

CS 162 HW 4

[Library functions](#) / Introduction

Introduction

TABLE OF CONTENTS

- 1 [Compilation](#)
 - 2 [Skeleton](#)
 - 3 [Background](#)
 - a [Heap data structure](#)
-

In this portion of the assignment, you will be implementing your own memory allocator from scratch.

Compilation

To compile the code for this part:

```
cd ~/code/personal/hw-memory/mm_alloc
make
```

Skeleton

Code for this part can be found in the `mm_alloc` subdirectory of the `hw-memory` directory.

You will find a simple skeleton in `mm_alloc.c`. `mm_alloc.h` defines an interface with three functions: `mm_malloc`, `mm_free`, `mm_realloc`. You will need to implement these functions. **Do not change the headers of any of these!**

The given `mm_test.c` performs some sanity checks but is not an exhaustive test. We recommend you write some custom tests when testing locally by changing this file.

Background

The man pages for `malloc` and `sbrk` are excellent resources for this assignment. **Note that you must use `sbrk` to allocate the heap region. You are NOT allowed to call the standard `malloc/free/realloc` functions.**

You may want to refer back to the [first part of the homework](#) for information on how process memory works. In this part, you will be allocating blocks of memory in the mapped region of the heap and moving the break appropriately using the `sbrk` syscall. Initially the mapped region of the heap will have a size of 0.

Heap data structure

A simple memory allocator for the heap can be implemented using a linked list data structure. The elements of the linked list will be the allocated blocks of memory on the heap. To structure our data, each allocated block of memory will be preceded by a header containing metadata.

For each block, we include the following metadata:

`prev, next`

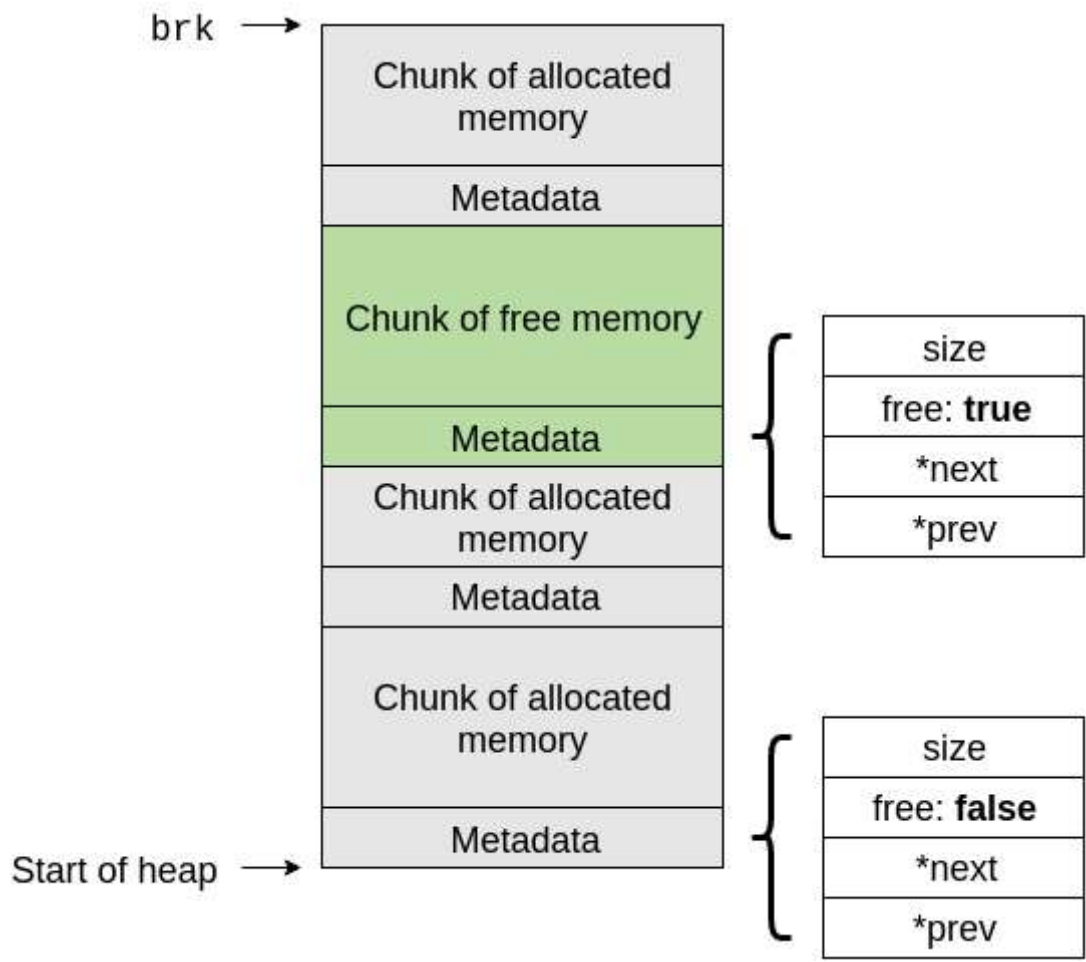
Pointers to metadata describing the adjacent blocks

`free`

Boolean describing whether or not this block is free

`size`

Allocated size of the block of memory



You might also consider using a [flexible array member](#) to serve as a pointer to the memory block.