

Multicore Programming Project 3

담당 교수 : 박성용 교수님

이름 : 이수빈

학번 : 20200422

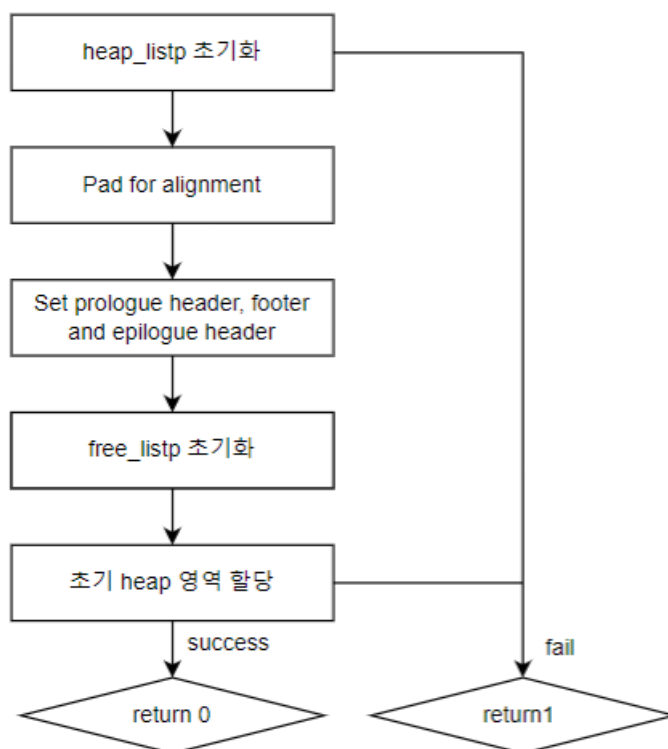
1. 개발 목표

나만의 Malloc library를 구현한다. malloc, free, realloc 함수를 구현할 것이며, utilization과 throughput 측면에서 성능이 측정된다. Explicit list를 사용할 것이고, 해당 구현을 통해 동적으로 할당 혹은 해제되는 메모리의 동작에 대해 이해하는 것이 목표이다.

2. Flow Chart

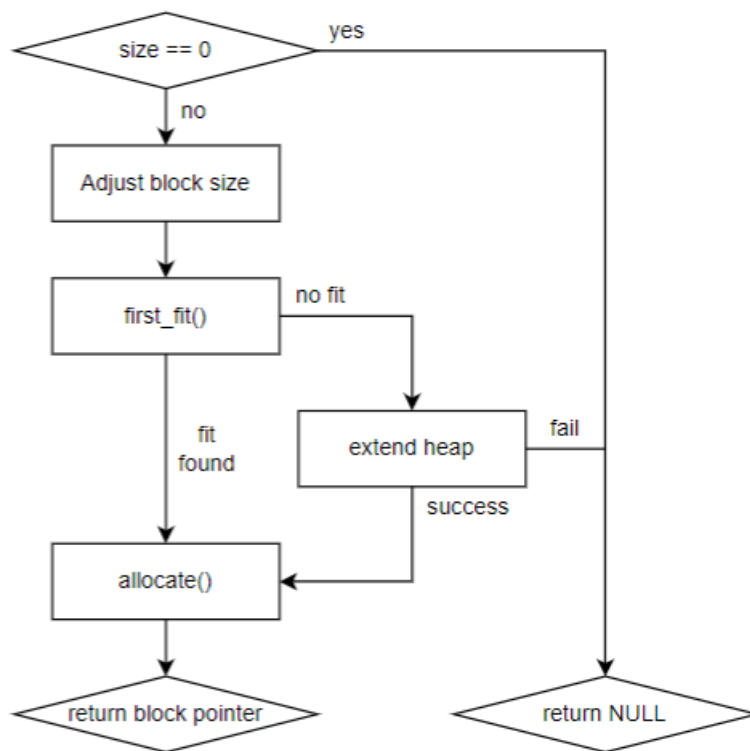
A. mm_init

heap 영역의 초기 상태를 구성한다.



