

# CS518

## Operating System

### [A1 Report]

Hittishi Kurlagunda  
Sowmiyanarayan Selvam

[Net ID: hk919]  
[Net ID: ss4370]

#### **Objective:**

To implement a memory allocator using the malloc library as the template. Memory management is done with a small sample of user memory.

#### **FUNCTIONS**

##### **Main Functions:**

##### **1) void \* umalloc(size\_t size, char \*file, int line)**

Checks whether the requested memory is available along with the metadata size for storing that particular data in user memory.

##### **2) void ufree(void \*to\_free, char \*file, int line)**

Frees the blocks located at the pointer provided. If already freed it returns an error. Calls merge free blocks after for smooth memory management.

##### **Helper Functions:**

##### **3) char \* find\_first\_block(size\_t size)**

Modification of the first fit algorithm. It finds the first block with the same or greater memory required for allocation.

#### **4) bool is\_block\_free(block \*block)**

Returns the status of the block whether it is free or not.

#### **5) char \* next(char \*curr\_block)**

Checks if there is any next block and if available return the pointer to that block.

#### **6) void \* break\_block(char \*curr\_block, size\_t size)**

When there is excess memory in the block that has been assigned, the block is broken into 2 one in which the allocation will be done. The other would be a new free block with an updated size for further allocations.

#### **7) void merge\_free\_blocks()**

Traverses throughout the user memory and merges adjacent free blocks for future allocations.

#### **8) void traverse\_blocks()**

Traverses the memory and pretty prints it. Used predominantly while debugging the code.

### **Block Structure**

```
typedef struct block
{
    unsigned short block_size;
    bool is_free;
}block;
```

The metadata of a memory block is defined by this structure. It uses 4 bytes for block\_size and 1 byte for bool so the metadata for a particular block is 5 bytes.

### **ERROR HANDLING**

Handles the boundary cases and makes sure that the memory allocator doesn't stop and causes issues. This is cardinal because in the absence of the memory allocator a system can crash easily.

## Malloc:

- 1) When there is a request to allocate more memory than is available in the user memory space.
- 2) When there are no more free blocks to allocate.

## Free:

- 1) The memory size requested to free is either more than the available user memory space or exceeds the range.
- 2) The pointer was freed already.
- 3) The memory hasn't been allocated at all(as in it was not allocated using umalloc).

## MEMGRIND

### 0. Consistency

```
FIRST
initialized
156565529
1:156565512 free.
*****
SECOND
156565529
1:156565512 free.
*****
```

### 1. Maximization

```
2,156565529 - 156565531
4,156565529 - 156565533
8,156565529 - 156565537
16,156565529 - 156565545
32,156565529 - 156565561
64,156565529 - 156565593
128,156565529 - 156565657
256,156565529 - 156565785
512,156565529 - 156566041
1024,156565529 - 156566553
2048,156565529 - 156567577
4096,156565529 - 156569625
8192,156565529 - 156573721
16384,156565529 - 156581913
32768,156565529 - 156598297
65536,156565529 - 156631065
131072,156565529 - 156696601
262144,156565529 - 156827673
524288,156565529 - 157089817
1048576,156565529 - 157614105
2097152,156565529 - 158662681
4194304,156565529 - 160759833
8388608,156565529 - 164954137
16777216,156565529 - 173342745
Error: memgrind.c in line 38 not enough memory.
8388608
Error:memgrind.c in line 56, requested more memory than available.
Error:memgrind.c in line 56, requested more memory than available.
max:9437184
```

## 2. Basic Coalescence

```
9437184
1/2:156565529 - 161284121
1/4:161284138 - 163643434
1:156565529 - 166002713
```

## 3. Saturation

```
Error:memgrind.c in line 103, requested more memory than available.
saturation:9495436
last:167051248
```

## 4. Time Overhead

```
last:167051248
Maximum time overhead:0.000612
```

## 5. Intermediate Coalescence and clearing using makefile

```
1:156565512 free.
*****
1:156565529 - 166002713
Finished%
```