Operating Systems
Project 0-2. Data Structure

담당교수: 박성용

과목코드: CSE4070-01

제출자: 신현수 (20111636)

1. Function Description Format

Return type	***Function name***	*Parameters*
Function Descriptions		

2. List Library Functions

void	***list_init***	*struct list *list*
Double Linked List 를 초기화 한다. Head / Tail 연결.		

*struct list_elem **	***list_begin***	*struct list *list*
llist 의 첫 노드를 반환한다.		

*struct list_elem **	***list_next***	*struct list_elem *elem*
현재 노드의 다음 노드를 반환한다.		

*struct list_elem **	***list_end***	*struct list *list*
list 의 마지막 노드를 반환한다.		

void	***list_insert***	*struct list_elem *before, struct list_elem *elem*
elem 을 before 노드의 앞에 삽입한다.		

void	***list_splice***	*struct list_elem *before, list_elem *first, struct list_elem *last*
first 노드부터 last 노드까지를 현재 list 에서 제거한 후 before 노드의 앞에 삽입한다.		

void	***list_push_front***	*struct list *list, struct list_elem *elem*
elem 을 list 의 처음에 삽입한다.		

void	***list_push_back***	*struct list *list, struct list_elem *elem*
elem 을 list 의 끝에 삽입한다.		

*struct list_elem **	***list_remove***	*struct list_elem *elem*
elem 을 리스트에서 삭제하고 elem 의 다음 노드를 반환한다.		

*struct list_elem **	***list_pop_front***	*struct list *list*
list 의 처음 노드를 제거하고 그 노드를 반환한다.		

*struct list_elem **	***list_pop_back***	*struct list *list*
list 의 마지막 노드를 제거하고 그 노드를 반환한다.		

*struct list_elem ** **list_front** *struct list *list*

list 의 처음 노드를 반환한다.

*struct list_elem ** **list_back** *struct list *list*

list 의 마지막 노드를 반환한다.

size_t **list_size** *struct list *list*

list 의 전체 노드들의 개수를 반환한다.

bool **list_empty** *struct list *list*

list 의 head 와 tail 이 같으면, 즉 list 가 비어있으면 true 를 반환하고, 그렇지 않으면 false 를 반환한다.

void **list_reverse** *struct list *list*

list 의 순서를 뒤집는다. 즉, list 의 노드들의 위치를 거꾸로 바꾼다.

void **list_sort** *struct list *list, list_less_func *less, void *aux*

less 함수를 이용하여 list 의 노드들을 정렬한다.

void **list_insert_ordered** *struct list *list, struct list_elem *elem, list_less_func *less, void *aux*

less 함수에 대해서 정렬된 list 에 elem 을 정렬되는 위치에 삽입한다.

void **list_unique** *struct list *list, struct list *duplicates, list_less_func *less, void *aux*

list 의 노드들 중 서로 값이 같은 데이터가 있으면, 한 개를 제외하고 중복되는 나머지 노드들을 list 에서 삭제하고 duplicates 가 NULL 이 아닐 때 duplicates 에 list_push_back 함수를 이용하여 넘겨준다.

*struct list_elem ** **list_max** *struct list *list, list_less_func *less, void *aux*

less 함수에 대해서 list 의 노드들 중 값이 가장 큰 노드를 반환한다.

*struct list_elem ** **list_min** *struct list *list, list_less_func *less, void *aux*

less 함수에 대해서 list 의 노드들 중 값이 가장 큰 노드를 반환한다

void **list_swap** *struct list_elem *a, struct list_elem *b*

노드 a 와 노드 b 를 swap 한다.

void **list_shuffle** *struct list *list*

list 의 노드들을 무작위로 섞는다.

3. Hash Library Functions

bool ***hash_init*** *struct hash *h, hash_hash_func *hash, hash_less_func *less, void *aux*

hash 함수, less 함수, aux 데이터를 이용하는 새로운 해시테이블 h 를 초기화 한다.

*struct hash_elem ** ***hash_insert*** *struct hash *h, struct hash_elem *new*

new 와 같은 값을 지닌 원소가 해시테이블 h 에 존재하면 그 원소를 반환하고 new 를 삽입하지 않는다. 만약 존재하지 않으면 new 를 삽입한 후 NULL 을 반환한다.s

*struct hash_elem ** ***hash_replace*** *struct hash *h, struct hash_elem *new*

new 와 같은 값을 지닌 원소가 해시테이블 h 에 존재하면 그 원소를 제거하고 new 를 삽입하고 제거된 원소를 반환한다. 존재하지 않으면 new 를 삽입하고 NULL 을 반환한다.

*struct hash_elem ** ***hash_find*** *struct hash *h, struct hash_elem *e*

e 와 같은 값을 지닌 원소를 해시테이블 h 에서 찾아서 반환한다. 만약 같은 원소가 없을 경우 NULL 을 반환한다.

