

ECE 354: Project 2 Part B

Group: ECE.354.S11-G031

Members: Ben Ridder (brridder), Casey Banner (cccbanne), David Janssen (dajjanss)

This document is concerned with the high level implementation of the specified primitives. As such, most of the low level back end is omitted for simplicity. Any kernel structures required to understand the primitive are described when the primitive is discussed.

RTX Initialization of Memory Management

The RTX initialization of memory management is called by `init()`. This function then uses the beginning of free memory to call `init_memory(void* memory_start)`. The pseudo code for this initialization can be seen in Listing 1. The data structure that is used to implement our memory management is known as a free list. A free list is a singly linked list that stores the address of the next free memory in the first four bytes of the current memory block. The only pointer pointer required is to the head of the free list. This allows the list to be traversed in $O(1)$ time. In order to prevent against allocating the same block multiple times and also double deallocation a bit field with a bit for each memory block was created. This bit field stores a value of 0 if the memory is free and a value 1 if it is currently allocated, this check can also be made in $O(1)$.

Listing 1: Pseudo code for RTX initialization of memory management

```
void init() {
    init_memory(void* memory_start)
}

void init_memory(void* memory_start) {
    declare local variables

    head of memory = start of free memory + (number of
        memory blocks - 1) * (size of memory block)

    iterate over all blocks in free list {
        store address of next free memory block in
        current block
    }

    initialize the memory allocation bit field to 0
}
```