

Group #17: mjt207, kl624, xl234

iLab machine used: man

### **Malloc**

Each block of memory is stored as a node in a linked list. Each node is a multiple of the system's page size, with meta data, a structure called memblock, stored at the front of the memory address space. Meta data includes information such as which thread owns the block, if it is free, the size, and so on. When memory is requested the list is searched for a free block that has enough free space for the requested size. If no block is found, and there is enough space in the 8MB to allocate a new block of the requested size, a new block is created of the requested size rounded up to a multiple of the system's page size. In the case that the 8MB is full, the list is searched for page(s) to evict to the swap file and that newly freed space will be returned. The evict page method is to search the list for a big enough block that is being used by a thread that is not currently running, and then copy it to the swap file. If no block can be found, that means the currently running thread is using the entire 8MB space and NULL is returned.

### **Free**

When an address is requested to be freed, the address is taken and subtracted by the size of memblock and then casted as a memblock pointer. The free flag is then set to true allowing that block to be used in a future malloc request.

### **Changing Threads**

The memory manager is aware of the currently running thread kept in the current\_thread variable. Each time the thread manager changes threads, it calls the set\_current\_thread(int tid) function. The memory manager then goes through the list and mprotects every block that is not owned by the current thread, and un-mprotects every block that is owned by the current thread.

### **Swap File**

A separate linked list is used to keep track of the contents of the swap file using the swapfilemeta structure. This list works similarly to the memory block list as each node holds the location in the swap file, the size, and which thread owns it. This way, the list can be used to easily know what pages are in the swap and list's data to easily read/write pages to and from the swap file.

### **Shalloc**

4 pages are reserved as shared memory. This works the same as malloc does, except which thread owns which page is disregarded. These pages do not get swapped out to the swap file, and are not mprotected, so any thread can access any page in this space even if they were not the one who allocated it. If all 4 pages are full, NULL is returned.