**OS\_Assignment\_4**

**-memory allocation-**

**학과 : 소프트웨어학과**

**학번 : 201620882**

**이름 : 최민정**

이번 과제는 다른 과제에 비해 개념적인 부분을 이해하는데 덜 어려웠고, 설계를 하는데 이전만큼 많은 시간이 걸리지는 않았다. 기본적으로 메모리 할당에 대한 이해가 필요했고, malloc, realloc, free가 정확히 어떤 역할을 하는지 알아보는 시간을 가졌다. 사실 c를 배울 때, 동적할당 부분은 마지막 파트이다 보니깐 다른 부분에 비해 소홀해지게 되었었는데 이번 기회를 통해 다시금 공부하는 기회를 가지게 되었다. 또한 동적할당이 왜 필요한가에 대해서도 더 잘 알 수 있는 기회였다.

이번 과제에서 메모리 할당은 LinkedList로 진행하였다. Meta Structure을 만들어서 이안에 이전의 structure과 이후의 structure을 연결하여, 현재의 structure하나만 가지고 있다면 이후의 structure을 줄줄이 알 수 있는 방식으로 진행하였다. 이 LinkedList를 통해 어떻게 메모리 구조가 만들어지는지 알 수 있다.

* Malloc

: malloc은 메모리를 할당해주는 명령어로 first fit, best fit, worst fit으로 나뉘게 된다.

* Firstfit

: 가장 처음으로 남아있는 메모리에게 할당하는 방식이다. 이를 위해서는 시작부터 free가 1인 그리고 size가 넣어야하는 size보다 큰 것을 찾는다. 처음 찾게 되는 곳을 temp로 넣어주고, 이를 기점으로 size만큼 잘라주고, 새로 만드는 곳을 중간에 할당해주는 식으로 진행 하였다.

* Bestfit

: size가 주어지면 이 size보다 크지만 size에 가장 가까운 크기를 가진 free가 1 인 곳에 할당해준다. 이후에는 firstfit과 마찬가지로 진행한다.

* Worstfit

: free가 1인 메모리 중에 size가 가장 큰 곳에 할당을 해주는 것이다. 이후는 firstfit과 마찬가지로 진행한다.

* Exception

: 이외에 예외가 하나 추가 되었다. 할당을 처음 할 때이다. 이때는 정해진 주소도 없고, 어떠한 structure도 존재하지 않기 때문에 처음 설정해줘야 한다. 주소가 존재하지 않기 때문에 sbrk를 통해 주소를 처음 할당해준다. 그리고 첫 생성이기 때문에 이전의 structure과 이후의 structure은 NULL값을 넣어준다.

* Realloc

: Realloc은 할당된 메모리의 크기를 조정해주는 것이다. Realloc 해야하는 메모리 이후의 메모리가 확보가 된다면 그 메모리를 segmentation하여 메모리 크기를 늘린다. 하지만 만약 충분하지 않다면 다시 malloc을 하여 다시 메모리 할당을 해주어야 한다. 만약 이 malloc 조차 되지 않는다면 기록만 남긴 채 누락이 된다. 하지만 코드 내에서 초과가 될 정도의 메모리를 사용하지 않기 때문에 구현하지는 않았다.

* Free

: free는 할당해준 메모리를 해체해주는 것이다. 단지 structure에 있는 free를 0에서 1로 바꿔주는 것만을 하는 것이 아닌 앞뒤의 메모리가 만약 free상태라면 split까지 해주여야 한다. 그래서 나는 만약 앞뒤의 메모리가 NULL이 아니고, free가 1이라면 각각의 size를 합쳐주고 메모리들의 앞뒤의 structure을 저장하여 prev\_memory와 next\_memory에 담아 새로운 structure을 만들어 free를 합친 memory를 다시 생성해주었다.

* Meta structure
* Prev\_memory : 해당 메모리의 이전 메모리를 가진 structure
* Next\_memory : 해당 메모리의 이후 메모리를 가진 structure
* Free : free면 1, 할당 되었으면 0
* Size : 메모리의 크기
* Commit : 메모리가 가진 내용(문장)
* Fit : malloc 혹은 realloc을 해줄 때, 할당해주는 방식
* head : 할당이 되어 있는 첫번째 구조체
* first : 할당 여부와 관계 없이 첫번째 구조체

구현을 다했으나 Segmentation fault로 인해 실행해보지는 못하였다. 아직도 동적할당이나 null값이 접속하는 것에 대한 주의가 부족한 것 같다. 다음 번에 linux로 코딩을 하는 기회가 되면 더 신경을 많이 써야겠다. 마지막 과제는 적어도 이해를 하고 구현을 해볼 수 있어서 기분이 좋았다. 이 정도의 난이도라면 시간이 되고 일정 정도의 개념 이해가 있다면 할 수 있다고 생각이 되어 딱 좋다고 생각한다. 그러면 안되지만 만일 이 과목을 다시 듣는다면 그때는 이 정도의 난이도였으면 좋겠다.