**System Programming Project 4**

교수 : 박성용

이름 : 이예준

학번 : 20212022

**Introduction**

이 document는 코드에서 구현된 memory allocator에 대한 포괄적인 개요를 제공하며,

allocator의 설계, subroutines, 구조체 및 global variable에 대한 설명이 포함되어 있다.

**Design of the Allocator**

allocator는 동적 메모리를 관리하기 위해 segregated free list 방식을 사용한다.

이 방식은 여러 크기의 class 로 나누어진 free blocks list를 유지한다.

allocator는 이러한 segregated free list에서 적절한 block을 선택하여

dynamic memory의 할당, 해제, 재할당을 효율적으로 수행한다.

**Key Components**

1. Segregated Free Lists: 특정 크기 class에 대한 free list를 가리키는 포인터 배열이다.

2. Block Header and Footer: block 크기 및 할당 상태를 관리하기 위해

각 block에 저장된 Metadata 이다.

3. Pointer Manipulation Macros: Block Header, Footer 및 free list에서 다음과 이전 block에 대한

pointer에 접근하고 조작하기 위한 Macro이다.

**Flowchart**

할당 및 해제를 위한 process flow는 다음과 같다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

**Subroutines**

**- mm\_init()**

memory allocator를 초기화한다. 초기 empty heap을 설정하고, segregated free list를 초기화하며,

초기 free block으로 heap을 확장한다.

**- find\_fit()**

요청된 size에 맞는 free block을 적절한 segregated free list에서 검색한다.

**- mm\_malloc(size\_t size)**

지정된 size의 메모리 block을 할당한다. 먼저 segregated free lis에서 맞는 block을 검색하고

배치한다. 적절한 block을 찾을 수 없는 경우, heap을 확장한다.

**- mm\_free(void\* ptr)**

“ptr”가 가리키는 메모리 block을 해제한다. 블록의 header와 footer를 업데이트하여

block이 해제되었음을 표시하고, 적절한 segregated free list에 다시 추가하며,

인접한 free block과 병합한다.

**- mm\_realloc(void\* ptr, size\_t size)**

메모리 block을 새로운 size로 재할당한다. block을 제자리에서 재조정하거나

새 위치로 이동해야 하는 경우를 처리한다.

**- extend\_heap(size\_t words)**

지정된 수의 단어로 heap을 확장하고, 새로운 free block을 초기화하며, 인접한 free block과

병합한다.

**- coalesce(void\* bp)**

“bp”가 가리키는 block과 인접한 free block을 병합하여 단일 더 큰 free block을 만든다.

segregated free list를 적절히 업데이트한다.

**- add\_seglist(char\* bp, size\_t size)**

free block을 크기의 class에 따라 적절한 segregated free list에 추가한다.

**- remove\_seglist(char\* bp)**

free block을 속한 segregated free list에서 제거한다.

**Structures**

Block 구조

각 block은 block 크기와 할당 상태를 저장하는 header 와 footer를 가진다. free block은

free list에서 다음과 이전 블록에 대한 포인터도 포함한다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

**Global Variables**

allocator는 다음 global variables를 사용한다.



segregated free list의 head를 가리키는 포인터 배열이다. 각 list는 특정 크기 class에 대응한다.

**Conclusion**

memory allocator는 segregated free list방식을 사용하여 동적 메모리 할당을 효율적으로 관리한다.

segregated free list를 사용하면 free block 검색이 빠르며, 인접한 free block을 병합하면

단편화를 최소화할 수 있다. 제공된 subroutine과 structure는 이 기능을 구현하여 allocator가

시간과 공간 모두에서 효율적으로 작동하도록 한다.