B. mm_malloc

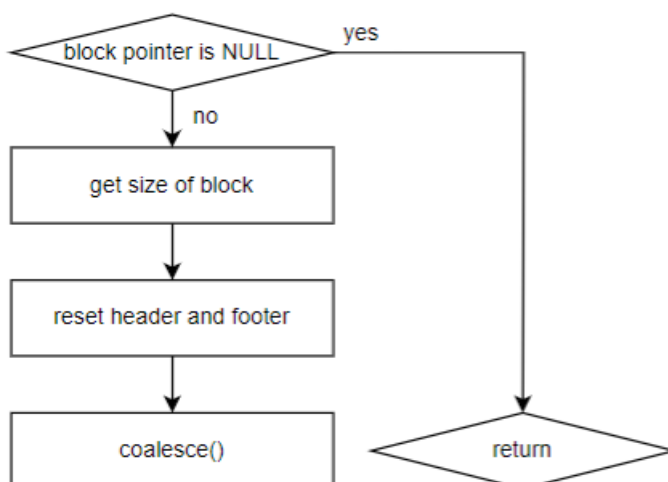
가상 heap 영역에 메모리를 동적으로 할당한다.



C. mm_free

할당한 메모리를 해제한다.

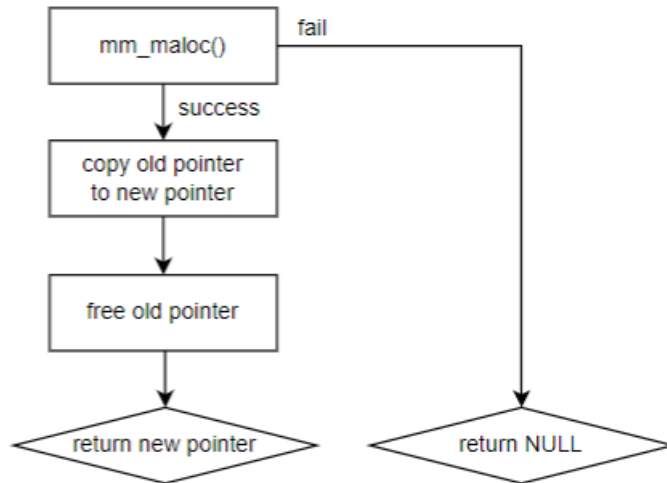
< mm_free >



D. mm_realloc

동일 변수에 대한 메모리 재할당 요청을 처리한다.

< mm_realloc >



3. Description

✓ Subroutines

함수명	설명	쓰임
first_fit	Free list를 처음부터 탐색하며 할당 가능한 block을 찾는다. 적합한 block이 없는 경우 NULL을 반환한다.	- mm_malloc : allocation 요청에 맞게 할당 가능한 block을 찾는다.
extend_heap	초기 heap 생성 이후에 빈 공간이 부족할 때 자동으로 호출되어 heap 영역을 확장한다.	- mm_init : 초기 heap 영역을 할당한다. - mm_malloc : first_fit 실패 시 heap 영역을 확장하여 할당 공간을 확보한다.
coalesce	이전 블록과 다음 블록을 확인하여 가능한 경우 통합한다. 이를 통해 외부 단편화를 줄일 수 있다.	- mm_free : 할당 해제 후 coalesce 작업을 진행한다.
allocate	주어진 block을 사용 가능한 크기로 나누고, 필요한 경우 나머지 부분을 새로운 block으로 분할한다.	- mm_malloc : 요청에 맞도록 block을 할당한다.
insert	주어진 block을 free list에 삽입한다.	
delete	주어진 block을 free list에서 제거한다.	

✓ Global variables

- heap_listp

mm_init 함수에서 `heap_listp = mem_sbrk(8 * WSIZE);` 로 초기화되며

가상 heap memory의 시작 주소를 가리키는 포인터로 쓰인다.

- free_listp

mm_init 함수에서 header와 footer 등을 세팅한 이후에, `heap_listp`에 그만큼을 더한

`free_listp = heap_listp + 2 * WSIZE;` 로 초기화되며

`free_listp`의 시작 주소를 가리키는 포인터로, 사용 가능한 block을 관리하는 데 쓰인다.

4. 성능 평가 결과

```
cse20200422@cspiro:~/mp$ ./mdriver
[20200422]::NAME: Subin Lee, Email Address: leechlo71@sogang.ac.kr
Using default tracefiles in ./tracefiles/
Perf index = 42 (util) + 40 (thru) = 82/100
```

Utilization과 throughput 측면에서 성능을 측정하였을 때, 어느 한 측에 치우침 없이 균등한 결과가 나왔다. 이를 통해 공간 활용도와 속도 모두 고려된 allocator임을 알 수 있었다.