

15213-ICS-Malloc

2017年8月28日 15:32

任务：

完成一个malloc函数。

Int mm_init(void)

做一些必要的出事前工作，比如初始化堆区。

Void mm_malloc(size_t size)

至少需要分配 size 这么大的空间（可能因为对齐的原因会更大一点，8 byte 对齐），不能超出堆的范围，也不能覆盖其他已分配的区域。

Void mm_free(void ptr)

通过指针释放空间。

Void mm_realloc(void ptr, size_t size)

重新分配，根据传入指针的不同，有不同的表现。

在正式写之前，先确定一堆宏定义。

```
/* single word (4) or double word (8) alignment */
#define ALIGNMENT 8
```

确定对齐大小。

```
/* rounds up to the nearest multiple of ALIGNMENT */
#define ALIGN(size) (((size) + (ALIGNMENT-1)) & ~0x7)
```

调整size，使得size%8==0，对齐原则。

```
#define SIZE_T_SIZE (ALIGN(sizeof(size_t)))
```

确定size_t的大小参数。

```
#define WSIZE 4
```

Word size = 4

```
#define DSIZE 8
```

Double size = 8

```
#define INITCHUNKSIZE (1<<6)
```

初始化块大小(2^6==64)

```
#define CHUNKSIZE (1<<12) //+(1<<7)
```

默认的扩展堆大小

```
#define LIST 20
```

分离列表大小

```
#define REALLOC_BUFFER (1<<7)
```

重新分配的缓存大小 (2^7)

```
#define MAX(x, y) ((x) > (y) ? (x) : (y))
#define MIN(x, y) ((x) < (y) ? (x) : (y))
```

```
// Pack a size and allocated bit into a word
#define PACK(size, alloc) ((size) | (alloc))
```

```
// Read and write a word at address p
#define GET(p) (*(unsigned int *) (p))
#define PUT(p, val) (*(unsigned int *) (p) = (val) | GET_TAG(p))
#define PUT_NOTAG(p, val) (*(unsigned int *) (p) = (val))
```

在地址p处读取或者写入一个word。

```
// Store predecessor or successor pointer for free blocks
#define SET_PTR(p, ptr) (*(unsigned int *) (p) = (unsigned int) (ptr))
```

存储空块的前向或者后向指针。

```
#define GET_SIZE(p) (GET(p) & ~0x7)
```

出去最后3位j就是p的大小

```
#define GET_ALLOC(p) (GET(p) & 0x1)
```

当前block是否已经allocated。

```
#define GET_TAG(p) (GET(p) & 0x2)
```

从最后2位获取tag。

```
#define SET_RATAG(p) (GET(p) |= 0x2)
```

设置reallocated tag。

```
#define REMOVE_RATAG(p) (GET(p) &= ~0x2)
```

删除reallocated tag。

```
// Address of block's header and footer
#define HDRP(ptr) ((char *) (ptr) - WSIZE)
#define FTRP(ptr) ((char *) (ptr) + GET_SIZE(HDRP(ptr)) - DSIZE)
```

当指针指向一个block或者第一个有效加载区的时候，返回对应的header 和footer指针。

```
// Address of free block's predecessor and successor entries
#define PRED_PTR(ptr) ((char *) (ptr))
#define SUCC_PTR(ptr) ((char *) (ptr) + WSIZE)
```

返回对应位置的next和previous。

```
// Address of free block's predecessor and successor entries
#define PRED_PTR(ptr) ((char *) (ptr))
#define SUCC_PTR(ptr) ((char *) (ptr) + WSIZE)
```

返回free blocks的next 和previous。

```
// Address of free block's predecessor and successor on the segregated list
#define PRED(ptr) (*(char **)(ptr))
#define SUCC(ptr) (*(char **)(SUCC_PTR(ptr)))
```

返回segregated list的next 和 previous。