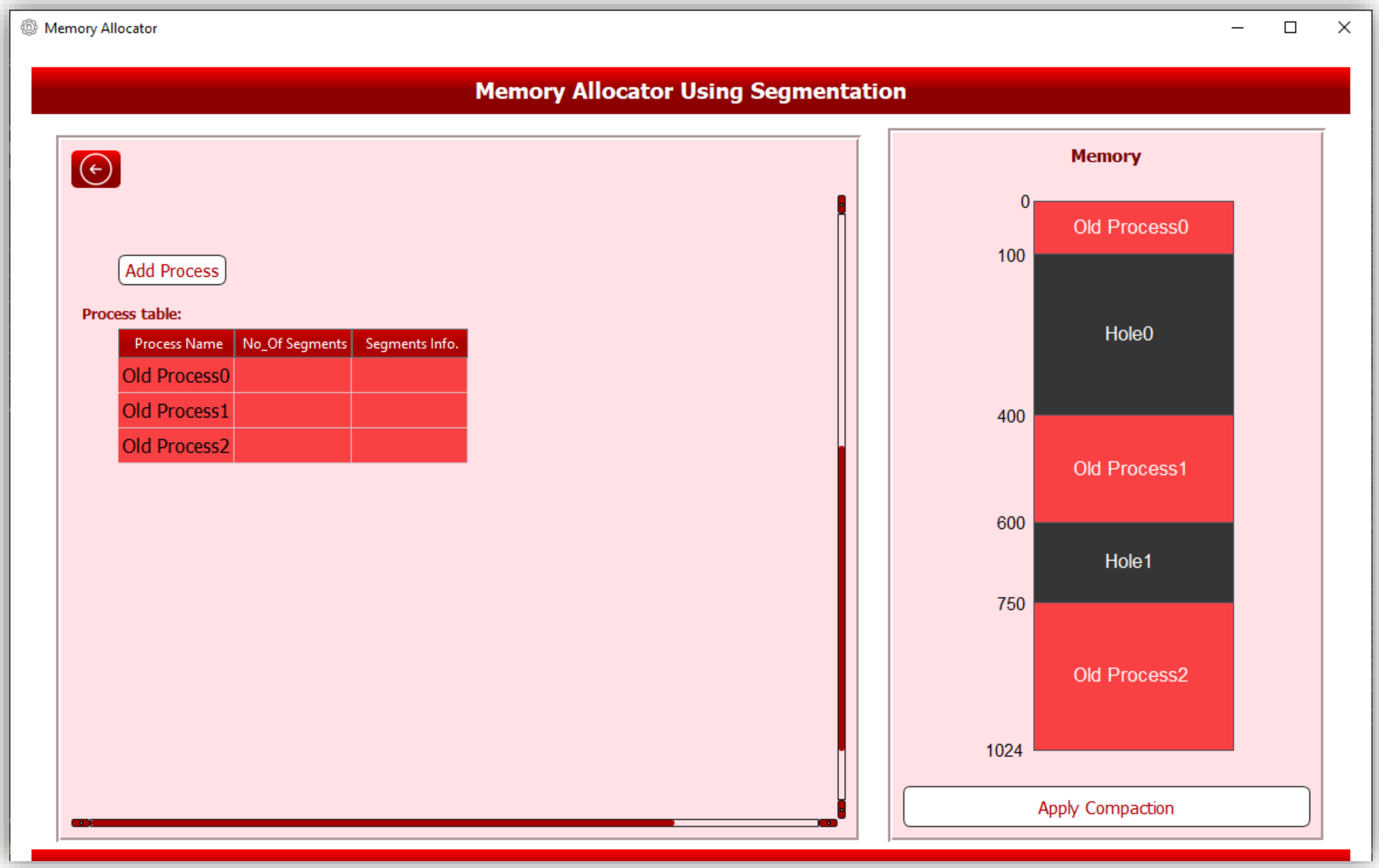
Memory

0	Old Process0
100	P0 Stack
200	Old Process1
300	P0 Main Program
500	P1 Code
600	Old Process2
700	Hole0
800	Old Process3
1024	

Figure 7, Example 2, populating memory using first fit

2.2 Best fit

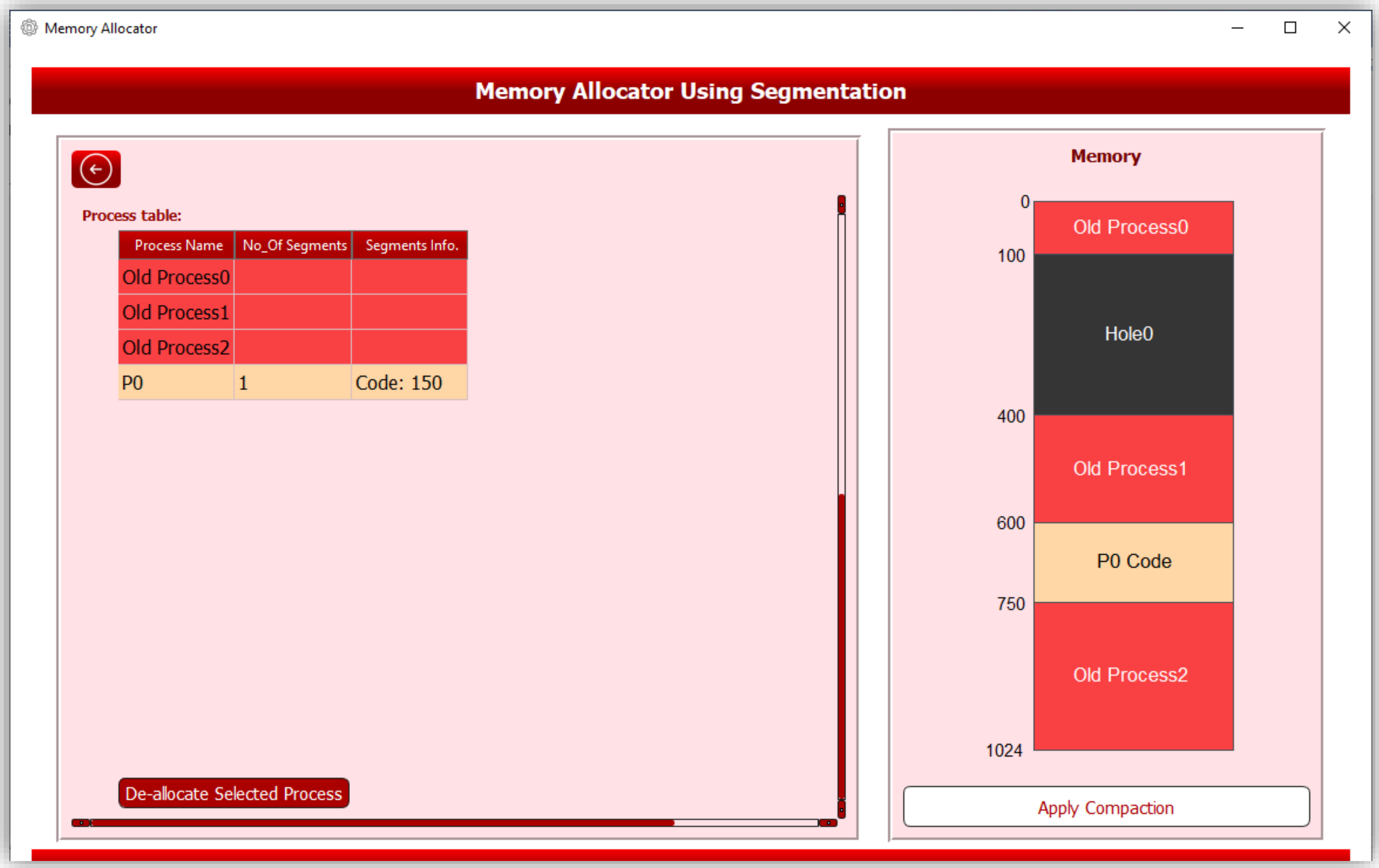
Example 1, before and after populating the memory with processes' segments based on the best-fit algorithm**:**



Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		

Figure 8, Example 1, old processes.

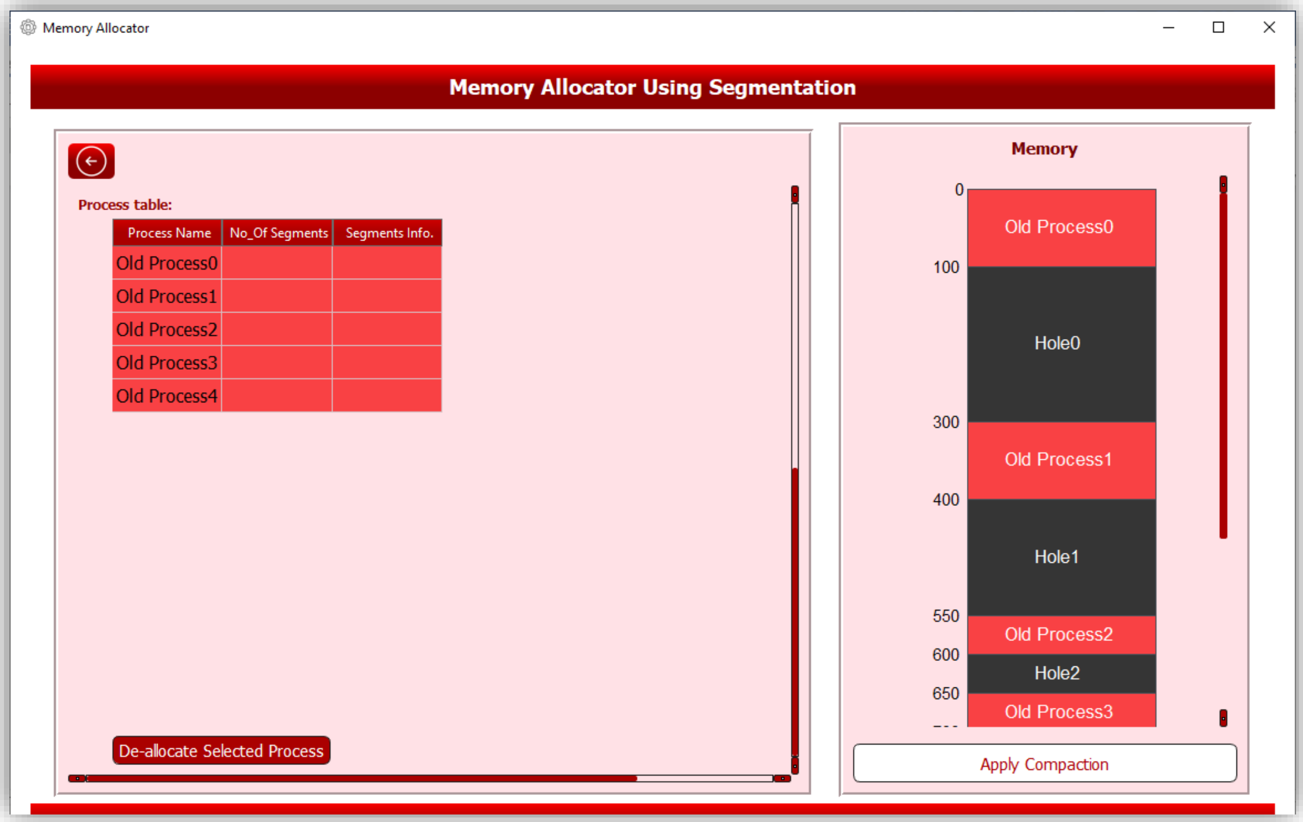


Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
P0	1	Code: 150

Figure 9, Example 1, after populating memory using best fit

Example 2, before and after populating memory using best fit:

