



Figure 7.5 Example Memory Configuration before and after Allocation of 16-Mbyte Block

Source of the picture: Stallings 9 edition

2nd Mandatory Assignment (to be done in a group)

Operating Systems(62588) Lecture 6

Teacher: Bhupjit Singh

Updated on Date: 05/10/2022

To be handed in DTU Learn

Setup

You applied for a job in Apple for a position of OS maintainer for MACOS. You aced the first round of interview and you have been invited to the second round of interview.

The Second round comprises of a task that you have to implement. There is a partial code of a previous team where they made a skeleton for memory allocation scheme.

You have to complete this code. They have given to access to the code one week ahead for their presentation of the possible solution.

The reason for this is – wrap your head around the **code structure** and refresh your concepts of **DOOUBLY LINK LIST** (very important!!!!).

You will be given the opportunity to ask questions after they have presented the possible solution. So, it is imperative that you know the code well and doubly linked list, so that you can ask the right questions!!! You will be provided with a **guideline** to complete the task next week but to understand the guideline **you must have understood the code structure and implementation of doubly link list.**

Then you will have couple of weeks to complete the task and submit it for a peer review. Details of that will follow in subsequent weeks.

Overview & Purpose

- 1) Complete the provide solution by implementing the four strategies for selecting in which block to place a new requested memory block namely
 - a) First-fit: select the first suitable block with smallest address.
 - b) Best-fit: select the smallest suitable block.
 - c) Worst-fit: select the largest suitable block.
 - d) Next-fit: select the first suitable block after the last block allocated (with wraparound from end to beginning).

Features to learn

- 1) Memory Allocation strategies
- 2) Implement them
- 3) Compare different strategies

Tasks

1. Follow the README file provided with the code in mandatory assignment 2 code.
2. Familiarize yourself with the provided assignment and the code

-
3. Implement the following
 - a. `initmem()`: Initialize memory structures.
 - b. `mymalloc()`: Like `malloc()`, this allocates a new block of memory,
 - c. `myfree()`: Like `free()`, this deallocates a block of memory,
 - d. `mem_holes()`: How many free blocks are in memory?
 - e. `mem_allocated()`: How much memory is currently allocated?
 - f. `mem_free()`: How much memory is NOT allocated?
 - g. `mem_largest_free()`: How large is the largest free block?
 - h. `mem_small_free()`: How many small unallocated blocks are currently in memory?
 - i. `mem_is_alloc()`: Is a particular byte allocated or not?
 4. You are not suppose to implement `malloc()` or `free()`, you can use/call them inside your `mymalloc()` and `myfree()` function. You just have to implement the different strategies such that when you call `malloc()` with that strategy it allocates the memory with that strategy.
 5. Second part of the assignment, **you will have to answer questions 1 to 10 from the README.txt**

