

Linker Lab Report

-2019-17577 경제학부 정지후-

1. 실행 결과

1.1 Part1

-test1

```
[0001] Memory tracer started.  
[0002]         malloc( 1024 ) = 0x56041211c2d0  
[0003]         malloc( 32 ) = 0x56041211c6e0  
[0004]         malloc( 1 ) = 0x56041211c710  
[0005]         free( 0x56041211c710 )  
[0006]         free( 0x56041211c6e0 )  
[0007]  
[0008] Statistics  
[0009]   allocated_total      1057  
[0010]   allocated_avg        352  
[0011]   freed_total         0  
[0012]  
[0013] Memory tracer stopped.
```

-test2

```
[0001] Memory tracer started.  
[0002]         malloc( 1024 ) = 0x5603d0db32d0  
[0003]         free( 0x5603d0db32d0 )  
[0004]  
[0005] Statistics  
[0006]   allocated_total      1024  
[0007]   allocated_avg        1024  
[0008]   freed_total         0  
[0009]  
[0010] Memory tracer stopped.
```

-test3

```
[0001] Memory tracer started.
[0002]         calloc( 1 , 42079 ) = 0x55b3eecd2d0
[0003]         malloc( 42777 ) = 0x55b3eece5740
[0004]         calloc( 1 , 35481 ) = 0x55b3eecefe70
[0005]         malloc( 53397 ) = 0x55b3eecf8920
[0006]         malloc( 39161 ) = 0x55b3eed059c0
[0007]         calloc( 1 , 35953 ) = 0x55b3eed0f2d0
[0008]         malloc( 55464 ) = 0x55b3eed17f50
[0009]         malloc( 39157 ) = 0x55b3eed25800
[0010]         malloc( 20432 ) = 0x55b3eed2f100
[0011]         malloc( 3355 ) = 0x55b3eed340e0
[0012]         free( 0x55b3eed340e0 )
[0013]         free( 0x55b3eed2f100 )
[0014]         free( 0x55b3eed25800 )
[0015]         free( 0x55b3eed17f50 )
[0016]         free( 0x55b3eed0f2d0 )
[0017]         free( 0x55b3eed059c0 )
[0018]         free( 0x55b3eecf8920 )
[0019]         free( 0x55b3eecefe70 )
[0020]         free( 0x55b3eece5740 )
[0021]         free( 0x55b3eecd2d0 )
[0022]
[0023] Statistics
[0024]   allocated_total      367256
[0025]   allocated_avg        36725
[0026]   freed_total          0
[0027]
```

1.2 Part2

-test1

```
[0001] Memory tracer started.
[0002]         malloc( 1024 ) = 0x55b987bed2d0
[0003]         malloc( 32 ) = 0x55b987bed710
[0004]         malloc( 1 ) = 0x55b987bed770
[0005]         free( 0x55b987bed770 )
[0006]         free( 0x55b987bed710 )
[0007]
[0008] Statistics
[0009]   allocated_total      1057
[0010]   allocated_avg        352
[0011]   freed_total          33
[0012]
[0013] Non-deallocated memory blocks
[0014]   block                size      ref cnt
[0015]   0x55b987bed2d0       1024       1
[0016]
[0017] Memory tracer stopped.
```

-test2

```
[0001] Memory tracer started.
[0002]         malloc( 1024 ) = 0x55d7f04492d0
[0003]         free( 0x55d7f04492d0 )
[0004]
[0005] Statistics
[0006]   allocated_total      1024
[0007]   allocated_avg        1024
[0008]   freed_total          1024
[0009]
[0010] Memory tracer stopped.
```

-test3

```
[0001] Memory tracer started.
[0002]         calloc( 1 , 53633 ) = 0x5609bf11a2d0
[0003]         calloc( 1 , 10637 ) = 0x5609bf127490
[0004]         calloc( 1 , 28078 ) = 0x5609bf129e60
[0005]         calloc( 1 , 5065 ) = 0x5609bf130c50
[0006]         calloc( 1 , 37980 ) = 0x5609bf132060
[0007]         calloc( 1 , 17308 ) = 0x5609bf13b500
[0008]         malloc( 1934 ) = 0x5609bf13f8e0
[0009]         malloc( 19932 ) = 0x5609bf1400b0
[0010]         malloc( 24905 ) = 0x5609bf144ed0
[0011]         malloc( 27818 ) = 0x5609bf14b060
[0012]         free( 0x5609bf14b060 )
[0013]         free( 0x5609bf144ed0 )
[0014]         free( 0x5609bf1400b0 )
[0015]         free( 0x5609bf13f8e0 )
[0016]         free( 0x5609bf13b500 )
[0017]         free( 0x5609bf132060 )
[0018]         free( 0x5609bf130c50 )
[0019]         free( 0x5609bf129e60 )
[0020]         free( 0x5609bf127490 )
[0021]         free( 0x5609bf11a2d0 )
[0022]
[0023] Statistics
[0024]   allocated_total      227290
[0025]   allocated_avg        22729
[0026]   freed_total          227290
[0027]
[0028] Memory tracer stopped.
```

1.3 Part3

-test4

```
[0001] Memory tracer started.
[0002]         malloc( 1024 ) = 0x5605201192d0
[0003]         free( 0x5605201192d0 )
[0004]         free( 0x5605201192d0 )
[0005]         *** DOUBLE_FREE *** (ignoring)
[0006]         free( 0x1706e90 )
[0007]         *** ILLEGAL_FREE *** (ignoring)
[0008]
[0009] Statistics
[0010]   allocated_total      1024
[0011]   allocated_avg        1024
[0012]   freed_total          1024
[0013]
[0014] Memory tracer stopped.
```

