Lab 12 Malloc Lab Report

20220665 유영준

Lab12에서는 Dynamic Memory Allocation에 대해 배웠다. malloc, free와 같은 메모리 할당 관련 함수의 동작에 대해 배우고 이를 구현하는 방법 중 하나로 implicit free list에 대해 배웠다. 이는 memory block을 찾기 위해 만든 doubly-linked list로, free에는 O(1), malloc에는 O(n)의 실행속도가 소요된다. 이와 관련하여 malloc, free 함수를 구현하는 것이 과제로 나왔다. 각 함수의 구현 방법은 다음과 같다.

1. mm_init

- mm_init 함수는 memory allocator를 초기화하는 함수로, heap의 초기 구조를 설정하고 header와 footer를 초기화하는 역할을 한다.
- Global static 변수로 선언한 heap_listp가 heap의 시작 위치를 가리키고 있을 때 mem_sbrk 함수를 사용하여 초기 공간을 설정한다. 만약 mem_sbrk 호출을 실패하면 -1을 반환하여 오류를 나타낸다.
- 첫 word 공간은 alignment를 위한 공간, 두 번째와 세 번째 word 공간은 Prologue header, Prologue footer를 위한 공간으로, double word의 크기를 가지고 할당된 상태로 표시한다. 네 번째 word 공간은 Epilogue header를 위한 공간으로, 크기는 0이고 할당된 상태로 표시한다.
- extend_heap 함수를 호출하여 heap을 CHUNKSIZE 바이트만큼 확장한다. 이는 초기 free block을 생성하는데 사용된다. 실패할 경우 -1을 return한다.
- 초기화가 성공적으로 완료되면 함수는 0을 반환한다.

2. extend_heap

- extend_heap 함수는 memoy allocator의 heap을 확장하는 데에 사용되는 함수로, 인자로 받은 word size만큼 heap에 추가 공간을 만들어 확장된 부분에 header와 footer를 초기화한다.
- 인자로 받은 word size가 홀수인 경우 짝수로 만들어 alignment를 맞춘 뒤, size 변수에 저장한다.
- mem_sbrk 함수를 사용하여 size만큼 heap을 확장한다. 호출을 실패하면 NULL을 반환한

다.

- 확장한 free block의 header와 footer를 초기화한다. 또한 새로운 epilogue header를 생성하여 heap의 끝을 나타낸다.
- 마지막으로 coalesce 함수를 호출하여 전 block이 free인 경우 두 free block을 합친다.

3. coalesce

- coalesce 함수는 앞과 뒤에 free block이 존재하면 이와 연결하여 포인터를 반환하는 함수이다.
- 먼저 prev_alloc, next_alloc 변수에 앞, 뒤 block이 할당되었는지 여부를 저장하고, 이를 활용하여 4개의 case로 나누어 coalesce를 진행한다.
- 4가지 case의 처리 방법은 다음과 같다.
  i. Case 1: 앞, 뒤 block이 모두 할당된 경우 현재 block을 그대로 반환한다.
  ii. Case 2: 앞 block만 할당되고 뒤 block은 free인 경우 현재 block과 뒤 block을 합친 후 합친 block의 새로운 크기로 현재 block의 header와 뒤 block의 footer를 업데이트 해준다.
  iii. Case 3: 뒤 block만 할당되고 앞 block은 free인 경우 현재 block과 앞 block을 합친 후 합친 block의 새로운 크기로 앞 block의 header와 현재 block의 footer를 업데이트 해주고 포인터는 이전 block을 가리키도록 한다.
  iv. Case 4: 앞, 뒤 block 모두 free인 경우 앞, 뒤 block과 현재 block을 모두 합친다. 합친 block의 새 크기로 앞 block의 header와 뒤 block의 footer를 업데이트 해주고 포인터는 이전 block을 가리키도록 해준다.
- 각 case에 대해 header, footer와 포인터를 변경해준 뒤 bp를 반환한다.

4. find_fit

- find_fit 함수는 first search 방법으로 인자로 받은 size보다 크기가 큰 size의 block을 찾아 순차적으로 탐색한다.
- heap의 시작 공간인 heap_listp부터 각 block의 header를 통해 해당 block의 size와 할당되었는지 여부를 확인한다.
- 할당되지 않았고 인자로 받은 size보다 크기가 크거나 같은 경우 해당 block의 포인터를

반환한다.

- 만약 해당 조건을 만족하는 block을 찾지 못했을 경우 함수는 NULL을 반환한다.

5. place

- place 함수는 free block에 해당 block을 넣은 후 남은 공간을 split하는 함수이다.
- 먼저 현재 block의 크기를 csize 변수에 저장한 뒤 현재 block의 크기에서 인자로 받은 크기를 뺀 값이 2 * DSIZE보다 크거나 같다면 split을 수행한다.
- 현재 block에서 인자로 받은 크기만큼 할당을 수행하고 남은 부분을 새로운 free block으로 split한다. 분할된 새로운 free block은 현재 block 다음에 위치하며, 크기와 header, footer를 새롭게 설정한다.
- 만약 split할 공간이 없는 경우 현재 block 전체를 할당 상태로 설정한다.

6. mm_malloc

- mm_malloc 함수는 인자로 받은 크기의 memory block을 할당한다. block의 크기를 조정하여 alignment를 맞추어 적절한 free block을 찾거나 heap을 확장하여 메모리를 할당한다.
- 인자로 받은 크기가 0인 경우 NULL을 반환하고, 메모리 할당을 수행하지 않는다.
- 인자로 받은 크기가 DSIZE 보다 작거나 같은 경우 최소 크기는 2*DSIZE 만큼의 크기가 할당된다. 그보다 큰 크기의 경우 DSIZE의 배수가 되도록 크기를 조정한다.
- 조정된 크기에 맞는 free block을 찾기 위해 find_fit 함수를 호출한다. 적절한 block을 찾은 경우 place 함수를 호출하여 해당 block에 메모리를 할당하고 return한다.
- 만약 적절한 block을 찾지 못한 경우 extend_heap 함수를 호출하여 heap을 확장한 뒤 새로 확장한 공간에 place 함수를 호출하여 메모리를 할당한다. 이때 확장하는 크기는 asize와 CHUNKSIZE 중 더 큰 값으로 한다.

7. mm_free

- mm_free 함수는 인자로 받은 포인터가 가리키는 메모리 block을 해제한다.
- block의 size를 header를 통해 가져온 뒤 block의 header와 footer를 업데이트하여 free

상태로 만든 뒤 coalesce 함수를 호출하여 앞 또는 뒤에 free block이 존재할 경우 합친다.

8. mm_realloc

- mm_realloc 함수는 기존에 할당된 메모리 block의 크기를 변경한다. 새로운 크기에 맞춰 메모리를 재할당한 후 기존 데이터를 새로운 위치로 복사한 뒤 기존 메모리를 해제한다.
- 만약 ptr이 NULL인 경우 mm_malloc(size)와 동일하게 작동한다. 만약 size가 0인 경우 mm_free(ptr) 함수와 동일하게 작동하여 기존 메모리를 해제한다.
- mm_alloc 함수를 호출하여 인자로 받은 크기만큼의 새로운 메모리 block을 할당한다. 할당에 실패할 경우 NULL을 반환한다.
- 기존 block에서 새로운 block인 newptr로 데이터를 복사한다. 복사할 데이터의 크기는 기존 block의 크기와 인자로 받은 크기 중 작은 값을 사용한다.
- 데이터 복사 후 mm_free를 호출하여 기존 메모리 block을 해제한다.