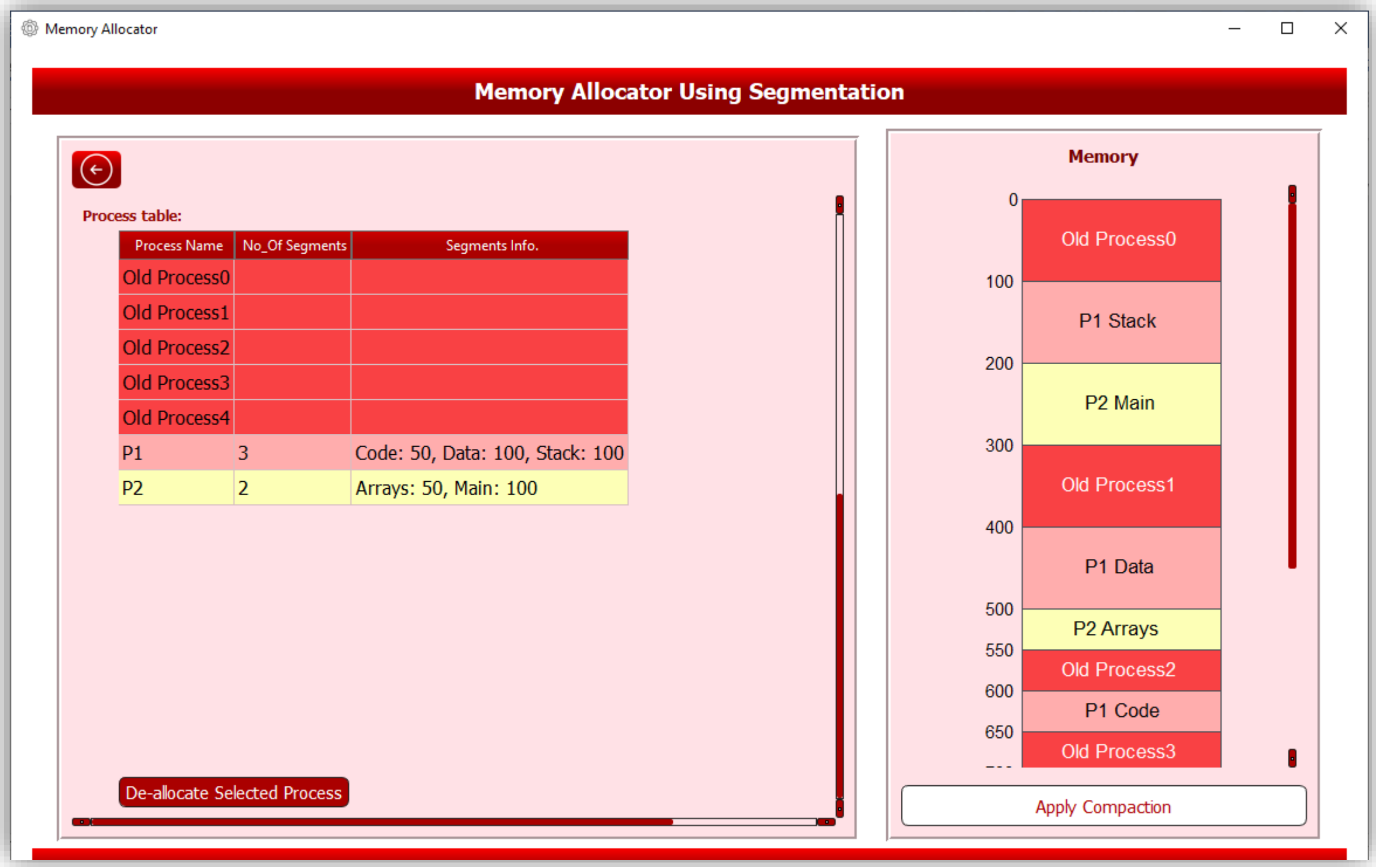

Memory Allocator

Memory Allocator Using Segmentation

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
Old Process4		

Figure 10, Example2, old processes


Memory Allocator

Memory Allocator Using Segmentation

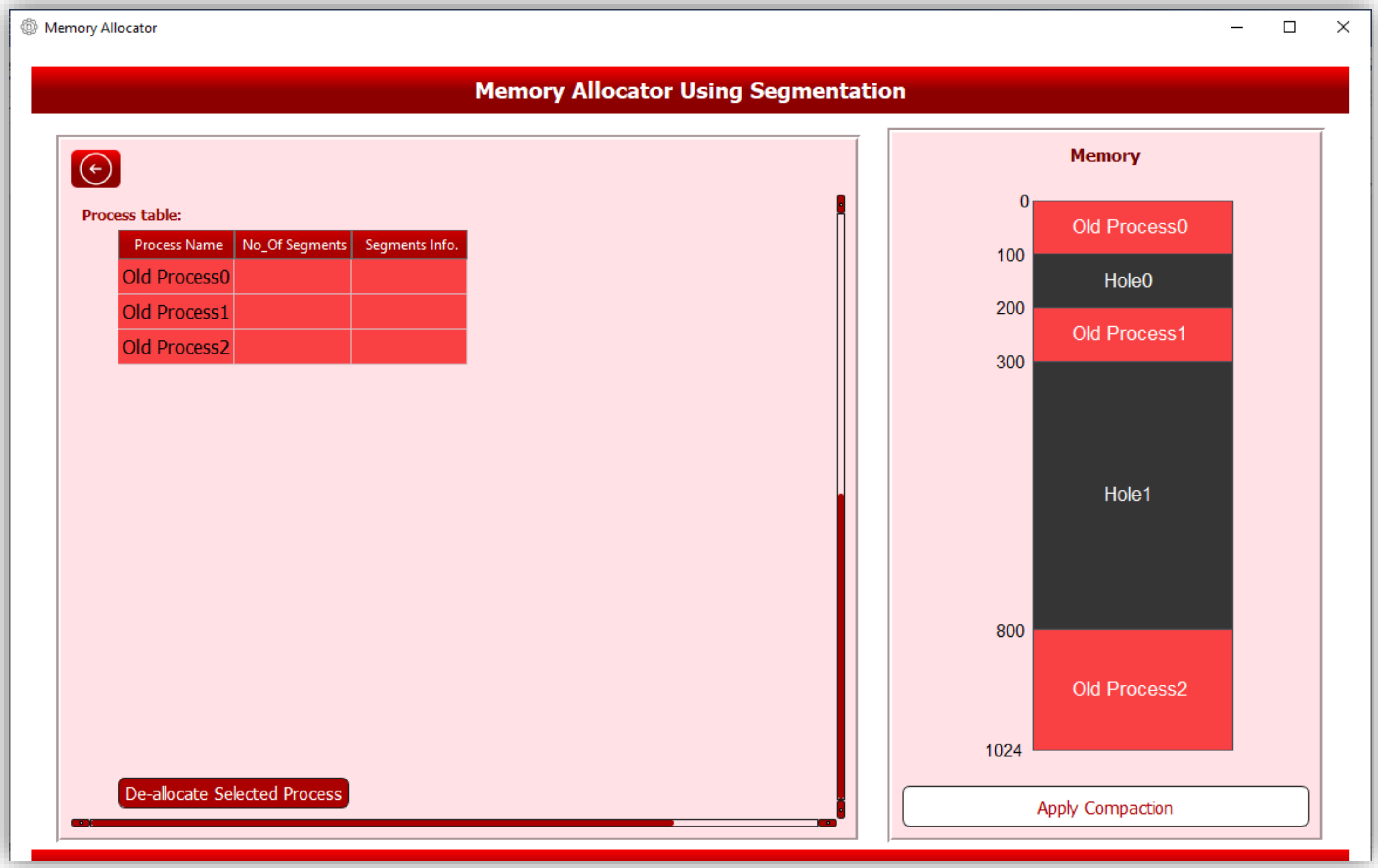
Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
Old Process4		
P1	3	Code: 50, Data: 100, Stack: 100
P2	2	Arrays: 50, Main: 100

Figure 11, Example 2, populating memory using best fit

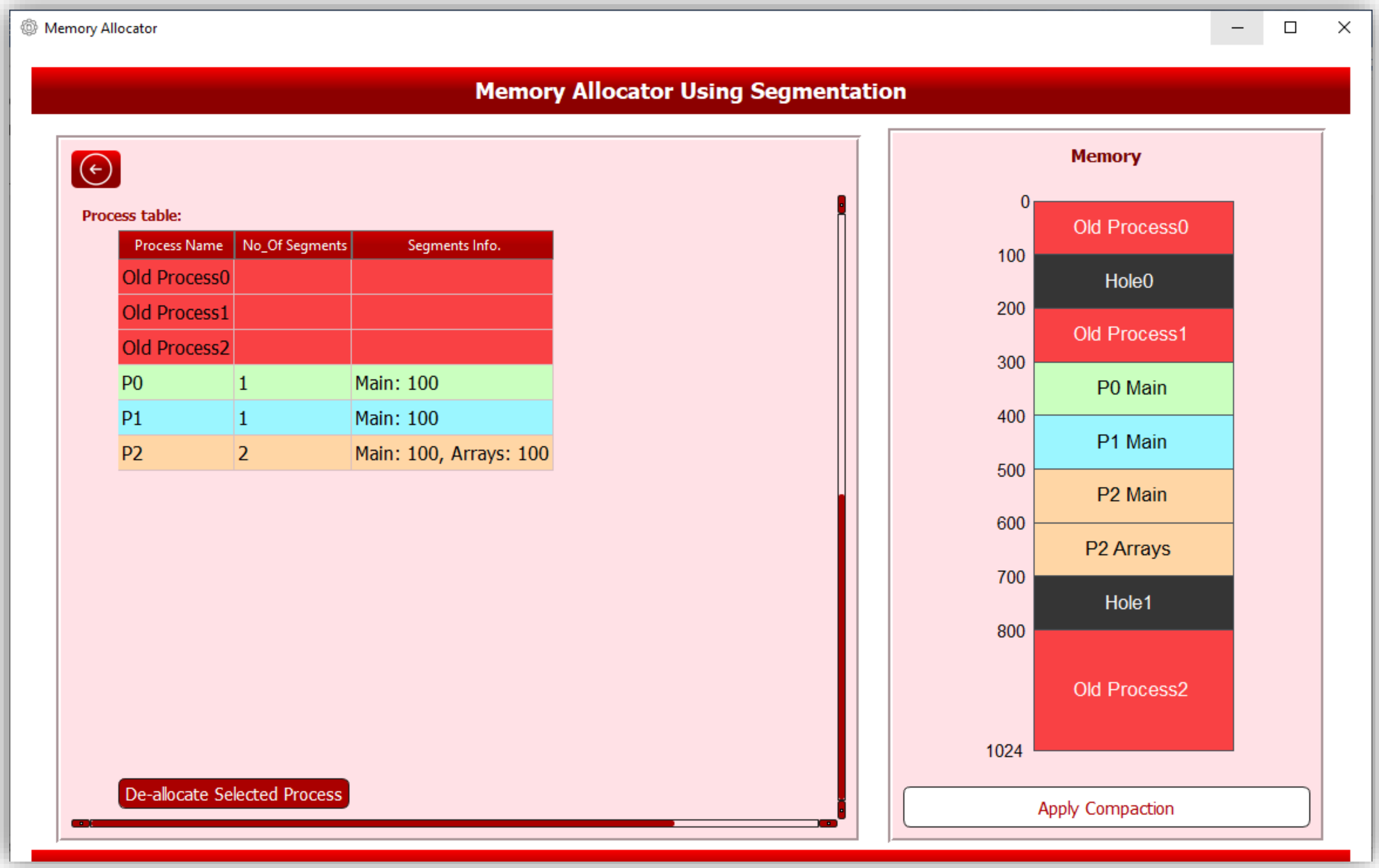
2.3 Worst fit

Example 1, before and after populating the memory with processes' segments based on the worst-fit algorithm**:**



Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		

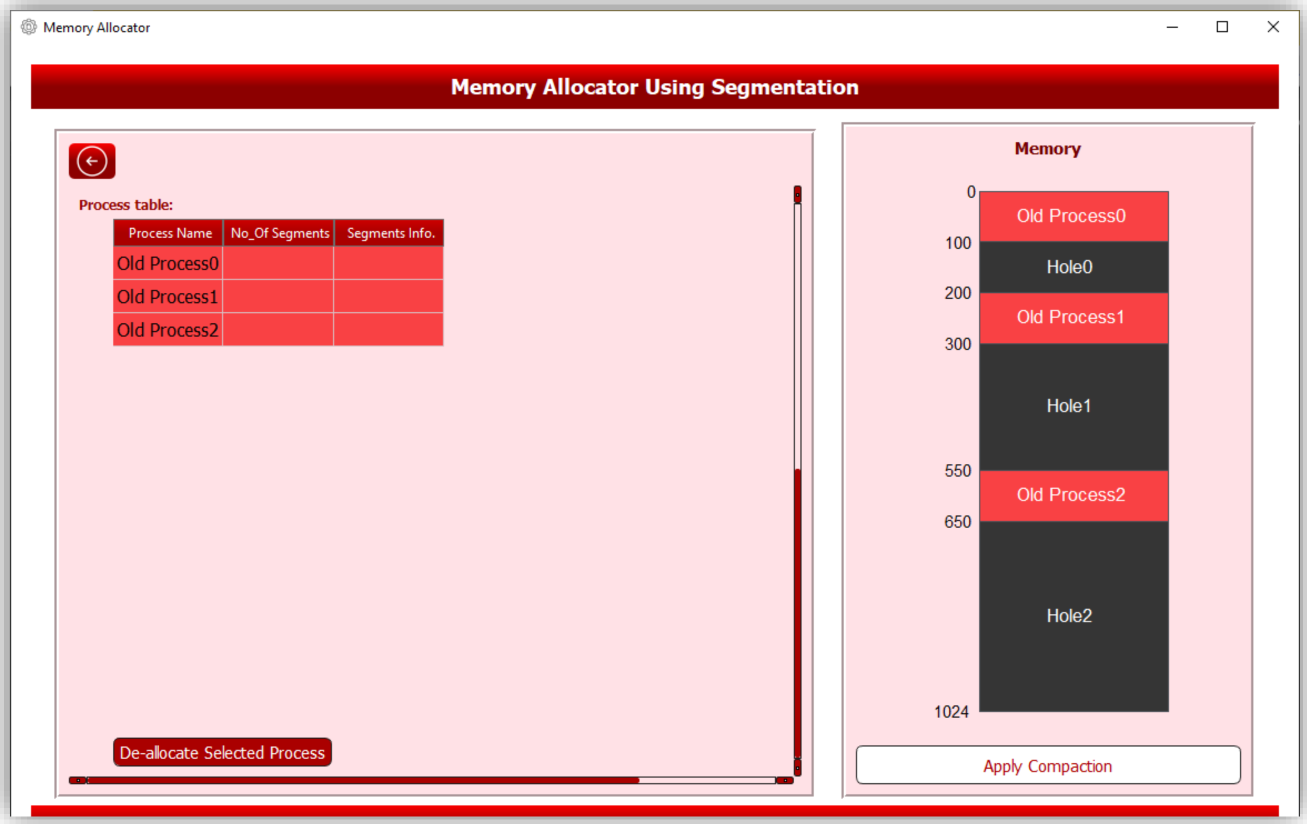
Figure 12, Example 1, old processes



Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
P0	1	Main: 100
P1	1	Main: 100
P2	2	Main: 100, Arrays: 100

Figure 13, Example 1, populating memory using worst fit

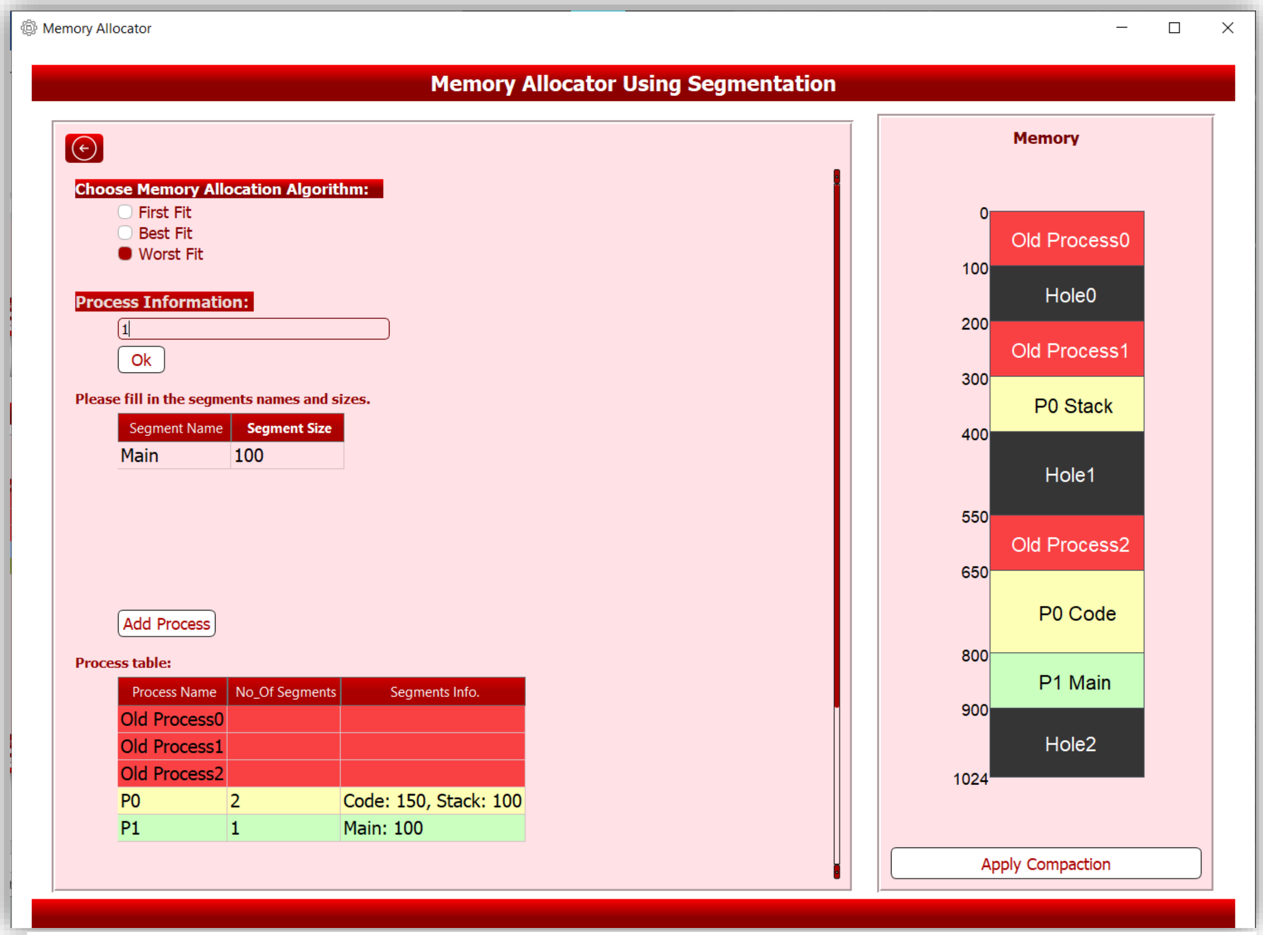
Example 2, before and after populating memory using worst fit:



Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		

Figure 14, Example 2, old processes



Choose Memory Allocation Algorithm:

First Fit Best Fit [✓ Worst Fit]

Process Information:

1

Please fill in the segments names and sizes.

Segment Name	Segment Size
Main	100

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
P0	2	Code: 150, Stack: 100
P1	1	Main: 100

Figure 15, Example 2, populating memory using worst fit

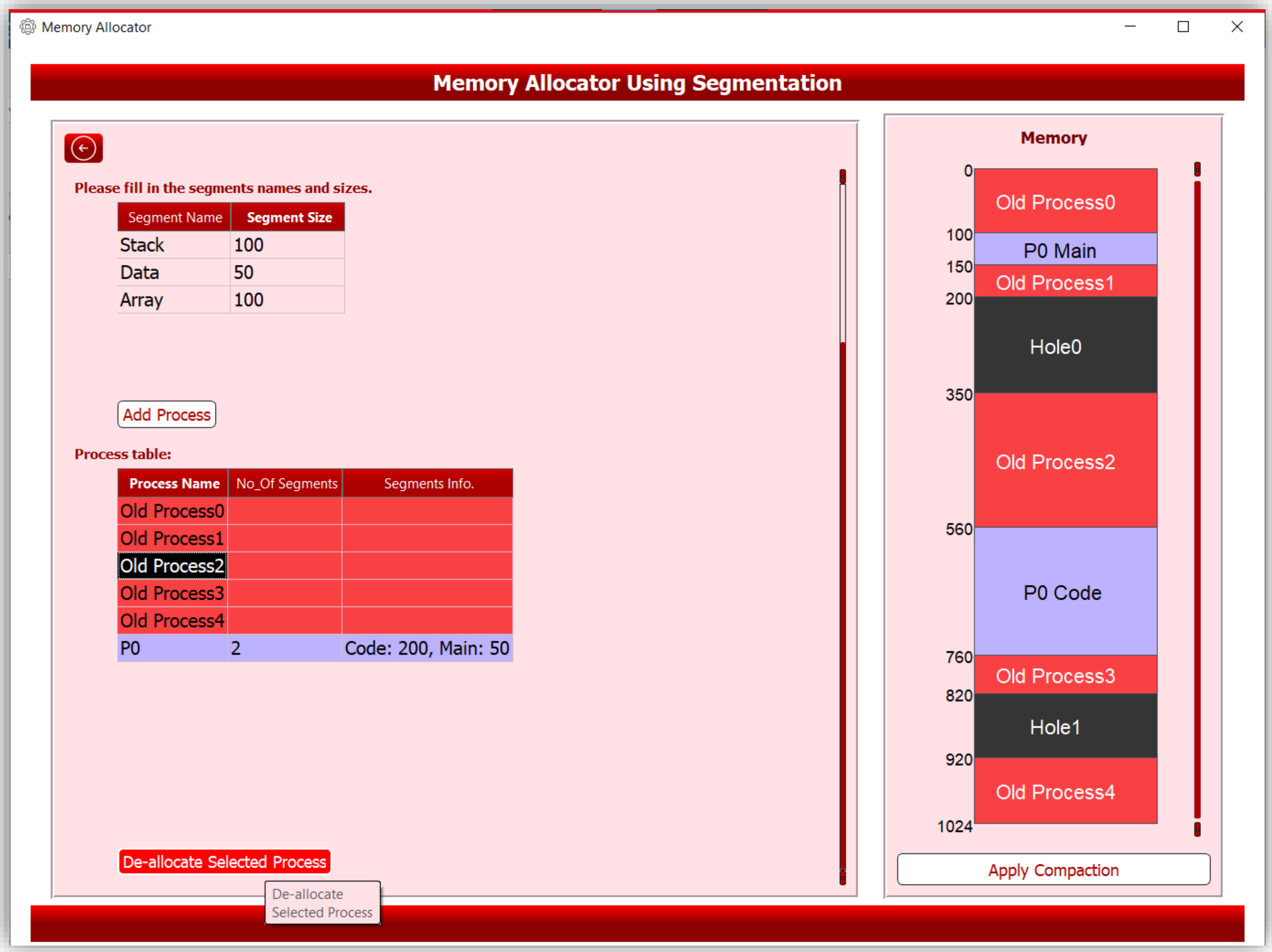
3. De-allocation of processes

The user has the option at this point to de-allocate selected process whether if it's an old or new process.

The user can refer to the process to be de-allocated by selecting any item in the same row of the process table, then clicking "De-allocate selected process" button.

3.1 De-Allocation of old process

Before de-allocation:


Memory Allocator
Memory Allocator Using Segmentation
Please fill in the segments names and sizes.
Add Process
Process table:
De-allocate Selected Process
De-allocate Selected Process
Memory
Apply Compaction

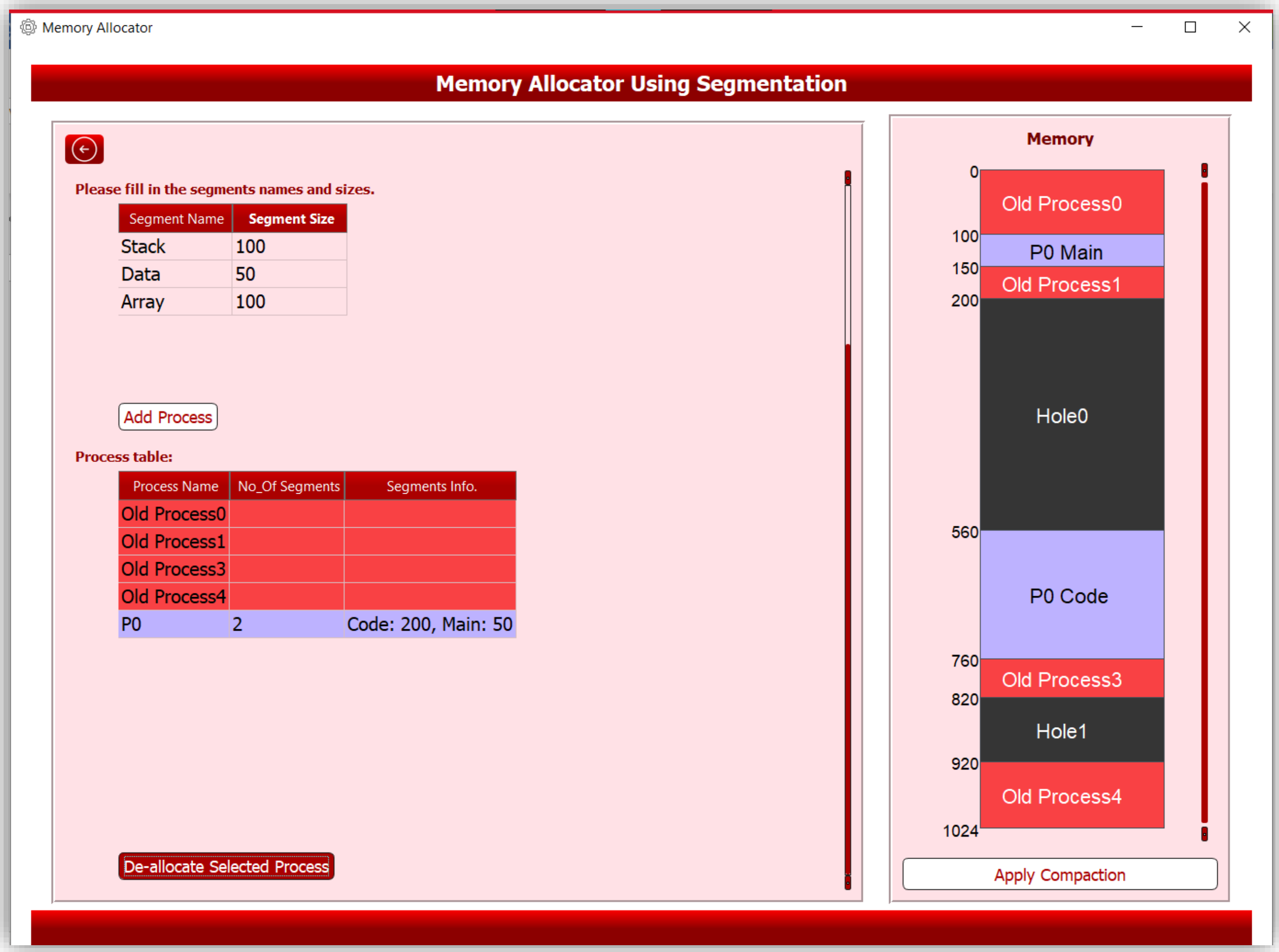
Segment Name	Segment Size
Stack	100
Data	50
Array	100

