Data Structures and Algorithms Report

Sarim Tahir

November 25, 2023

1 Data Structures

1. **Frame Status Structure (`struct framestatus`):**
   - Represents the status of a stack frame.
   - Attributes: function name, function address, frame address, and usage status.
   - Stored as a packed structure for optimized memory usage.
2. **Free List Structure (`struct freelist`):**
   - Describes a free memory region with start address, size, and pointer to the next free region.
   - Maintains a linked list of available memory regions.
3. **Allocation List Structure (`struct allocList`):**
   - Stores allocated memory details: start address, name, and pointer to next allocated memory.
   - Manages a linked list of allocated memory blocks.
4. **Memory Array (`char mem[MEMSIZE]`):**
   - Represents the entire memory space allocated for the program.
   - Used to simulate memory layout and store data structures and allocated memory blocks.

2 Algorithms and Operations

1. **Frame Management:**
   - `CF` (Create Frame), `DF` (Delete Frame): Manage stack frames.
2. **Data Storage on Stack:**

- `CI` (Create Integer), `CD` (Create Double), `CC` (Create Character): Allocate data on the stack.

3. **Heap Management:**
   - `CH` (Create Heap), `DH` (Delete Heap): Manage heap memory.
4. **Memory Management:**
   - `Expand`: Increase available space for stack or heap.
   - `CheckStack`: Check available space on the stack.
5. **Printing Memory Regions:**
   - `SM` (Show Memory): Print memory contents and regions.
6. **List Operations:**
   - `SetFree`, `SetAlloc`: Insert into free and allocation lists.
   - `DelAlloc`, `DelFree`: Delete from allocation and free lists.
7. **Input Handling and Control Flow:**
   - Parse user input commands and execute corresponding functions in a loop.