**운영체제**

**과제4**

**1801 OS assignment4**

소프트웨어학과

201320901이관주

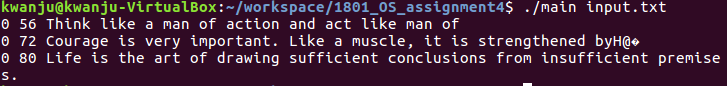
**Work Description**

이번 과제에서는 메모리를 동적 할당해주는 malloc, realloc, 그리고 free에 대한 함수를 직접 구성해보았다. 이 함수들을 구성할 때 sbrk()와 brk() 시스템콜을 사용했다. Meta data는 구조체를 만들어 사용하였으며 이를 관리하기 위해 double linked list를 사용했고 이 구조체는 size크기, free여부, 그리고 데이터를 챙기는 char포인터를 포함하고 있다. Malloc, free, realloc 이외에 사용된 함수는 search로 fit을 결정해주는 것이다. Fit은 external int로 설정해 main.c에서 0, 1, 2로 값을 결정한다. 이 fit에 따라 first, best, worst의 3가지 방식으로 메모리를 할당해준다.

Main.c 함수는 파일 입력을 받아 그를 토큰으로 나눠 명령어를 구분하는 파싱에 대한 코드를 짜고, 입력된 횟수만큼 메모리 할당 과정을 반복적으로 진행할 수 있도록 했다.

**Test**

해당 폴더에서 미리 만들어져 있는 make명령어를 사용해 main 파일을 만들어서 테스트를 했다. 실행은 example에서 첫번째 예시를 가져온 input.txt를 통해 한다. Make로 main 실행파일을 만들고 ./main input.txt 명령어를 통해 실행시켜 테스트해 보았다.



그리고 다음과 같은 결과가 나오는 것을 확인했다. 2번째 줄에서 뒷부분의 글자가 깨지긴 했지만 무슨 문제인지는 모르겠지만 중요한 문제는 아닌 것 같아서 무시했다. 하지만 example에 있던 다른 예시에서는 core dump가 발생했는데 어디서 에러가 난 것인지 어떤 것이 문제인지 확인하기가 너무 어렵고 시험때문에 시간이 부족해 이를 수정할 수 없었다.

**Lessons**

Memory allocation의 동작 방식과 원리를 배울 수 있어서 좋은 경험이 되었다.

**Feedback**

저번 과제에서는 어려운 과제였지만 힌트가 많이 있었고 스켈레톤 코드와 주석으로 어느정도 설명이 있어서 시작할 때 어떤 방식으로 코드를 구성할 지 좋은 가이드가 되었지만, 이번 과제에서는 그런 것이 하나도 없고 그저 어떤 기능을 구현하라는 내용과 int main과 헤더만 달린 아무것도 없는 코드에 생으로 작업해야 했다. 그래서 처음에 코드를 어떻게 구성해야 될지 너무 감도 안 잡히고 어려웠다. 그래도 기능적인 쪽인 alloc.c 코드를 짜기 시작하면서는 그렇게 어렵지 않게 했던 것 같다. 하지만 더 어려웠던 것은 main.c를 짤 때였다. 파일을 받고 그 내용을 읽어 파싱해 해당되는 명령어에 맞춰 실행될 수 있게 하는 내용인데 이부분을 처음에 어떻게 만들어야 할 지 몰라서 많은 시간을 고민했던 것 같다. 또 이 과제 작업중 가장 많은 시간을 차지한 것은 오류 수정이다. 이번 과제는 오류체크를 하기가 너무 어려웠던 것 같다. 4학점 수업이 아닌 만큼 기본적인 코드의 구조를 어느정도 알려주고 테스트를 할 수 있는 코드를 기본적으로 줘야한다고 생각한다.