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
Old Process4		
P0	2	Code: 200, Main: 50

0	Old Process0
100	P0 Main
150	Old Process1
200	Hole0
350	Old Process2
560	P0 Code
760	Old Process3
820	Hole1
920	Old Process4
1024	

Figure 16, First, select the old processes

And, after de-allocation: (with the adjacent holes gets combined)



Memory Allocator

Memory Allocator Using Segmentation

Please fill in the segments names and sizes.

Segment Name	Segment Size
Stack	100
Data	50
Array	100

Add Process

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process3		
Old Process4		
P0	2	Code: 200, Main: 50

De-allocate Selected Process

Memory

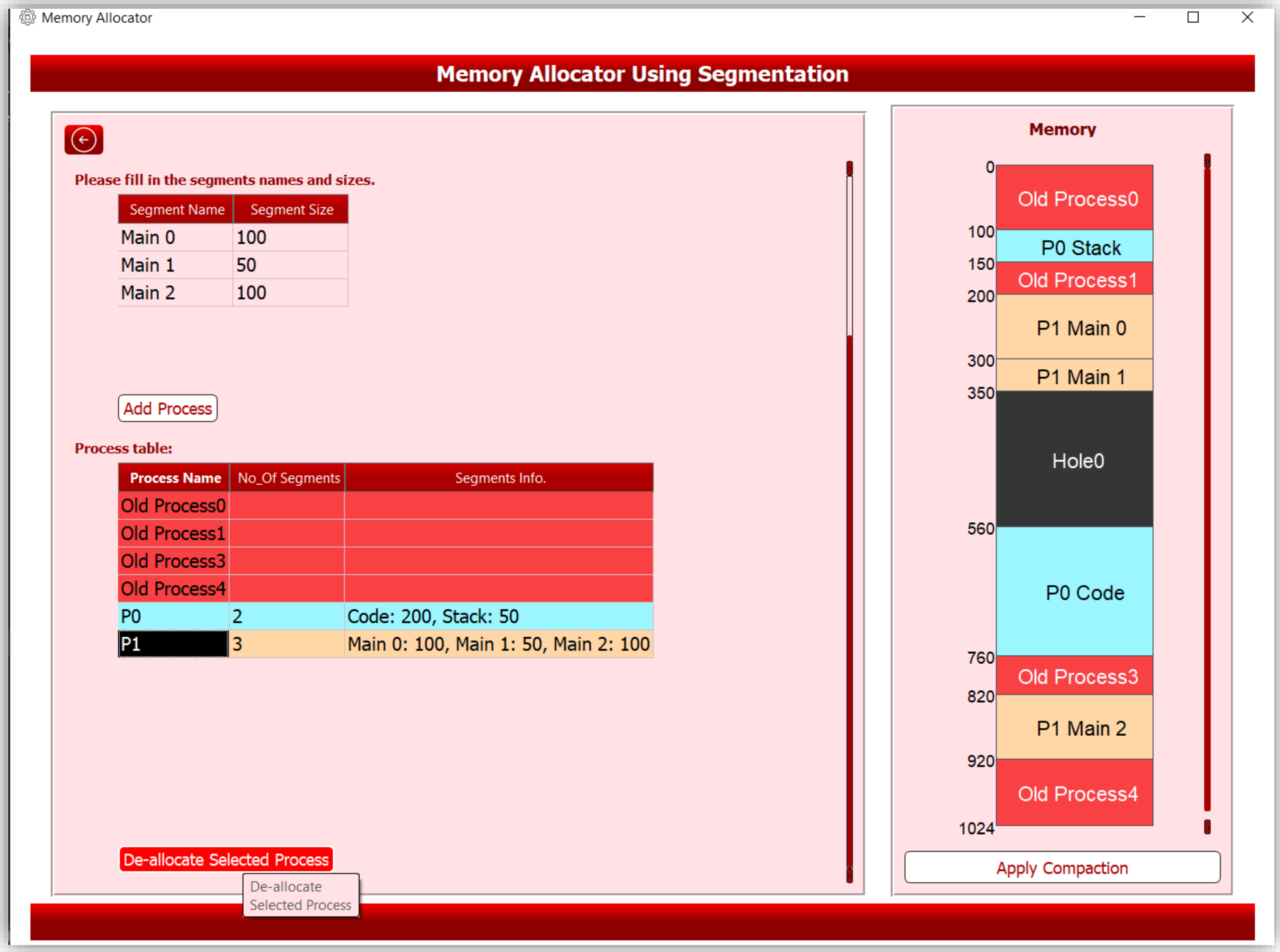
0	Old Process0
100	P0 Main
150	Old Process1
200	Hole0
560	P0 Code
760	Old Process3
820	Hole1
920	Old Process4
1024	

Apply Compaction

Figure 17, Then, click on "De-allocate Selected Process"

3.2 De-Allocation of new process

Before de-allocation:



Memory Allocator

Memory Allocator Using Segmentation

Please fill in the segments names and sizes.

Segment Name	Segment Size
Main 0	100
Main 1	50
Main 2	100

Add Process

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process3		
Old Process4		
P0	2	Code: 200, Stack: 50
P1	3	Main 0: 100, Main 1: 50, Main 2: 100

De-allocate Selected Process

De-allocate Selected Process

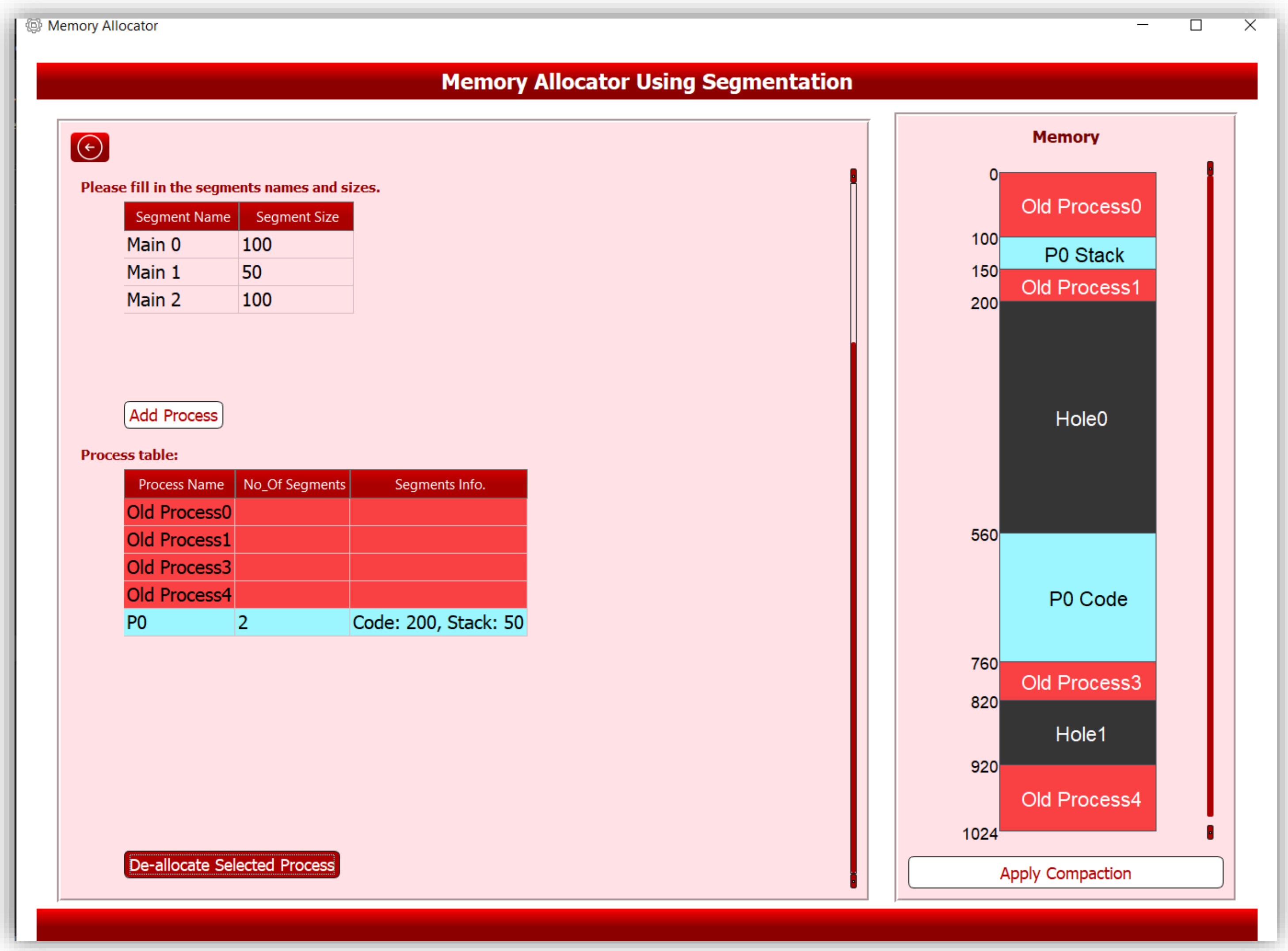
Memory

0	Old Process0
100	P0 Stack
150	Old Process1
200	P1 Main 0
300	P1 Main 1
350	Hole0
560	P0 Code
760	Old Process3
820	P1 Main 2
920	Old Process4
1024	

Apply Compaction

Figure 18, First, select the new process

And, after:



Please fill in the segments names and sizes.

