Operating system I
Assignment # 4

Memory Management

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

- Size of allocated memory within the partition

- Status of partition (either occupied or free)

Initially, the whole memory range is a single free partition. There are two types of memory requests in the simulator:

1- Memory allocate
2- Memory de-allocate

In addition to defragmentation option. Follows a detailed description for each operation:

1- Memory Allocate:

The memory allocate request accepts the size of memory to be allocated (M) as an input and allocation policy for that request. 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:

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

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

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 that 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

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 for the following cases:

Case 1:Suppose a partition with size N has allocated memory of size M, it is then split into two partitions (eliminating internal fragmentation)

a. Partition P with the first M bytes is considered a separate partition
b. Partition Q with the remaining N-M bytes as a new free partition.

Case 2: Grouping *contiguous* free blocks (eliminating external fragmentation)

a. Check the preceding block of each partition P in the list, if the block is free, then the two free blocks are merged as a new free block.
b. Check the succeeding block of P, if the block is free, then the two free blocks are merged as a new free block.

In the best case, both the preceding and succeeding blocks of P are free andthe three free blocks are merged as a new free block.

Case 3:**(Bonus Marks)**. Grouping *non-contiguous* free blocks (eliminating external fragmentation)

Maintain a list of all free partitions through the whole partition list. All partitions are reallocated to group all free partitions as a one new partition at the end of the list as shown below.

Assume the following memory allocation

After defragmentation we get the following partition allocation:

*Note: Defragmentation request takes case number as parameter to determine which case it's going to execute on the current memory. it will be called explicitly by simulator users.

Simulator Input:

Your Simulator should accept the following arguments as an input

- Initial Free Memory Size
- Operation to be performed
    1. Allocation(policy type)
    2. De-allocation()
    3. Defragmentation(case type)
    4. Print Memory status

Submission instructions:

1. **Submission deadline date is Saturday 9th December**
2. **The assignment is submitted in group of maximum 3 persons**.

Criteria	Grade
Memory Allocation: Best Fit	**5**
Memory Allocation: First Fit	**5**
Memory Allocation: Worst Fit	**5**
De-allocation	**5**
Defragmentation: Case 1	**10**
Defragmentation: Case 2	**10**
Defragmentation: Case 3	**5**
Print Memory status in a good way	**5**
Correct output	**5**