Assignment # 4

**Memory Management**

**Deadline is next week from lab week.**

**Maximum three students and in case of four, then bonus part is mandatory and considered a part of assignment and not bonus**

**Students should be from same lab**

**Objective**

Design and implement a memory management simulator to simulate memory allocation and de-allocation mechanisms.

**Description**

The memory management simulator will maintain a list of partitions available in its memory. Each partition node maintains the following information:

**-** Starting Memory address

- Size of whole partition

- Status of partition (either occupied or free)

There are two types of memory requests in the simulator:

1. Memory allocate
2. Memory de-allocate
3. Defragmentaion (Bonus Part)

Consider this is how memory looks at the program start:

|  |  |
| --- | --- |
| Partition no. | partition |
| 1 | Size = 50 bytes free |
| 2 | Size = 70 bytes occupied |
| 3 | Size = 20 bytes free |
| 4 | Size = 50 bytes occupied |
| 5 | Size = 50 bytes occupied |
| 6 | Size = 60 bytes free |
| 7 | Size = 30 bytes free |
| 8 | Size = 140 bytes occupied |
| 9 | Size = 42 bytes free |

**1- Memory Allocate:**

The memory allocate request accepts the size of memory to be allocated (M) as an input. The simulator seeks a suitable partition to allocate memory of size M. The partition list is searched for a free partition that fits M according to one of the following policies:

1. **First Fit**- The list is searched from the beginning, the first partition that is large enough to fit M is chosen
2. **Best Fit** – The list is searched from the beginning to end, the smallest partition that is large enough to fit M is chosen.
3. **Worst Fit** – The whole list is searched; the largest partition that can fit M is chosen.

When a partition is found its remaining size (its size – needed size) will be considered a free partition.

The memory allocate procedure returns the address of the allocated partition or negative value if it fails to allocate a memory partition.

Note: addresses are sequential i.e. first partition in the above memory has address = 1 and second partition has address = 51 and so on.

**2- Memory Deallocate:**

The memory de-allocate frees and marks a specific partition as available for allocation. It accepts the starting address S of the partition to be freed. The partition list is searched to find the occupied partition with starting address S. If found the partition is indicated as free.

Note: it is possible to fail to free a partition when the specified address S is not a valid starting address or the partition starting at S is already marked as free.

The memory deallocate procedure returns a Boolean value that is true if de-allocation was successfully performed and false otherwise.

**3- Defragmentation (BONUS MARK)**

External fragmentation occurs when small sized free partitions are scattered across the memory in a way that fails to satisfy memory allocation requests. While internal fragmentation occurs when memory is allocated into partitions of much larger size than the actual requested size and internal memory in a partition is wasted.

Defragmentation is a method used to group small partitions into larger ones to satisfy memory requests. Defragmentation includes traversing the whole partition list to do the following:

Eliminating external fragmentation for all memory (Grouping free blocks)

To eliminate external fragmentation, we should group all occupied partitions to make one big contiguous occupied block and thus grouping all free partitions together to make ONE big free partition that can be used later.

**Simulator Input:**

Your Simulator should do the following

* Ask user about operation to be performed
  1. Allocation
     + First-Fit
     + Best-Fit
     + Worst-Fit
  2. De-allocation
  3. Defragmentation

In case of allocation, the allocation policy should be determined

**Assignment #4**

**Memory Management Simulator**

**Students IDs and Names:**

|  |  |
| --- | --- |
| **ID** | **Name** |
|  |  |
|  |  |
|  |  |
|  |  |

**Features:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Grade** | **Team Grade** |
| Simulator definition and Initialization | 1 |  |
| First Fit Allocation | 2 |  |
| Best Fit Allocation | 2 |  |
| Worst Fit Allocation | 2 |  |
| Deallocation | 1 |  |
| Individual Student Evaluation (based on each student's work) | 2 |  |
| **Total** | **10** |  |

**Bonus:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Grade** | **Team Grade** |
| Defragmentation (Reallocation) | 2 |  |