Segment Name	Segment Size
Main 0	100
Main 1	50
Main 2	100

Process table:

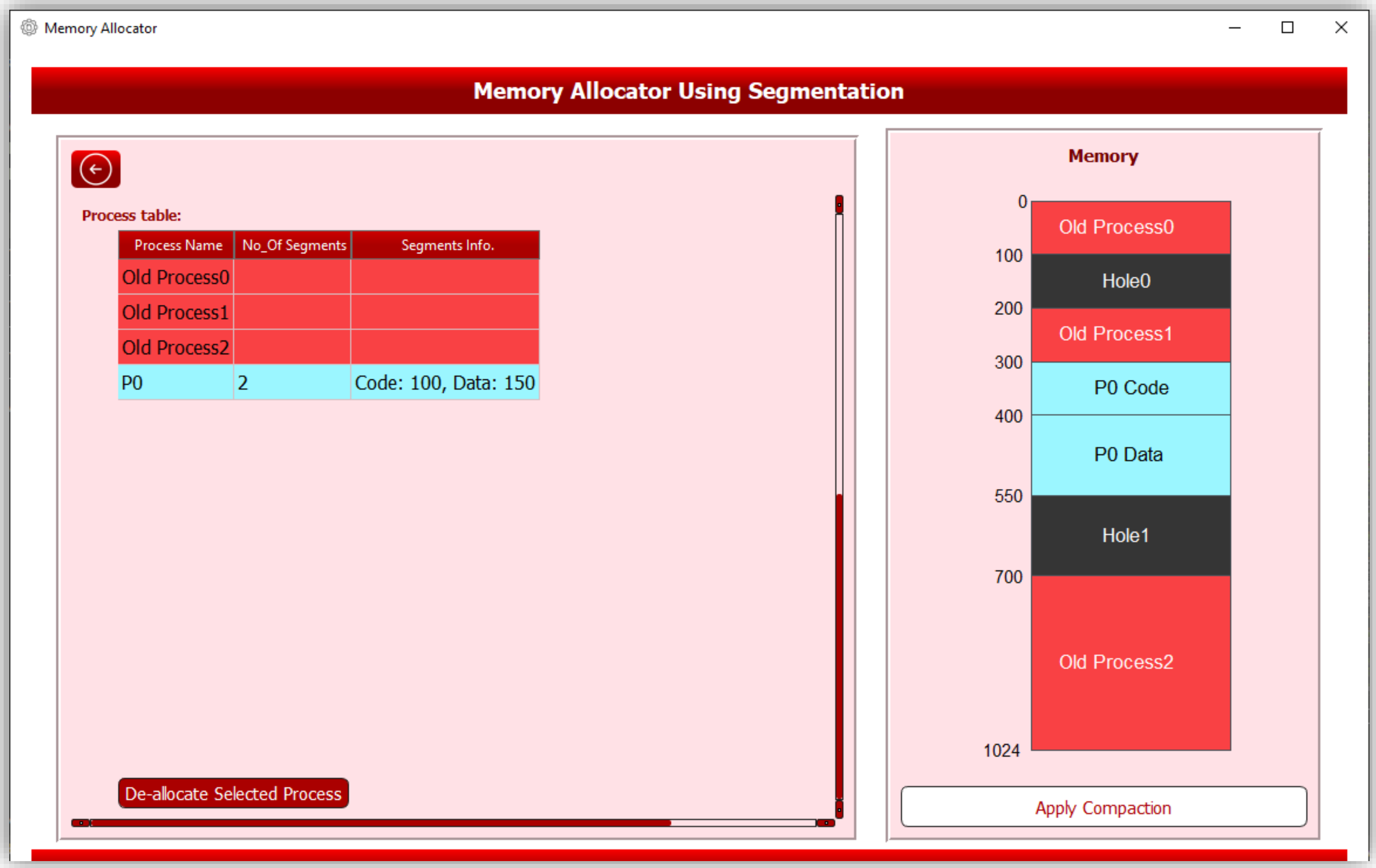
Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process3		
Old Process4		
P0	2	Code: 200, Stack: 50

Figure 19, Then, click on "De-allocate Selected Process"

4. Applying compaction

4.1 As an all-time option

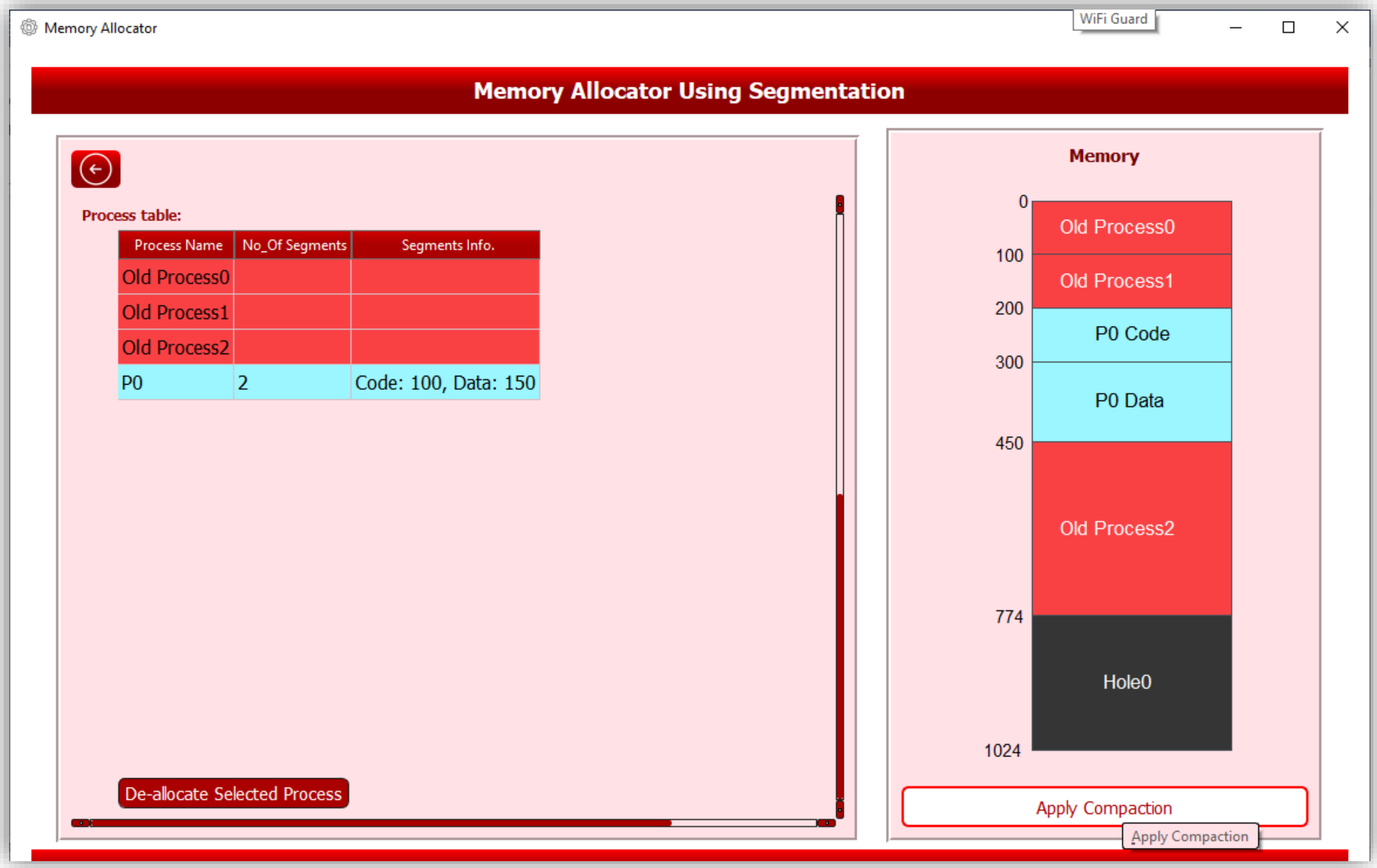
User can apply compaction on memory at any time during the insertion of processes or holes; to help allocate previously non-fit because of the external fragmentation. **Example**, before applying compaction:



Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
P0	2	Code: 100, Data: 150

Figure 20, A populated memory before applying compaction

After applying compaction:

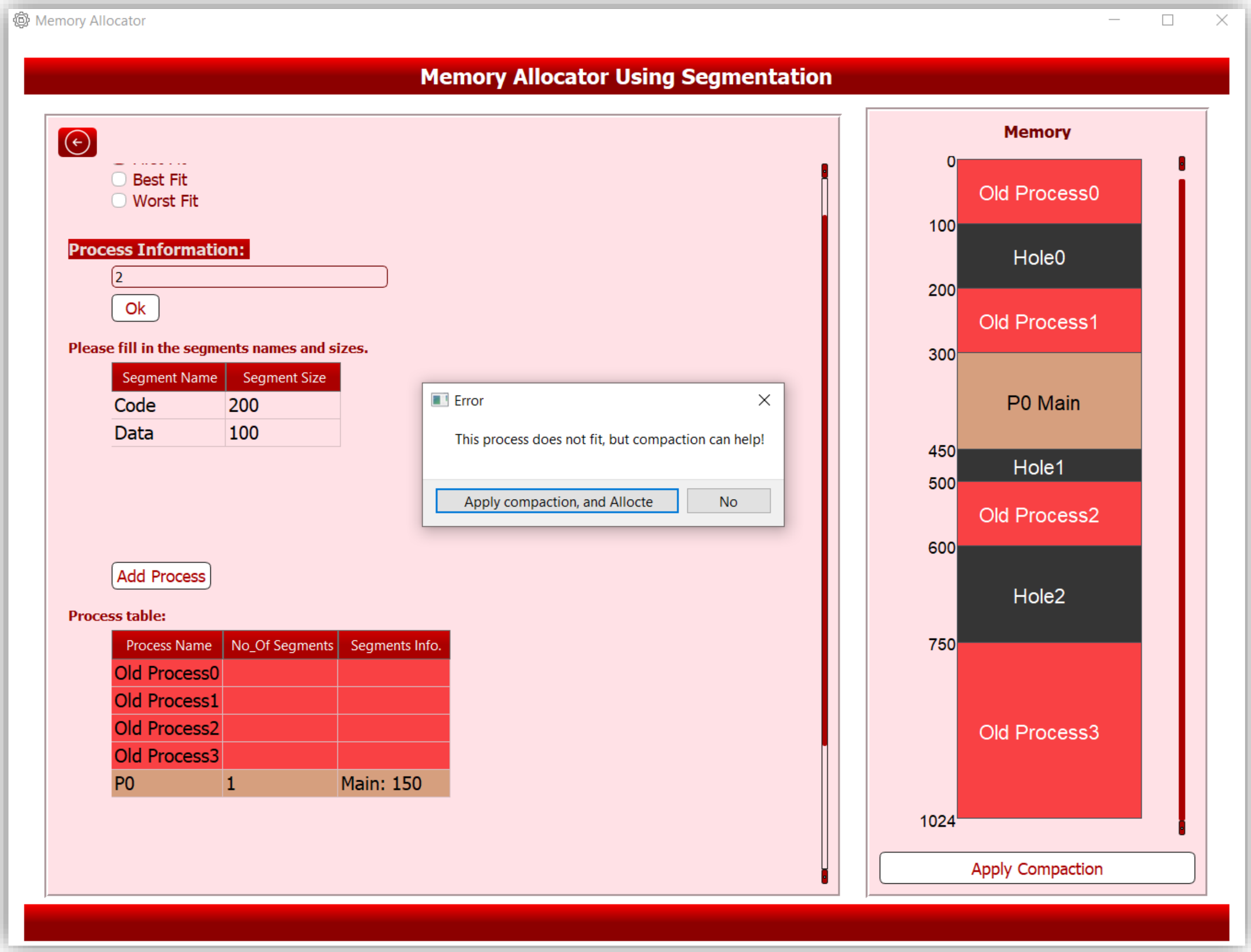


Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
P0	2	Code: 100, Data: 150

Figure 21, After applying compaction

4.2 As an offer when it can help

An error message normally emerges when the user tries to allocate a non-fit process, but if it can be allocated with the help of compaction, then the software asks him/her if he would in the error message.



Please fill in the segments names and sizes.

Segment Name	Segment Size
Code	200
Data	100

Error

This process does not fit, but compaction can help!

