Amal Khaled Mohamed

2017-12569

GitHub Link: https://github.com/Amal-Safwat/Memory-Allocation-Project

Memory Allocation Simulator

The access nature of main memory allows us to be more flexible when allocating processes. Many processes are usually always stored on the same main memory. The key issue is determining how much memory space should be allocated to these processes. Processes may be accessed quickly and effectively. There are three major memory techniques. The methods "First Fit," "Best Fit," and "Worst Fit" are commonly used. Each approach has its own set of benefits as well as drawbacks. Although some systems support all three methods, but it is more common for a system to support only one.

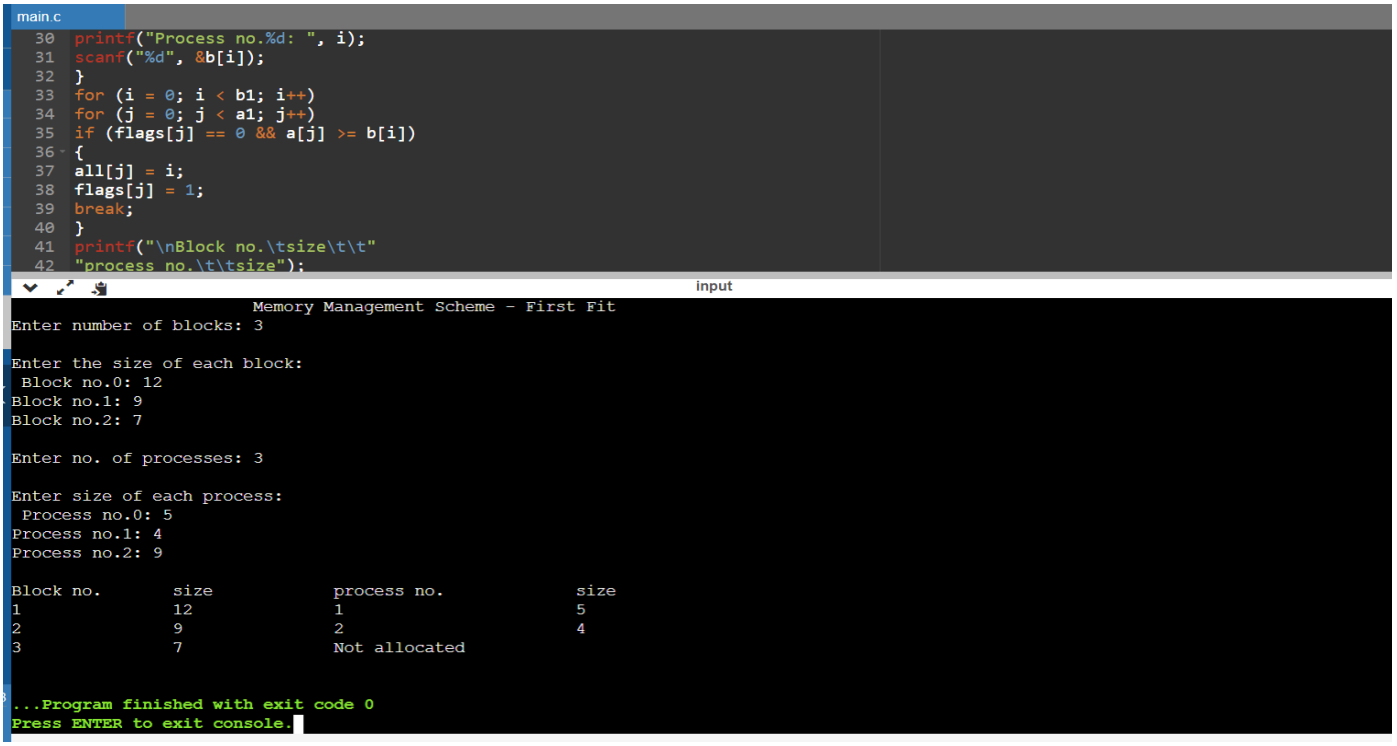
These are the memory allocation techniques which are as follows:

- **First Fit:**

  In this method, all of the blocks are checked in order. So, we'll take the first process and compare its size to the block's first size. If it's less, we'll go to the next block, and so on, until all of the processes have been assigned.

First Fit Algorithm:

1. Take the number of blocks and number of the processes from the user.
2. Read the size of each block and the size of all the process requests.
3. Start allocating the processes.
4. Display the results.

