

Unix 프로그래밍 및 실습

2장. 파일 입출력 (1)

강의 내용

- ▶ 시스템콜을 이용한 파일 입출력
 - ▶ 강의 내용 (1 ~ 25/51)
 - ▶ <http://lily.mmu.ac.kr/lecture/13u2/ch02.pdf>

기본 과제 2-1

- ▶ [예제 2-1], [예제 2-2], [예제 2-3] (각 10점씩)
 - ▶ 과제 개요 (2줄 이상)
 - ▶ 프로그램
 - ▶ 실행화면 캡처
- ▶ [예제 2-4], [예제 2-5] (각 10점씩)
- ▶ [예제 2-6], [예제 2-7], [예제 2-8] (각 10점씩)
- ▶ [예제 2-9], [예제 2-10] (각 10점씩)

응용과제 2-1 (1)

▶ 숙박카드 구조체

```
typedef struct {  
    int        no_of_guests;  
    char       name[NAME_SIZE];  
} guest_card;
```

▶ 호텔 객실 번호 부여 방식은 각자가 선택

- ▶ 예) 1호실부터 순서대로 부여
- ▶ 예) 각 층에 방 20개씩, 층 번호 2자리 + 방 번호 2자리로 설정 : 1205호

▶ 정상적인 방 번호인지 여부 확인 함수 (100점)

▶ 방 번호로 파일 내 위치(offset) 구하는 함수 (100점)

▶ 빈 방인지 여부 확인 함수 (100점)

▶ 입실 기능 (100점)

- ▶ 사용자로부터 원하는 객실 번호를 입력 받아 현재 투숙 중인지 확인하여 빈 방이면 손님 이름과 투숙 인원 저장
- ▶ 이미 손님이 있는 방이면 객실 번호를 다시 입력 받아 진행

▶ 전체 조회 기능 (100점)

- ▶ 손님이 투숙 중인 객실 번호와 손님 이름 표 형식으로 출력 (빈 방은 출력하지 말 것!)

응용과제 2-1(2)

- ▶ 손님 이름으로 객실 번호 조회 기능 (100점)
 - ▶ 손님 이름을 입력 받아 투숙 중인 객실 번호 출력 (해당 손님이 없는 경우 안내 메시지 출력)
- ▶ 객실 번호로 손님 이름 조회 기능 (100점)
 - ▶ 객실 번호를 입력 받아 해당 방에 투숙 중인 손님의 이름 출력 (빈 방인 경우 안내 메시지 출력)
- ▶ 퇴실 기능 (100점)
 - ▶ 사용자로부터 객실 번호와 이름을 입력 받아, 해당 객실에 투숙 중인 손님이 맞는 경우 숙박카드 삭제
- ▶ 메인에서는 메뉴를 출력하고, 해당 기능 선택 받아 기능을 수행하고, 완료하면 다시 메뉴 화면 출력 (100점)
- ▶ 결과 화면 캡처 순서
 - ▶ 메뉴 출력 >> 전체조회 >> 입실 (각 기 다른 층에서 전체 5명 이상 손님 투숙) >> 전체조회 >> 손님 이름으로 조회 (성공/실패) >> 객실 번호로 조회(성공/실패) >> 퇴실 >> 전체조회 >> 퇴실한 번호로 조회 >> 퇴실한 손님으로 조회

응용과제 2-1(3)

- ▶ 보고서 첫 장에 다음 같은 표 작성

과제	완성 여부 (O, X)	비고
정상 방번호 확인 함수	O	
방 번호로 파일 위치 구하는 함수	O	
빈방 여부 확인 함수	O	
입실 함수	O	
퇴실 함수	O	
전체 조회 함수	O	
이름 조회 함수	X	
호실 조회 함수	X	
main 함수	O	

- ▶ 제출기한 : 9월 25일 자정

과제 힌트 (1)

▶ 구조체

- ▶ 구조체의 멤버 중 문자열의 크기가 4의 배수가 아닐 때 어떤 일이 일어나는가?
- ▶ 다음 구조체의 실제 크기는 ?

```
typedef struct {  
    char    name[10];  
    int     no_of_guests;  
    char    sex; // 'm' or 'w'  
} guest_card;
```