*struct hash_elem ** ***hash_delete*** *struct hash *h, struct hash_elem *e*

e 와 같은 값을 지닌 원소를 해시테이블 h 에서 찾아서 제거하고 제거된 원소를 반환한다. 만약 같은 원소가 없을 경우 NULL 을 반환한다.

void ***hash_clear*** *struct hash *h, hash_action_func *destructor*

destructor 가 NULL 이 아니면, 해시테이블 h 의 모든 원소들을 destructor 을 이용하여 제거한다.

size_t ***hash_size*** *struct hash *h*

해시테이블 h 내의 원소의 개수를 반환한다.

bool ***hash_empty*** *struct hash *h*

해시테이블 h 이 비어있다면, 즉 원소가 존재하지 않다면, true 를 반환하고, 그렇지 않으면 false 를 반환한다.

void ***hash_apply*** *struct hash *h, hash_action_func *action*

해시테이블 h 내의 모든 원소들에게 action 을 적용시킨다.

void ***hash_first*** *struct hash_iterator *i, struct hash *h*

해시테이블 h 이터레이션을 위해 버킷의 첫번째 노드를 이터레이터로 셋팅한다.

*struct hash_elem ** **hash_next** *struct hash_iterator *i*

해시테이블 이터레이터의 다음 노드를 반환한다.

*struct hash_elem ** **hash_cur** *struct hash_iterator *i*

해시테이블 이터레이터의 현재 노드를 반환한다.

unsigned **hash_int_2** *int i*

해시테이블의 버켓 사이즈인 4 로 modular 연산을 취한 값을 Hash value 로 반환한다.

4. Bitmap Library Functions

*struct bitmap ** **bitmap_create** *size_t bit_cnt*

bit_cnt 사이즈의 비트맵을 생성하여 반환한다.

size_t **bitmap_size** *const struct bitmap *b*

비트맵 b 의 비트 개수를 반환한다.

void **bitmap_destroy** *struct bitmap *b*

비트맵 b 에 할당 된 메모리를 해제한다.

void **bitmap_set** *struct bitmap *b, size_t idx, bool value*

value 가 true 면 bitmap_mark 를 이용하여 비트맵 b 의 idx 번째 비트를 true 로 set 하고, false 면 bitmap_reset 을 이용하여 idx 번째 비트를 false 로 set 한다.

void **bitmap_mark** *struct bitmap *b, size_t bit_idx*

비트맵 b 의 bit_idx 번째 비트를 true 로 set 한다.

void **bitmap_reset** *struct bitmap *b, size_t bit_idx*

비트맵 b 의 bit_idx 번째 비트를 false 로 set 한다.

void **bitmap_flip** *struct bitmap *b, size_t bit_idx*

비트맵 b 의 bit_idx 번째 비트가 true 면 false 를 반환하고, false 면 true 를 반환한다.

void **bitmap_test** *const struct bitmap *b, size_t idx*

비트맵 b 의 idx 번째 비트의 값을 출력한다.

void **bitmap_set_all** *struct bitmap *b, bool value*

비트맵 b 의 모든 비트들을 value 로 set 한다.

void **bitmap_set_multiple** *struct bitmap *b, size_t start, size_t cnt, bool value*

비트맵 b 의 start 번째 비트부터 cnt 개수만큼의 비트들을 value 로 set 한다.

size_t ***bitmap_count*** *const struct bitmap *b, size_t start, size_t cnt, bool value*

비트맵 b 의 start 번째 비트부터 cnt 개수만큼의 비트들 중 value 로 set 되어있는 비트들의 개수를 반환한다.

bool ***bitmap_contains*** *const struct bitmap *b, size_t start, size_t cnt, bool value*

비트맵 b 의 start 번째 비트부터 cnt 개수만큼의 비트들 중 value 로 set 되어있는 비트가 존재하면 true 를 반환하고, 그렇지 않으면 false 를 반환한다.

bool ***bitmap_any*** *const struct bitmap *b, size_t start, size_t cnt*

비트맵 b 의 start 번째 비트부터 cnt 개수만큼의 비트들 중 true 로 set 되어있는 비트가 존재하면 true 를 반환하고, 그렇지 않으면 false 를 반환한다.

bool ***bitmap_none*** *const struct bitmap *b, size_t start, size_t cnt*

비트맵 b 의 start 번째 비트부터 cnt 개수만큼의 비트들 중 true 로 set 되어있는 비트가 존재하면 false 를 반환하고, 그렇지 않으면 true 를 반환한다.

bool ***bitmap_all*** *const struct bitmap *b, size_t start, size_t cnt*

비트맵 b 의 start 번째 비트부터 cnt 개수만큼의 비트들 중 false 로 set 되어있는 비트가 존재하면 false 를 반환하고, 그렇지 않으면 true 를 반환한다.

size_t ***bitmap_scan*** *const struct bitmap *b, size_t start, size_t cnt, bool value*

비트맵 b 의 start 번째 비트부터 마지막 번째 비트들 중 cnt 개수만큼 value 로 set 되어있는 비트들의 가장 처음 비트를 반환한다.

size_t ***bitmap_scan_and_flip*** *struct bitmap *b, size_t start, size_t cnt, bool value*

비트맵 b 의 start 번째 비트부터 마지막 번째 비트들 중 cnt 개수만큼 value 로 set 되어있는 비트들의 가장 처음 비트부터 cnt 개수만큼의 비트들을 value 의 반대로 set 한다. 즉, bitmap_scan 을 하여 얻은 위치의 비트부터 cnt 개수만큼의 비트들을 value 가 true 면 false 로, false 면 true 로 set 한다.

void ***bitmap_dump*** *const struct bitmap *b*

비트맵 b 내에 들어있는 값을 16 진수로 나타낸다.

*struct bitmap ** ***bitmap_expand*** *struct bitmap *bitmap, int size*

bitmap 비트맵의 크기를 size 만큼 더 늘린 후 그 비트맵을 반환한다. 새로 생긴 비트들은 false 로 초기화한다.