-test5

```
[0001] Memory tracer started.
[0002]         malloc( 10 ) = 0x5593b11252d0
[0003]         realloc( 0x5593b11252d0 , 100 ) = 0x5593b1125320
[0004]         realloc( 0x5593b1125320 , 1000 ) = 0x5593b11253c0
[0005]         realloc( 0x5593b11253c0 , 10000 ) = 0x5593b11257e0
[0006]         realloc( 0x5593b11257e0 , 100000 ) = 0x5593b1127f30
[0007]         free( 0x5593b1127f30 )
[0008]
[0009] Statistics
[0010]   allocated_total      111110
[0011]   allocated_avg        22222
[0012]   freed_total          111110
[0013]
[0014] Memory tracer stopped.
```

2. 어떻게 구현했는지

2.1 Part1

- fini 함수 안 LOG_STATISTICS()안의 매개변수를 넣었다.
- malloc 함수: dlsym을 이용해 malloc 함수를 call했다. LOG_MALLOC함수를 활용해 내용을 프린트하고, n_allocb에 allocated size를 더하고, n_malloc에는 1을 더해주었다.
- calloc 함수: dlsym을 이용해 calloc 함수를 call했다. LOG_CALLOC함수를 활용해 내용을 프린트하고, n_allocb에 allocated size인 mem*size를 더하고, n_calloc에는 1을 더해주었다.
- realloc 함수: dlsym을 이용해 realloc 함수를 call했다. LOG_REALLOC함수를 활용해 내용을 프린트하고, n_allocb에 allocated size를 더하고, n_calloc에는 1을 더해주었다.
- free 함수 : dlsym을 이용해 free 함수를 call했다. LOG_FREE 함수를 활용해 내용을 프린트했다.

2.2 Part2

- fini 함수 안 LOG_STATISTICS()안의 매개변수를 넣었다. 또, list 를 iterate 하면서 freed되지 않은 block이 있는지 item 안의 cnt를 이용해 확인하며, 있다면 이를 출력했다.
- malloc 함수: Part1의 구현에 alloc함수를 추가해, list에 새로운 allocation에 대한 내용을 넣었다.
- calloc 함수: Part1의 구현에 alloc함수를 추가해, list에 새로운 allocation에 대한 내용을 넣었다.
- realloc 함수: Part1의 구현과 더불어, 만약 realloc의 매개변수로 받은 ptr에 해당하는 메모리가 allocated되어 있고, 해당 사이즈가 0이 아닐 때, 해당 메모리를 dealloc함수를 이용해 리스트에서 cnt=0으로 만들어주고, 새로운 사이즈로 alloc함수를 이용해 allocate했다. 이 때, deallocate시킨 사이즈는 n_freeb에 더해준다. 만약, realloc의 매개변수로 받은 ptr에 해당하는 메모리가 allocated되어 있지 않다면, alloc 함수 이용해 메모리를 allocate해준다.
- free 함수 : Part1의 구현과 더불어, free의 매개변수로 받은 ptr에 해당하는 메모리가 allocated되어 있으면, 이를 dealloc을 이용해 리스트에서 cnt=0으로 만들어주고, n_freeb에 해당 사이즈를 더하고, 실제로 메모리를 free해주었다.

2.3 Part3

- free 함수 : Part2의 구현과 더불어, 이미 free되어 있는 상태를 뜻하는 List에서 해당 포인터는 존재하지만 cnt=0인 상태일 때는 LOG_DOUBLE_FREE 함수를 불러주었고, list에 해당 ptr이 존재하지 않는 상태일 때는 LOG_ILL_FREE() 함수를 불러주었다. 그리고 실제로 메모리를 free해주는 함수를 앞의 상황이 아닌 상황에서만 call되도록 if문 안으로 넣어 주었다.

3. 어려웠던 점

- Part2,3는 Part1에서 약간의 변화만 주면 되기 때문에 그렇게 어렵지 않았는데, Part1에서 dlsym을 이용해 실제 함수를 불러오는 과정이 잘 이해가 되지 않아서, 약간 헤매었다. 포인터의 개념이 확실하게 잡혀있지 않았어서, 함수 포인터 등에 대해 다시 공부하고 전체적인 흐름을 이해하기까지 시간이 꽤 걸렸다. 이후 linked list 등을 다루는 Part2, Part3은 어렵지 않게 해결할 수 있었다.

4. 새롭게 배운 점

- 실제로 library interposition을 해보는 과정을 배웠다. pdf와 책을 읽는 것만으로는 library interposition의 정확한 개념을 잡기 어려웠다. 특히, dlopen, dlsym 등의 함수는 생소한 개념이어서 실제로 사용해보기 전까지는 의미가 모호했는데, 이번 과제를 통해 이를 확실히 이해하게 되었다. 또, 포인터에 대한 내용은 언제 배워도 새롭다. 포인터에 대한 개념이 나올 때마다 거듭 반복학습하다 보니 이제는 조금 이해가 더 잘 된 것 같다. 또, c언어에서 구조체도 실제로 사용해 본 적 없었는데, 이번 기회에 더 정확하게 학습하는 계기가 되었다. 또, malloc과 calloc, realloc의 차이에 대해 확실하게 알지 못했는데, 이의 차이점에 대해서도 새롭게 배웠다.