Data Structures
Assignment #1 (2014)
Memory Management ADT Overview

A major task of an operating system is to allocate blocks of memory to different processes so that each may execute concurrently on the system. A simple operating system is given a contiguous block of free memory from which to allocate space. This space can be broken down into smaller blocks. The operating system must keep track of all of the free space and all of the allocated space.

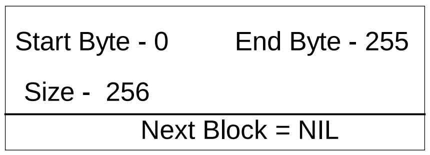
The operating system will use two lists to manage this memory.

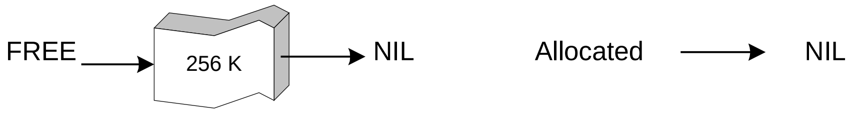
1. **FREE**: This list contains information on all free continuous blocks of memory.
2. **ALLOCATED**: This list contains information on all allocated blocks of memory.

Processes must be given contiguous blocks of memory. If there is no single free block of contiguous memory large enough to satisfy the request then it must be rejected. Note that the overall amount of free memory may be large enough if it was in a contiguous block.

To start, **FREE** will have one node that represents the whole of the memory. **ALLOCATED** will be an empty list.

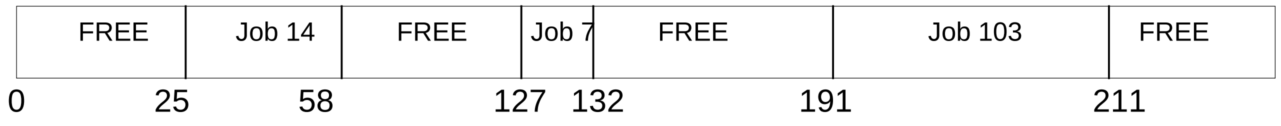
The nodes for these lists need to keep track of the size of the block and where the next node is
Assuming the Operating System is managing 256K of memory, the lists would start as:

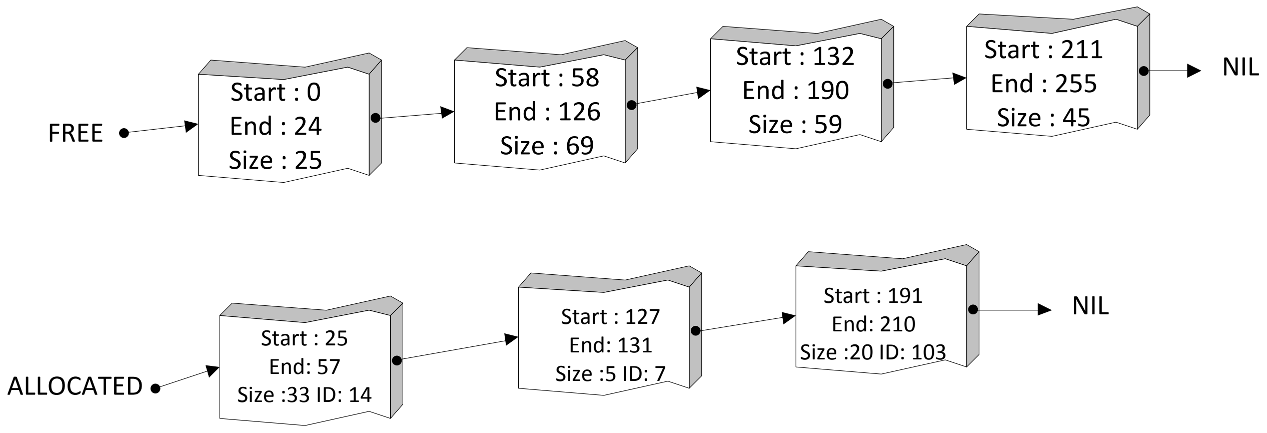




After several allocations and de-allocations things may look like this.

Logical Memory Space





Each node must hold its size, its starting byte and ending byte along with a pointer to the next node.

When allocation a portion of a block, the memory will be taken form the end of the block.

If a block is returned to the FREE list, it should be inserted in the correct order. If it can be combined with other blocks to create a larger free block it should be.

If a block is inserted into the ALLOCATED list it should be in order. It will also need to hold the Job # unlike the FREE list nodes.

The ADT should manage the free and allocated space. The user should be able to request memory blocks for jobs and de-allocate the memory blocks when they are done. The user should also be able to ask things like; How much free memory is there in the system? How much allocated memory is there in the system? What is the largest continuous block of free memory available? What is the total amount of memory being managed by the ADT? How much memory does a job have allocated?

The user should also be able to get a report with a listing of all jobs in the system with how much memory each has allocated and where that memory is. (A report that basically does a memory dump showing usage of all memory.)
==

PHASE I: Design (done in class)

A) Design the Memory Manager.

As a team we will develop the design for the memory manager program (ADT). We must make sure that it meets the requirements set out by our customer.

Deliverables:

1) A detailed description of the data structures and each function used as part of the ADT. We will not develop complete solutions to each function but will develop a few high level algorithms. We will develop the interface and pre and post conditions, etc..

B) Design Test Cases

As a team we will develop test cases to be used to test the memory manager ADT. These test cases will be used to test the final implementation of the memory manager.

Deliverables:

1) Black box test cases for selected methods of the memory manager
2) An overall black box test case for the memory manager ADT.

PHASE II: Implementation (Assignment #1)

A) Implement the Memory ADT design

Each of you will be tasked to implement the memory manager ADT design. You will be required to develop and implement the detailed algorithms needed. You can use the test cases we developed as a team to test your functions, etc.

B) Final Testing

You will use our team final test case to test your implementation.

Deliverable:

1) The source code of your memory manager implementation with our final test case as the main driver.

Note: The process we are using is not meant to represent the full Development Life Cycle. Many parts have been modified, condensed or just left out to protect the innocent. ☺