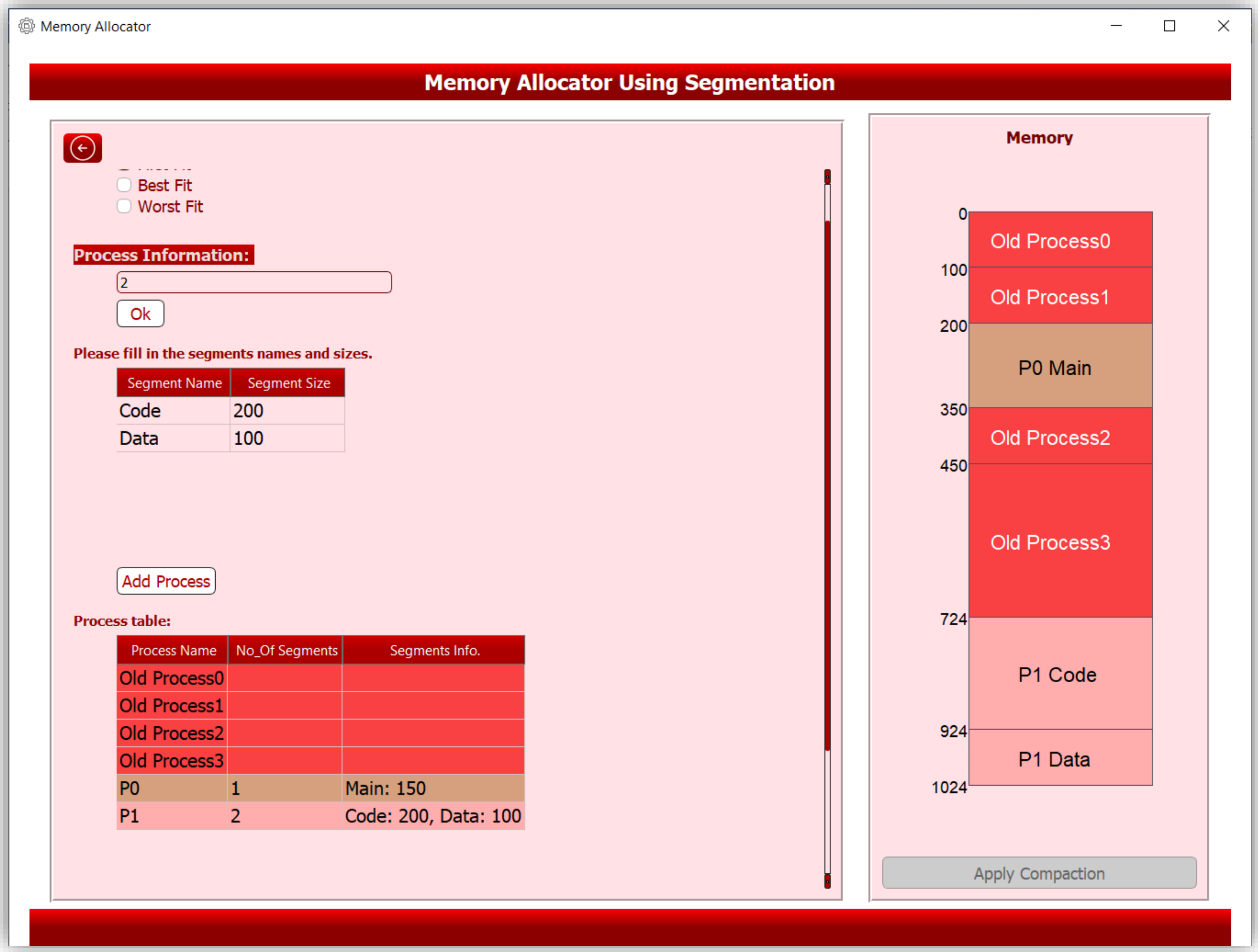
Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
P0	1	Main: 150

0	Old Process0
100	Hole0
200	Old Process1
300	P0 Main
450	Hole1
500	Old Process2
600	Hole2
750	Old Process3
1024	

Figure 22,The user tries to allocate a non-fit process, but the program tells that compaction can help

And, after agreeing to apply compaction, and allocate the process automatically:



Please fill in the segments names and sizes.

Segment Name	Segment Size
Code	200
Data	100

Process table:

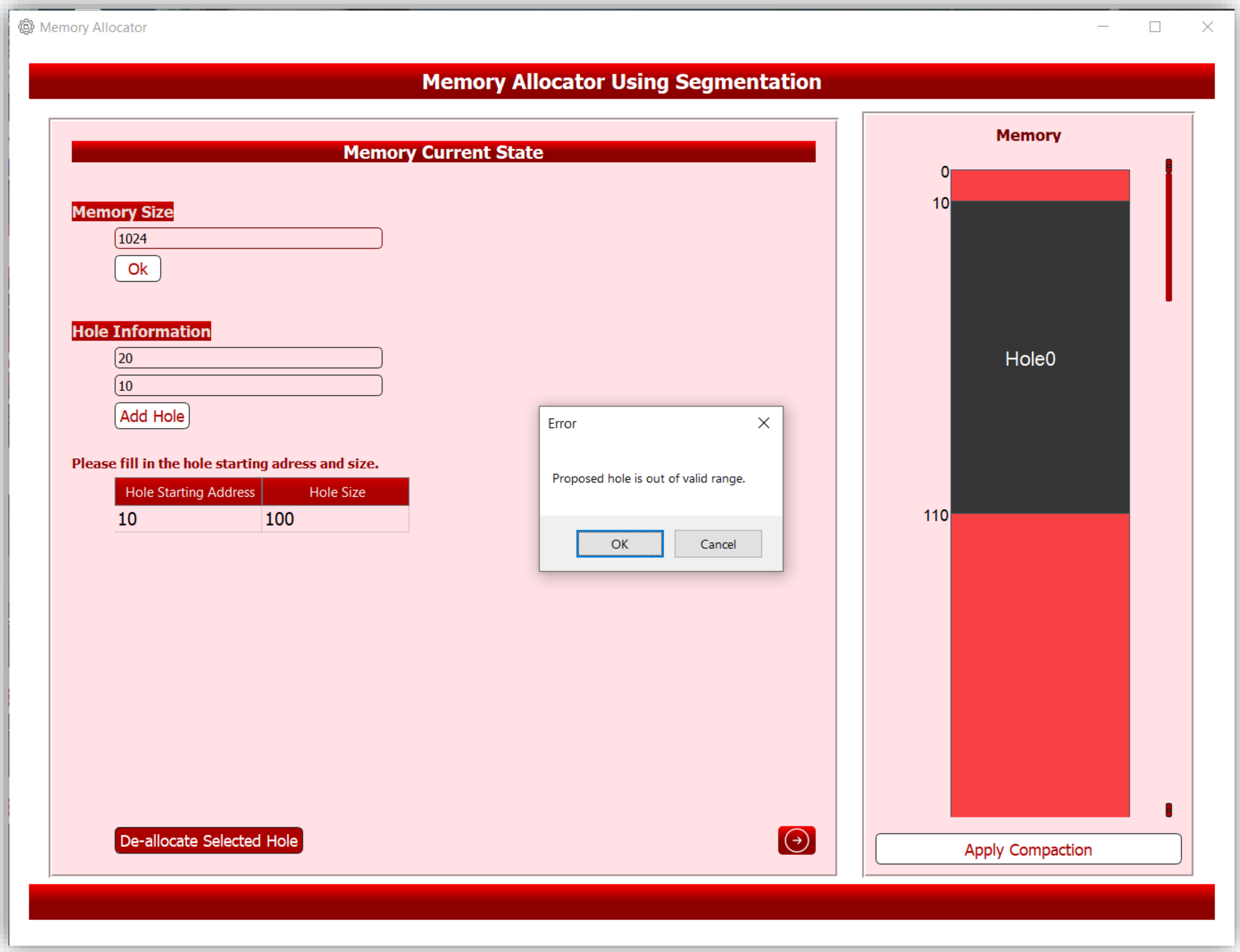
Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
P0	1	Main: 150
P1	2	Code: 200, Data: 100

Figure 23, The user allows the software to apply compaction, and allocate

5. Error Checking for more user-friendly GUI

5.1 Check if the entered hole is out of valid range

An error message emerges when the user tries to allocate a hole that over-laps with previous holes or its size is larger than available size.



Please fill in the hole starting adress and size.

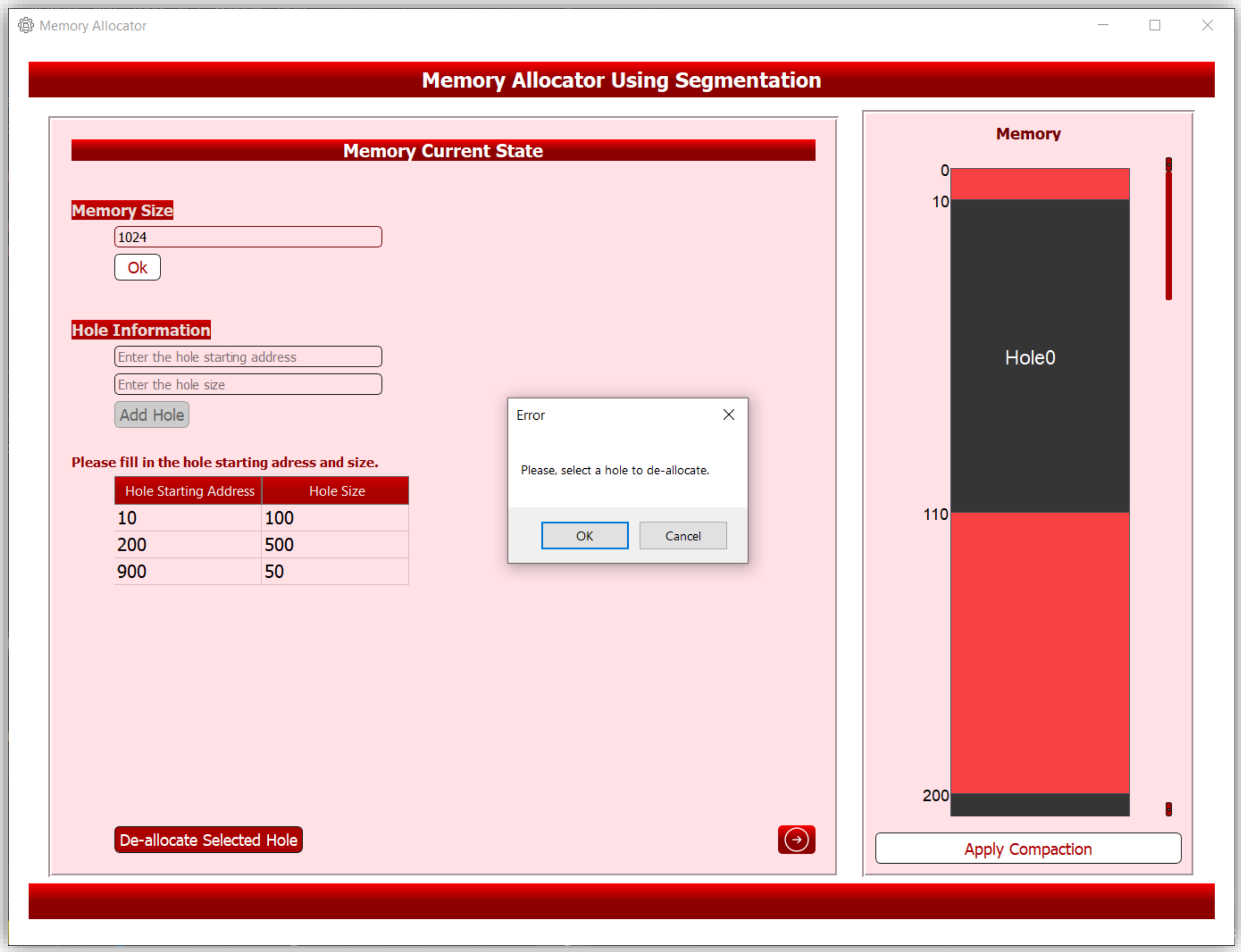
Hole Starting Address	Hole Size
10	100

Proposed hole is out of valid range.

Figure 24, The error message of adding a hole that is out of range

5.2 Trying to de-allocate (a hole or a process) with no selection

trying to de-allocate un-selected hole:



Memory Size: 1024

Please fill in the hole starting adress and size.

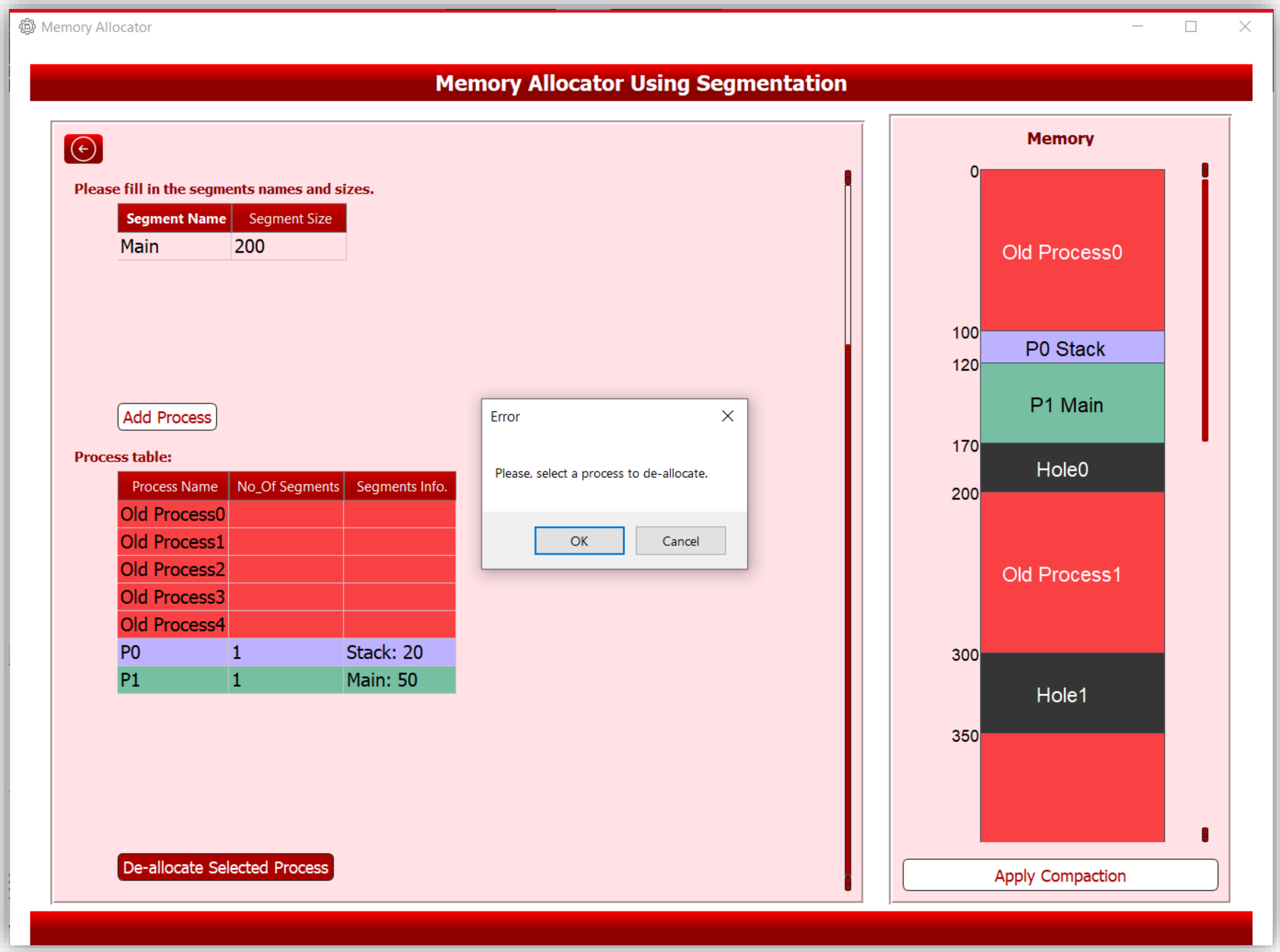
Hole Starting Address	Hole Size
10	100
200	500
900	50

