

Memory Allocation Lab

This lab project addresses the various memory-management schemes in an operating system.

We model the memory managed in an operating system as a list of memory blocks. Each block of memory is modeled by a data structure of MEMORY_BLOCK defined below:

```
struct MEMORY_BLOCK {  
    int start_address;  
    int end_address;  
    int segment_size;  
    int process_id; //0 indicates a free block  
}
```

Thus, the operating system code can set up memory blocks for the available memory and keep track of which blocks are allocated to which processes. In particular, note that for each memory block, the system tracks the starting and ending addresses, along with the size of the block and the process to which the block is currently allocated. For free blocks, the process is indicated as 0.

The system manages the memory as an array of memory blocks, stored in an array. As memory is allocated and released, the list of blocks in the memory map changes.

In this lab project, we address the various ways in which memory can be allocated (based on different schemes - Best Fit, First Fit, Next Fit, Worst (Largest) Fit). In addition, we also develop a method to manage the release of memory blocks by processes.

For the purposes of these labs we define the NULLBLOCK as [start_address: 0, end_address: 0, segment_size: 0, process_id: 0]

Implement the following five methods in a file called **memory.c**. You should include the [oslabs.h](#) file.

