

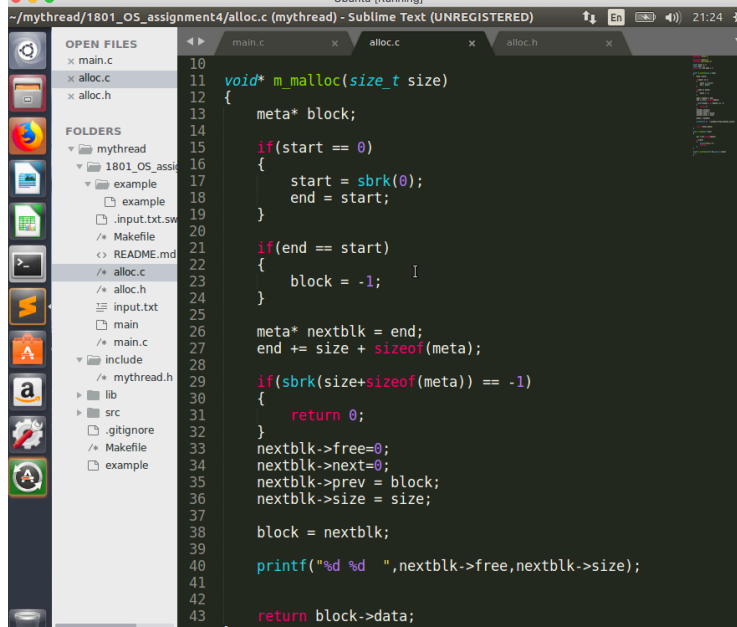
운영체제 과제 4 레포트

201521082 미디어학과 김우정

1. 이번 과제는 메모리 할당자를 구현해보는 과제였다.

시간 관계 상 이번 과제에서는 malloc만을 구현할 수 있었다.

다음은 malloc 함수 코드이다.



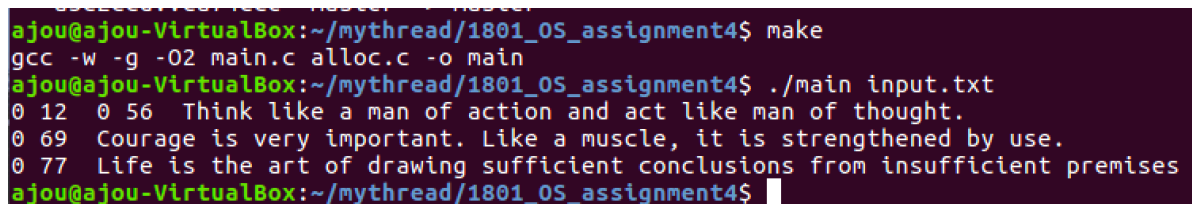
```
10
11 void* m_malloc(size_t size)
12 {
13     meta* block;
14
15     if(start == 0)
16     {
17         start = sbrk(0);
18         end = start;
19     }
20
21     if(end == start)
22     {
23         block = -1;
24     }
25
26     meta* nextblk = end;
27     end += size + sizeof(meta);
28
29     if(sbrk(size+sizeof(meta)) == -1)
30     {
31         return 0;
32     }
33     nextblk->free=0;
34     nextblk->next=0;
35     nextblk->prev = block;
36     nextblk->size = size;
37
38     block = nextblk;
39
40     printf("%d %d ",nextblk->free,nextblk->size);
41
42
43     return block->data;
```

함수 실행 화면은 아래 그림과 같다.

실행 시 주의할 점은 main의 인자로 input.txt 값을 받아와야 실행이 제대로 된다.

Input.txt는 example 파일의 첫번째 input 값을 그대로 옮겨왔다.

참고로 0 12는 시스템의 malloc을 호출하면 프린트 되는 값이다. 인풋의 글들을 시스템의 메모리 크기로 할당해주기 위해 기존의 malloc을 일부 사용하였다.



```
ajou@ajou-VirtualBox:~/mythread/1801_OS_assignment4$ make
gcc -w -g -O2 main.c alloc.c -o main
ajou@ajou-VirtualBox:~/mythread/1801_OS_assignment4$ ./main input.txt
0 12 0 56 Think like a man of action and act like man of thought.
0 69 Courage is very important. Like a muscle, it is strengthened by use.
0 77 Life is the art of drawing sufficient conclusions from insufficient premises
ajou@ajou-VirtualBox:~/mythread/1801_OS_assignment4$
```

Malloc을 구현하고 나서 구현 원리를 어느 정도 파악하였지만 나머지는 시간이 부족하여 구현하지 못하였다.

2. Lessons

이번 과제를 통해 메모리 할당과 포인터 위치 설정을 해보았다. 메모리 할당 함수인 malloc을 직접 구현해보니 오랜만에 자료구조 linked list 공부를 하는 느낌이 들었다. 메모리 개념이 아

직 잘 잡히지 않은 상태에서 시작하고, 구현 도중에 메모리 관련 공부를 추가적으로 더 한 뒤
에서야 코드를 어느 정도 건들 수 있었다.

3. Feedback

프로젝트 제출 기간 및 시험 기간에 주어진 과제인데 코드가 뼈대 하나 없이 백지로 주어진
것이 과제를 포기하게 만드는 요인이었다. 이전 과제 1, 2, 3에서는 뼈대가 충분히 주어져서
pdf를 보면서 공부하고 스스로 구현할 수 있는 방향을 제시해주었다. 하지만 이번 과제는 베
이스 코드가 하나도 없어서 인터넷으로 관련 코드를 찾아보면서 베껴가면서 원리를 익혔다.
투자한 시간에 비해 아웃풋이 잘 나오지 않은 것 같아서 시간낭비 한 기분이 든다.