Error

Please, select a hole to de-allocate.

Figure 25, The error message of de-allocating a hole without selecting any one

trying to de-allocate un-selected process:


Memory Allocator

Memory Allocator Using Segmentation

Please fill in the segments names and sizes.

Segment Name	Segment Size
Main	200

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
Old Process4		
P0	1	Stack: 20
P1	1	Main: 50

Error

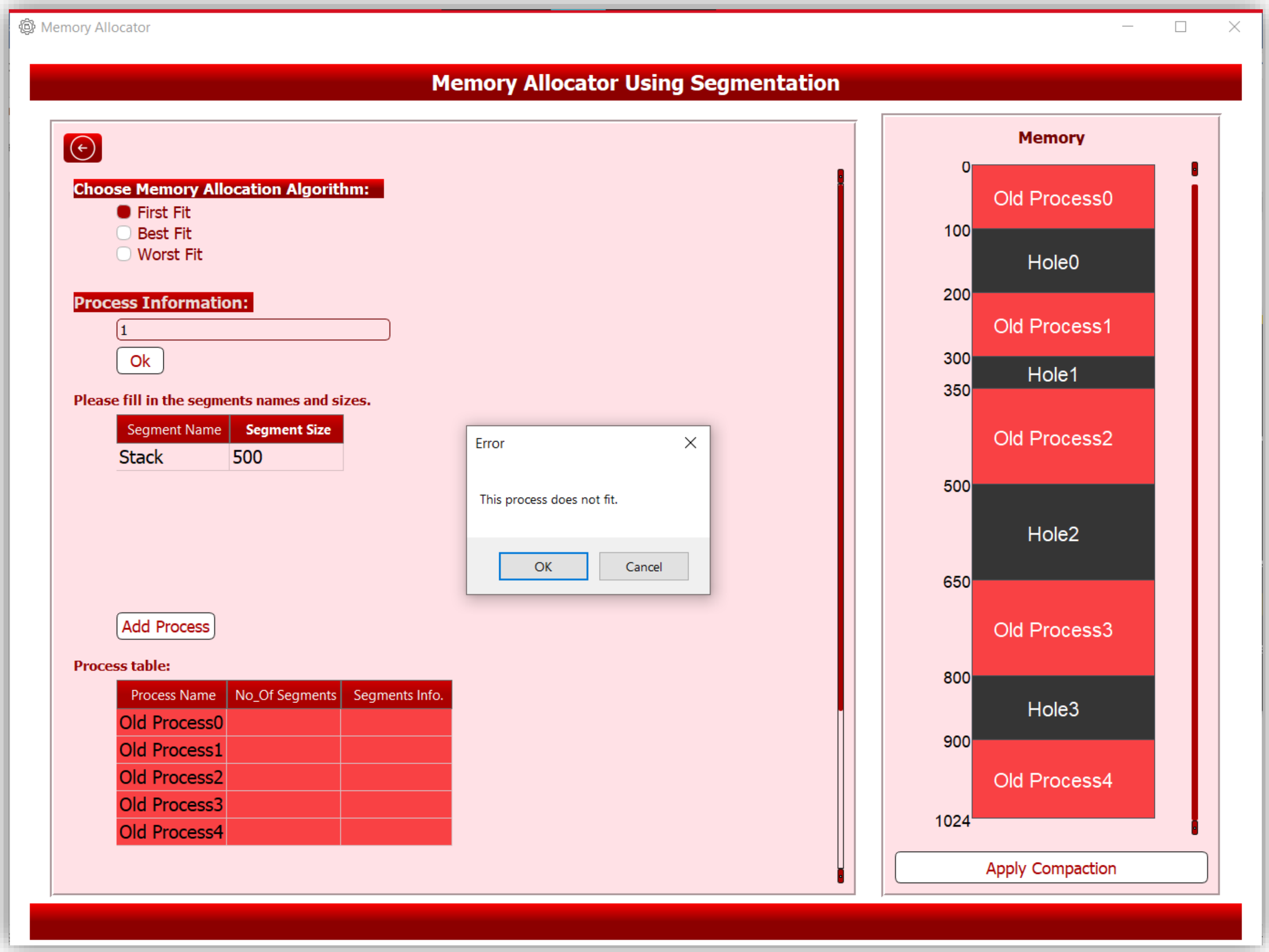
Please, select a process to de-allocate.

Figure 26, The error message of de-allocating a process without selecting any one

5.3 Entering a process that does not fit

An error message emerges when the user tries to enter a process that does not fit within any hole, or its size is bigger than the memory size.

But if this process can be allocated after applying compaction, then the error message will provide compaction and allocation as mentioned before.



Choose Memory Allocation Algorithm:

● First Fit
○ Best Fit
○ Worst Fit

Process Information: 1

Please fill in the segments names and sizes.

Segment Name	Segment Size
Stack	500

Error

This process does not fit.

OK Cancel

Process table:

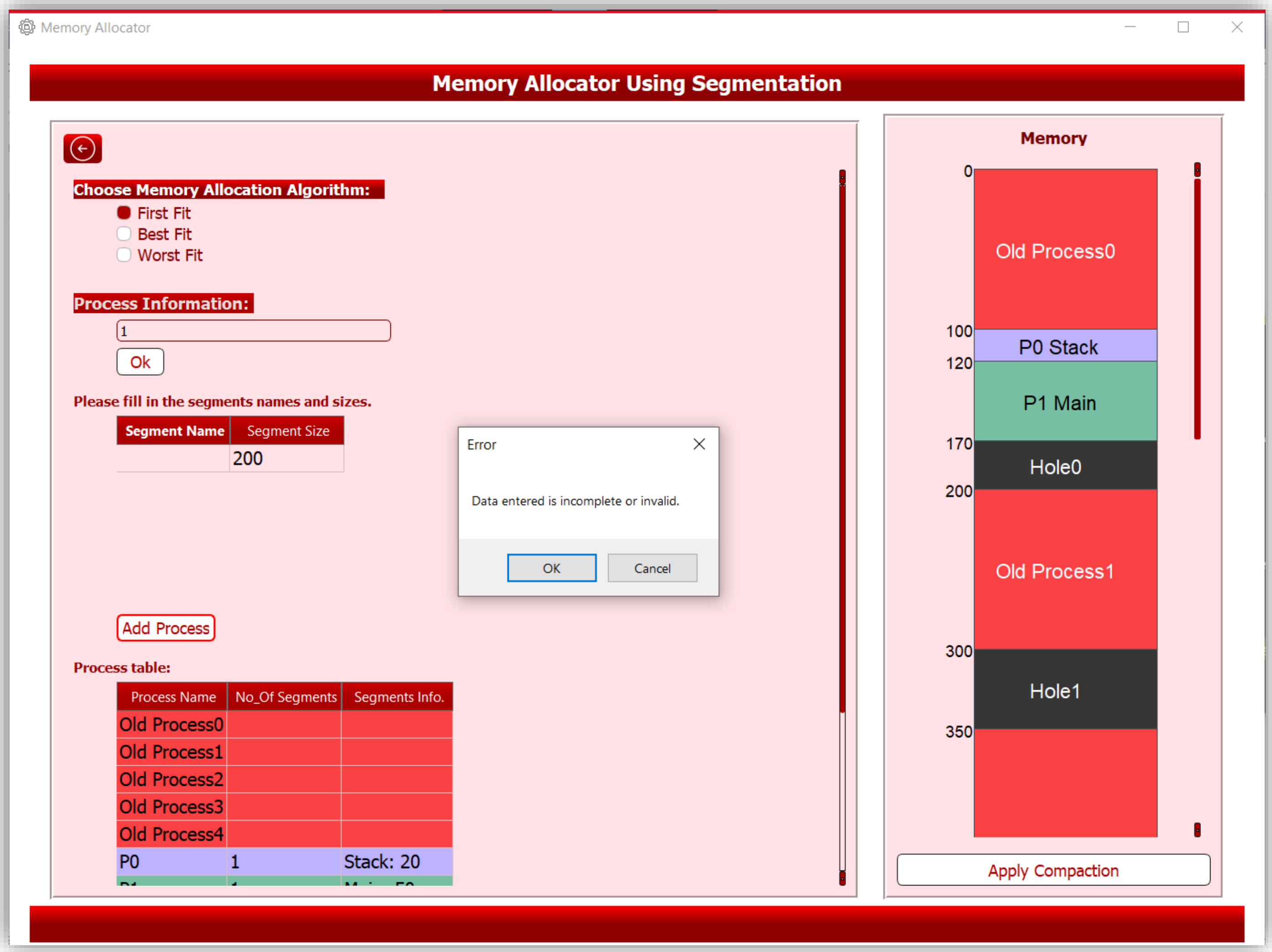
Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
Old Process4		

0	Old Process0
100	Hole0
200	Old Process1
300	Hole1
350	Old Process2
500	Hole2
650	Old Process3
800	Hole3
900	Old Process4
1024	

Figure 27, The error message appears when trying to allocate a process that does not fit

5.4 Trying to enter invalid or incomplete data

An error message appears when the user tries to add a process without filling all the data required.


Memory Allocator

Memory Allocator Using Segmentation

Choose Memory Allocation Algorithm:

[✓ First Fit] Best Fit Worst Fit

Process Information:

1

Please fill in the segments names and sizes.