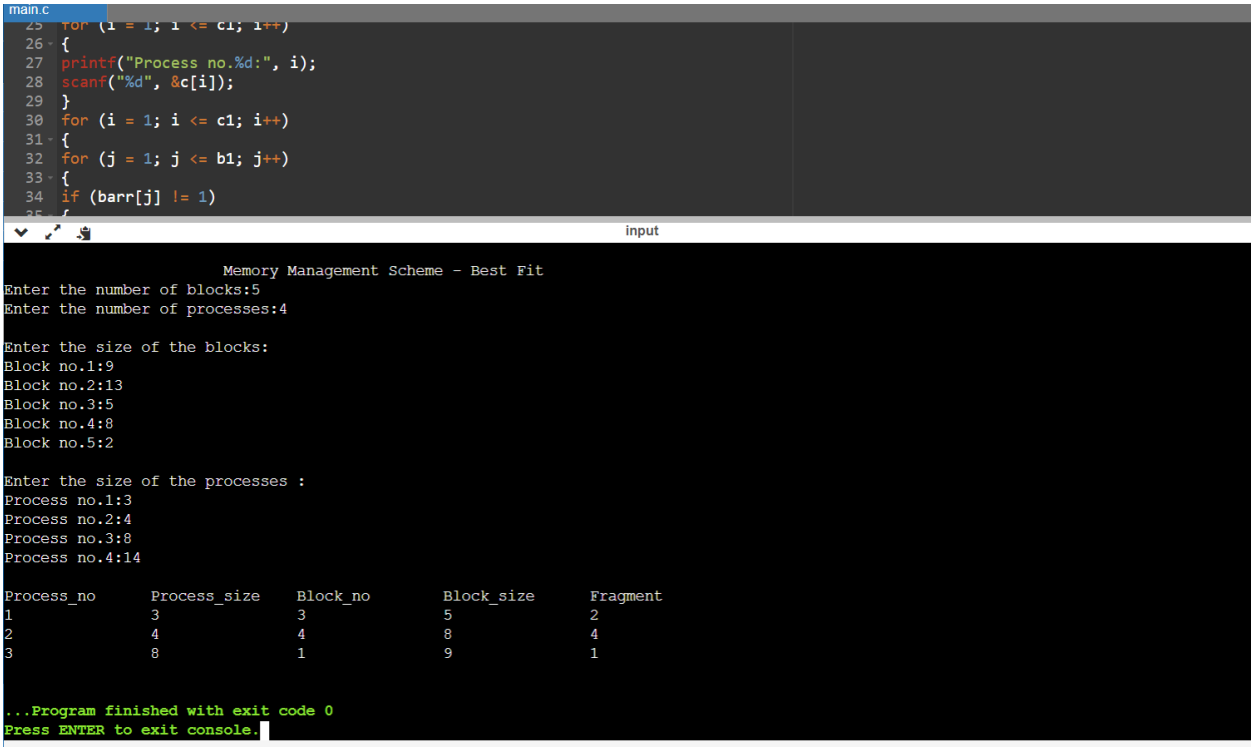


- **Best Fit:**

  In this method, the memory management algorithm selects the best memory block. As a result, we select the smallest block and allow the incoming process to fulfil the memory request. As a result, the memory consumed in this technique is put to the best possible use. The disadvantage is that it increases the time required because it compares the block with all memory sizes. As a result, this procedure is slower than the others.

Best Fit Algorithm:

1. Take the number of blocks and number of processes from the user.
2. Get the size of each block and process.
3. Then select the best memory block.
4. Display the result.
5. The fragmentation column will keep track of wasted memory.



- **Worst Fit:**

  This memory management technique allocates the largest free partition that will be large enough to hold the processes. It searches for all available memory for free partitions and assigns the largest of them to the partition. Because of the many drawbacks of this approach, it is not recommended for use in the real world.

Worst Fit Algorithm:

1. Read total number of block and files.
2. Get the size of each block and the files from the user.
3. Start from the first process and find the maximum block size that can be assigned to the current process, if found then assign it to the current process.
4. If not found then leave that process and check the rest of the processes.
5. Display the result.
6. The fragmentation column keeps the track of wasted memory.

main.c

```
38  if (top < temp)
39  {
40  file_arr[i] = j;
41  top = temp;
42  }
43  }
44  }
45  frag[i] = top;
```

input

```
                        Memory Management Scheme - Worst Fit
Enter the Total Number of Blocks: 5
Enter the Total Number of Files: 4

Enter the Size of the Blocks:
Block No.1:     5
Block No.2:     7
Block No.3:     4
Block No.4:     6
Block No.5:     3
Enter the Size of the Files:
File No.1:      1
File No.2:      3
File No.3:      4
File No.4:      2

File Number     File Size       Block Number    Block Size      Fragment
0               1               4               3               2
1               3               0               5               0
2               4               0               5               0
3               2               0               5               0


...Program finished with exit code 0
Press ENTER to exit console.
```