▶ <string.h> - 문자열 관련 주요 함수

- ▶ **strncmp**
 - ▶ 2개의 문자열 비교 (동일하면 0 반환)
- ▶ **strncpy**
 - ▶ 문자열 복사
- ▶ **memset**
 - ▶ 메모리 영역을 특정 값으로 채움 (보통 0으로 초기화)
 - ▶ 구조체 초기화

과제 힌트 (2)

▶ Makefile

```
CC = gcc
CFLAGS = -g
OBJS = ext3.o show_menu.o guest_in.o check_out.o list.o search_by_name.o search_by_room_no.o

ext3: $(OBJS)
    $(CC) $(CFLAGS) -o ext3 $(OBJS)
```

▶ 임시 파일 예 (search_by_name.c)

```
void search_by_name()
{
    // Stub function -- 아직 구현하지 않은 함수
}
```


과제 힌트 (3)

▶ ext3.c (일부)

```
int end_of_program = 0;
int choice;
char garbage[255];

...

while (!end_of_program) {
    show_menu();
    printf("Choose the menu no : ");
    if (scanf("%d", &choice) == 1) {
        switch (choice) {
            case 1:
                guest_in();
                break;

                ...

            case 6:
                end_of_program = 1;
                break;
            default:
                printf("Invalid choice !\n");
        }
    }
    else {
        printf("Invalid input !\n");
        scanf("%s", garbage);
    }
}

...
```

과제 힌트 (4)

▶ guest_in.c (일부)

```
...  
  
    if (new_card.no_of_guests > 0) {  
        fd = open(DATA_FILE_NAME, O_RDWR);  
  
        if (fd > 0) { // 기존 숙박 파일 존재  
            lseek(fd, (room_no-1)*sizeof(new_card), SEEK_SET);  
            read(fd, &org_card, sizeof(org_card));  
  
            if(org_card.no_of_guests == 0) {  
                lseek(fd, (room_no-1)*sizeof(new_card), SEEK_SET);  
                write(fd, &new_card, sizeof(new_card));  
            }  
            else {  
                printf("빈 방이 아닙니다!!!!\n");  
            }  
            close(fd);  
        }  
        else { // 숙박 파일이 존재하지 않는 경우 - 새로 생성  
            fd = open(DATA_FILE_NAME, O_CREAT | O_RDWR, S_IRWXU);  
  
            if (fd > 0) {  
                lseek(fd, (room_no-1)*sizeof(new_card), SEEK_SET);  
                write(fd, &new_card, sizeof(new_card));  
                close (fd);  
            }  
        }  
    }  
  
}
```

...

과제 힌트 (5)

▶ list.c (버전 1 - 일부)

```
...  
  
int      room_no = 1;  
  
guest_card_t  g_record;  
  
memset(&g_record, 0, sizeof(g_record));  
fd = open(DATA_FILE_NAME, O_RDONLY);  
  
if (fd > 0) {  
    while(read(fd, &g_record, sizeof(g_record)) > 0) {  
        if (g_record.no_of_guests != 0) {  
            printf("%10d호실 \t이름 : %-20s 투숙 인원 : %d \n",  
                room_no, g_record.name, g_record.no_of_guests);  
        }  
        room_no ++;  
        memset(&g_record, 0, sizeof(g_record));  
    }  
    close(fd);  
}  
  
...
```

과제 힌트 (6)

▶ list.c (버전 2 - 일부)

```
...  
  
int room_no = 1;  
off_t last_pos, cur_pos;  
guest_card_t g_record;  
  
memset(&g_record, 0, sizeof(g_record));  
fd = open(DATA_FILE_NAME, O_RDONLY);  
  
if (fd > 0) {  
    last_pos = lseek(fd, 0, SEEK_END); // file 끝 위치 저장  
  
    cur_pos = lseek(fd, 0, SEEK_SET);  
    while(cur_pos < last_pos) {  
        read(fd, &g_record, sizeof(g_record));  
        if (g_record.no_of_guests != 0) {  
            printf("%10d호실 \t이름 : %-20s 투숙 인원 : %d \n",  
                  room_no, g_record.name, g_record.no_of_guests);  
        }  
        room_no ++;  
        memset(&g_record, 0, sizeof(g_record));  
        cur_pos = lseek(fd, 0, SEEK_CUR); // 현재 위치 저장  
    }  
  
    close(fd);  
}  
  
...
```