Segment Name	Segment Size
	200

Error

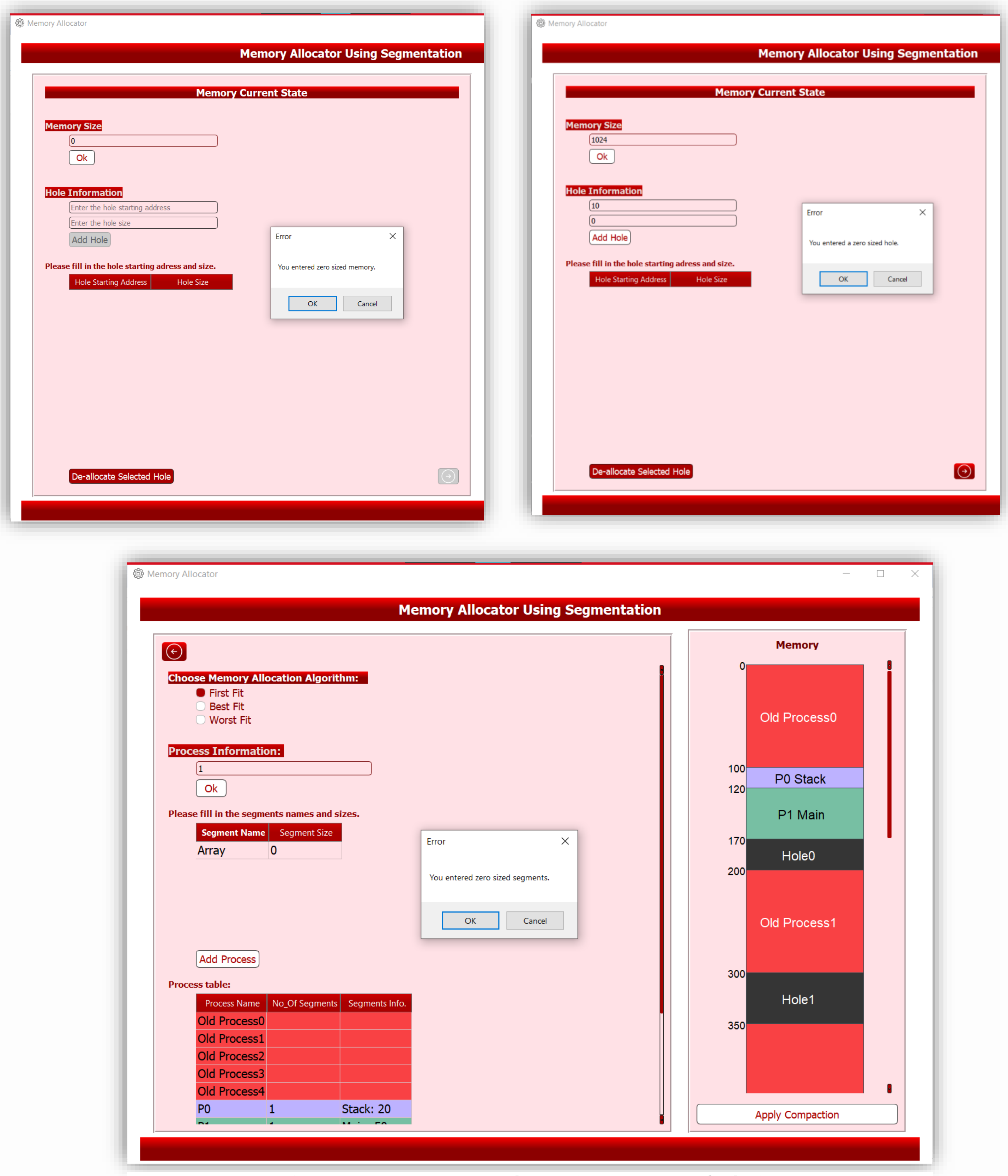
Data entered is incomplete or invalid.

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
Old Process4		
P0	1	Stack: 20

Figure 28, The error message that appears when the data entered is not complete or invalid

As well as an error message appears if the user enters the memory size, the hole size or the segment size equal zero.



Memory Allocator

Memory Allocator Using Segmentation

Memory Current State

Memory Size

0

Hole Information

Enter the hole starting address

Enter the hole size

Please fill in the hole starting adress and size.

Hole Starting Address	Hole Size

Error

You entered zero sized memory.

Memory Allocator

Memory Allocator Using Segmentation

Memory Current State

Memory Size

1024

Hole Information

10

0

Please fill in the hole starting adress and size.

Hole Starting Address	Hole Size

Error

You entered a zero sized hole.

Memory Allocator

Memory Allocator Using Segmentation

Choose Memory Allocation Algorithm:

[✓ First Fit] Best Fit Worst Fit

Process Information:

1

Please fill in the segments names and sizes.

Segment Name	Segment Size
Array	0

Error

You entered zero sized segments.

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		
Old Process2		
Old Process3		
Old Process4		
P0	1	Stack: 20

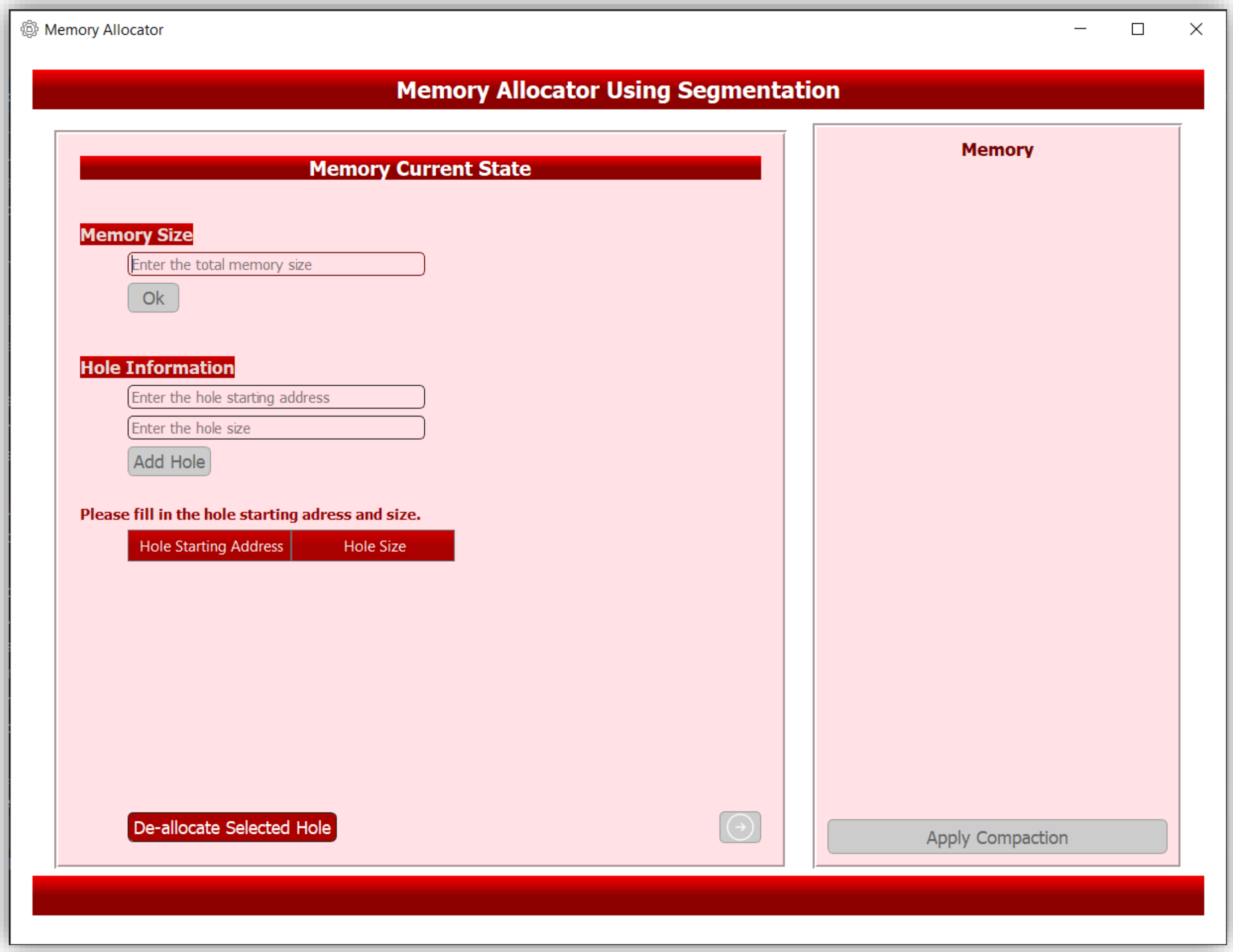
Figure 29, Error message appears when entering zero as an input for the size

5.5 Disabling buttons at certain conditions

The software does not allow the user to proceed at any point without filling all the needed data in integer type; either by disabling the buttons such as (the memory size insertion button “Ok”, the hole information insertion button “ Add Hole” and the number of segments insertion button “Ok”).

Or by displaying error messages that indicates if the data is incomplete in the case of the segments table as mentioned before.

Other buttons are also disabled at certain conditions such as (the “Next” button is disabled until entering the memory size, the “Apply Compaction” button is disabled until the memory size is defined and there is at least one hole.



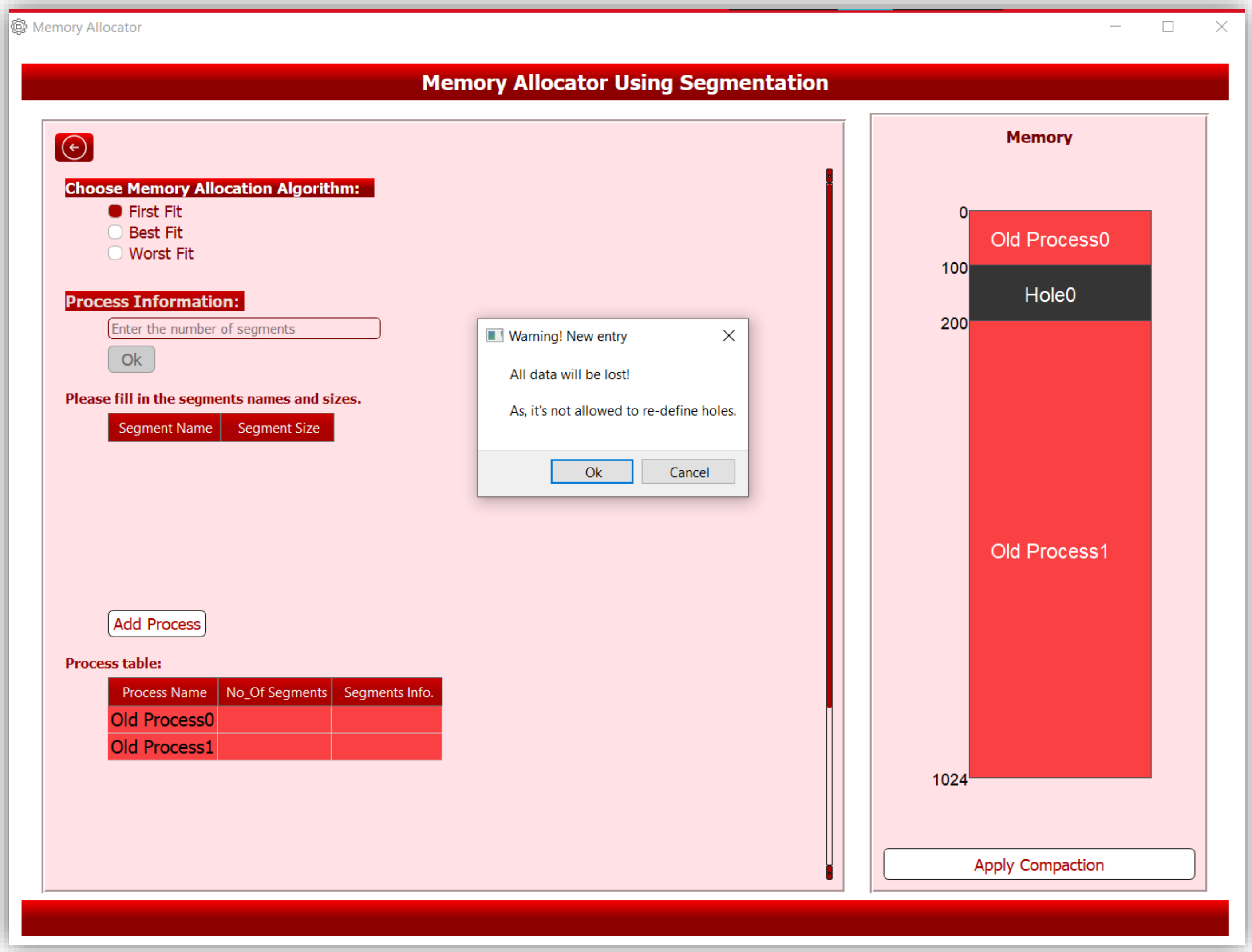
Please fill in the hole starting adress and size.

Figure 30, Disabling buttons at certain conditions

5.6 Warning when trying to redefine memory state

The software disallows the user to redefine the holes' locations after navigating to the next page, as it's not possible to reshape the memory after recognizing old processes and entering new ones.

So, a warning message appears to inform the user that "all the data will be lost" if such an action to be taken.


Memory Allocator

Memory Allocator Using Segmentation

Choose Memory Allocation Algorithm:
[✓ First Fit] Best Fit Worst Fit

Process Information:

Enter the number of segments

Ok

Please fill in the segments names and sizes.

Segment Name	Segment Size

Add Process

Process table:

Process Name	No_Of Segments	Segments Info.
Old Process0		
Old Process1		

Warning! New entry

All data will be lost!

As, it's not allowed to re-define holes.

Ok Cancel

Figure 31, Error message when hitting back

6. Acknowledgement

We would like to state that we have learned a lot doing this project and the previous one, it was just fun 😃.

And, we would like to thank all the teaching staff of the O.S. course for their help throughout the semester and their immediate responses for our questions.

Thank you.