

2019-2학기

# 컴퓨터 SW시스템 개론

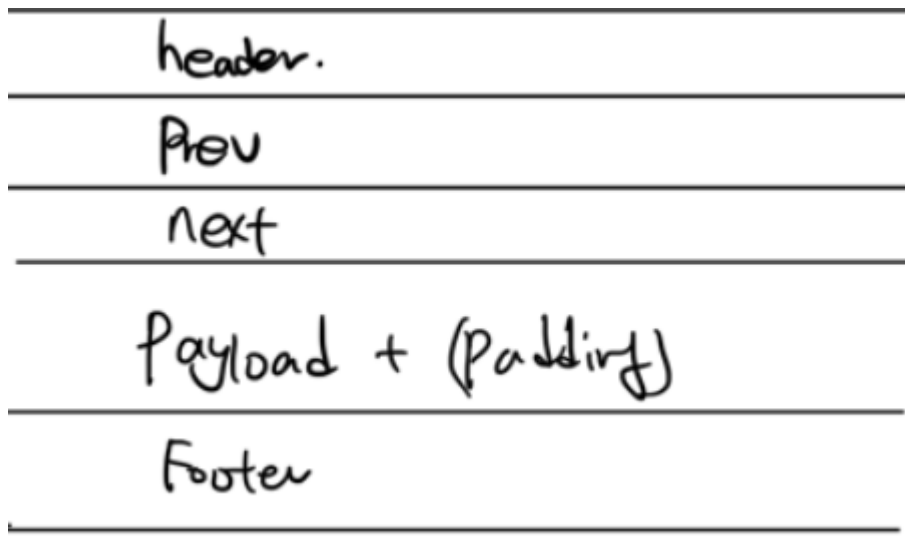
Lab #7

## Lab #7 Efficient Dynamic Memory Allocator

수업시간에 malloc에 대해 배우면서, 다양한 malloc 기법이 있는 것을 배웠다. 크게 Implicit list, Explicit list, Segregated Free list가 있다. 이번 과제에서는 효율적인 동적 할당 코드를 작성하는 것이다. 본 과제에서 사용한 기법은 Explicit Free List이다. 이 기법을 사용한 이유는 채점 기준에 Performance를 평가하는 항목이 있는데, 이 항목에서 높은 점수를 받기 위해서는 Big - O 에 의한 Time complexity가 뛰어나야 한다고 생각했고, PDF 19번을 읽은 후, log의 output을 가진다고 나와있어 원래는 Segregated Free List를 선택하기로 하였다. 하지만, Global Data structure variable을 사용하지 않고는 구현의 어려움을 느껴 Explicit Free List & First Fit을 택하게 되었다.

(편의를 위한 Macro와 Constant를 교과서 Figure 9.43에서 가져왔고, helper function에 대한 concept 또한 책에서 가져왔습니다.)

먼저 본 과제에서 implement한 Block structure이다.



Mm\_init(), mm\_malloc, mm\_free()는 코드 상에 설명이 되어있으므로, 나머지 Helper Function에 대해 설명하도록 하겠다.

### 1. extend\_heap

이 함수는 함수명 대로, heap size를 확장하는 것이다. Free List에서 알맞은 크기에 공간이 없으면 heap을 늘려주고 거기에 allocate하게 된다. 확장하는 함수는 mem\_sbrk를 사용하였다. 그리고 나서 init처럼 setting을 하여준다.

### 2. insert\_node

이 함수는 free나 extend\_heap에 사용된다. 말 그대로 Free List에 넣어주는 함수이다. 그 과정을 실행 할 때, 전 후가 있는지 없는지를 케이스로 나누어 처리한다.

### 3.Place

Malloc을 한뒤 Split을 처리해주는 함수이다.

### 4. delete

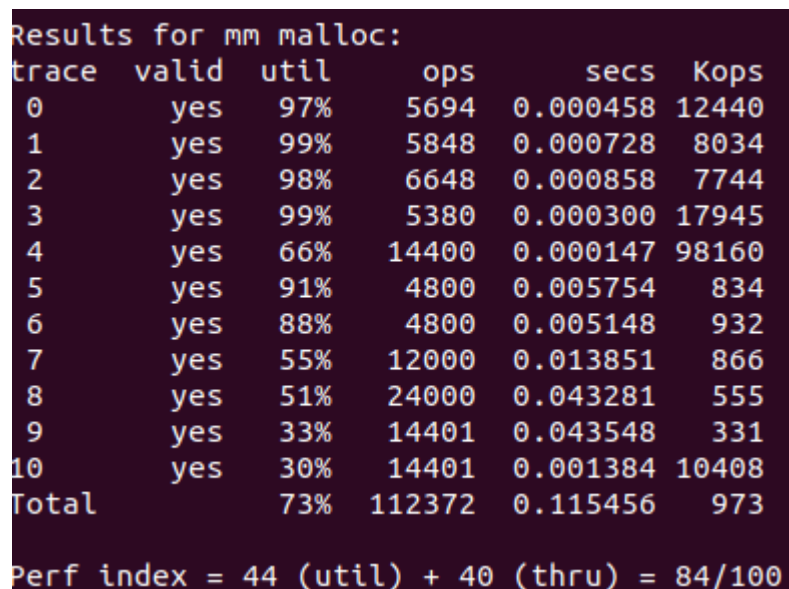
이 함수는 앞에서 연결 지어준 것을 해제할 때 쓰이는 경우이다. 예를 들어 후술할 coalesce를 할 때, 전에 지어준 연결을 해제하고 새로 지어야 하는 과정이 필요한데, 이경우 사용하는 method이다.

### 5. coalesce

PDF에 나와있는 것처럼, Free를 하고 나서 앞 뒤 메모리를 확인하여 같은 Free 상태이면 합치는 method이다. PDF에 나와있는 Case 별로 구분지어 처리한다.

### 6. SearchForFit

First Fit에 의거하여 Fit 되는 Free List를 찾는 함수이다.



```
Results for mm malloc:
trace  valid  util      ops      secs  Kops
0      yes   97%    5694  0.000458 12440
1      yes   99%    5848  0.000728  8034
2      yes   98%    6648  0.000858  7744
3      yes   99%    5380  0.000300 17945
4      yes   66%   14400  0.000147 98160
5      yes   91%    4800  0.005754   834
6      yes   88%    4800  0.005148   932
7      yes   55%   12000  0.013851   866
8      yes   51%   24000  0.043281   555
9      yes   33%   14401  0.043548   331
10     yes   30%   14401  0.001384 10408
Total              73%  112372  0.115456   973

Perf index = 44 (util) + 40 (thru) = 84/100
```

trace	valid	util	ops	secs	Kops
0	yes	97%	5694	0.000458	12440
1	yes	99%	5848	0.000728	8034
2	yes	98%	6648	0.000858	7744
3	yes	99%	5380	0.000300	17945
4	yes	66%	14400	0.000147	98160
5	yes	91%	4800	0.005754	834
6	yes	88%	4800	0.005148	932
7	yes	55%	12000	0.013851	866
8	yes	51%	24000	0.043281	555
9	yes	33%	14401	0.043548	331
10	yes	30%	14401	0.001384	10408
Total		73%	112372	0.115456	973

Perf index = 44 (util) + 40 (thru) = 84/100

Explicit Free List로 구현한 결과물이다. Segregated Free List로 구현하였다면 더 좋은 결과가 나왔을 것이라고 기대한다.