

Assignment - 6

Manual Memory Management

Group - 15

Utsav Mehta - 20CS10069
Sidharth Vishwakarma - 20CS10082
Pranav Kulkarni - 20CS30029
Swapnil Yasasvi - 20CS30054

Structure of Internal Page Table

- The approach for using pages is different from the one conventionally used.
- Each page contains exactly one linked list, with its size dynamically adjusted according to the size of the linked list.
- We also maintain the generated pages/linked lists in a set(as discussed in the Memory data structure later).
- These are used to keep a track of holes as well and to use the first-fit memory allocation algorithm for newly created linked lists.

Additional Data Structures functions used in the library

- The below-mentioned data structures and their corresponding attributes were used in the library.
 - Mem_block
 1. **base (uint64_t)**: base address for this memory block.
 2. **limit (uint64_t)**: total address space for this limit-bound memory block.
 3. **name (string)**: name of this memory block.
 4. **in_use (bool)**: boolean value to specify whether this memory block is in use or not. If not, then we will free this space appropriately.
 - Memory
 1. **mem_start (uint64_t)**: starting address for this memory segment.
 2. **mem_size (long)**: total address space for this memory segment.
 3. **blocks (set<mem_block>)**: set of all memory blocks in the given memory segment.
 4. **scope_stack(stack<map<>>)**: stack to store the symbol table entries for a scope, in the form of a string to mem_block map.
 - ListElement
 1. **val (int)**: the value of the doubly linked list node
 2. **next (ListElement*)**: Pointer to the next element of the doubly linked list.
 3. **prev (ListElement*)**: Pointer to the previous element of the doubly linked list.

Impact of FreeElem() on mergesort:

- To get the memory footprint, we need to get the maximum resident set size(RSS).
- We used the getrusage function for memory footprint and runtime calculation.
- With FreeElem():
 - Memory Footprint (in MB): 2.4, During execution maximum of 2.4 MB is used by the program.
 - Runtime(in milliseconds): 1037.78
- Without FreeElem():
 - Memory Footprint (in MB): 20.027, During execution maximum of 20.027 MB is used by the program.
 - Runtime(in milliseconds): 53837.6

Maximal and minimal performance for the given code structure

- The performance will be maximized when the given code structure includes the use of free Elem and it substantially improves the performance in comparison to no use of freeElem
- The performance will be minimized when no freeElem is done

Locks used in library

- No Locks were used during the creation of this library.
- This approach was chosen since there is no thread/ process which updates the content of the internal data structures or the state of the allocated memory, apart from the functions present in the library.