C프로그래밍 및 실습2(가)

과제(업무)관리 프로그램

문지환(20213069)

< 목 차 >

서론
1. 프로그램 소개
2. 기능 소개

본론
3. 구현 방법
4. 결과 화면

결론
5. 발전 방향

서론

서 론

1. 프로그램 소개

일상생활에서 유용한 관리 프로그램

넘쳐나는 과제나 업무

과제(업무) 관리 프로그램

서 론

2. 기능 소개

1	2	3	4
print_all	print_task	add_task	delete_task
과제의 전체 목록을 출력	원하는 과제 하나를 출력	새로운 과제 하나를 추가	원하는 과제 하나를 삭제

2. 기능 소개

5	6	7	8
delete_all	write_file	search_task	exit
과제의 전체 목록을 삭제	txt파일에 내용을 업데이트	원하는 과제를 검색	프로그램 종료

본론

본 론

1. 구현 방법

```c
struct task{
    char name[SIZE];
    int date;
    struct task* next;
};
typedef struct task* task_ptr;
```

```c
void read_file(task_ptr* head_ptr);
int menu();
int count_tasks(const task_ptr head);
void print_all(task_ptr head);
void print_task(task_ptr head);
void add_task(task_ptr* head_ptr);
void delete_task(task_ptr* head_ptr);
void delete_all(task_ptr* head_ptr);
void write_file(task_ptr head);
void search_task(task_ptr head);
```

```c
int main(){
    task_ptr head = NULL;
    read_file(&head);

    while (1){
        printf("\n");

        int i = menu();

        switch (i){
        case 1:
            print_all(head);
            break;
        case 2:
            print_task(head);
            break;
        case 3:
            add_task(&head);
            break;
        case 4:
            delete_task(&head);
            break;
        case 5:
            delete_all(&head);
            break;
        case 6:
            write_file(head);
            break;
        case 7:
            search_task(head);
            break;
        case 8:
            printf("Good Bye\n");
            delete_all(&head);
            write_file(head);
            exit(0);
        default:
            printf("%d is not included. \n", i);
        }
    }
}
```

```c
void print_all(task_ptr head){
    task_ptr node = head;

    int count = 0;
    while (node != NULL){
        printf("%d : %s %d\n", count, node->name, node->date);
        node = node->next;
        count++;
    }
}
```

본 론

1. 구현 방법

```c
void print_task(task_ptr head){
    printf("Input the index of task to print.₩n");
    int index = input_int();

    task_ptr node = head;

    int count = 0;

    while (node != NULL){
        if (count == index) break;

        node = node->next;
        count++;
    }

    if (node != NULL)
        printf("%d : %s %d₩n", count, node->name, node->date);
    else
        printf("Invalid task.₩n");
}
```

```c
void add_task(task_ptr* head_ptr) {
    printf("Input name.₩n");
    printf(">> ");

    task_ptr new_task = (task_ptr)malloc(sizeof(struct task));
    if (new_task == NULL){
        printf("Malloc failed₩n");
        exit(1);
    }

    new_task->next = NULL;

    int i = scanf("%[^₩n]%*c", &new_task->name);
    printf("Input date.₩n");
    printf(">> ");
    i = scanf("%d%*c", &new_task->date);

    int count = 0;
    task_ptr node = *head_ptr;

    if (node == NULL)
        *head_ptr = new_task;
    else{
        while (node->next != NULL){
            node = node->next;
            count++;
        }
        node->next = new_task;
        count++;
    }

    printf("%d : %s %d₩n", count, new_task->name, new_task->date);
}
```

```c
void delete_task(task_ptr* head_ptr){
    printf("Input the index of task to delete.₩n");
    int index = input_int();

    task_ptr node = *head_ptr;
    task_ptr prev = NULL;

    int count = 0;
    while (node != NULL){
        if (count == index) break;

        prev = node;
        node = node->next;
        count++;
    }

    if (node == NULL){
        printf("₩rong index₩n");
        return;
    }

    if (prev == NULL)
        *head_ptr = node->next;
    else
        prev->next = node->next;
    free(node);
}
```

본 론

1. 구현 방법

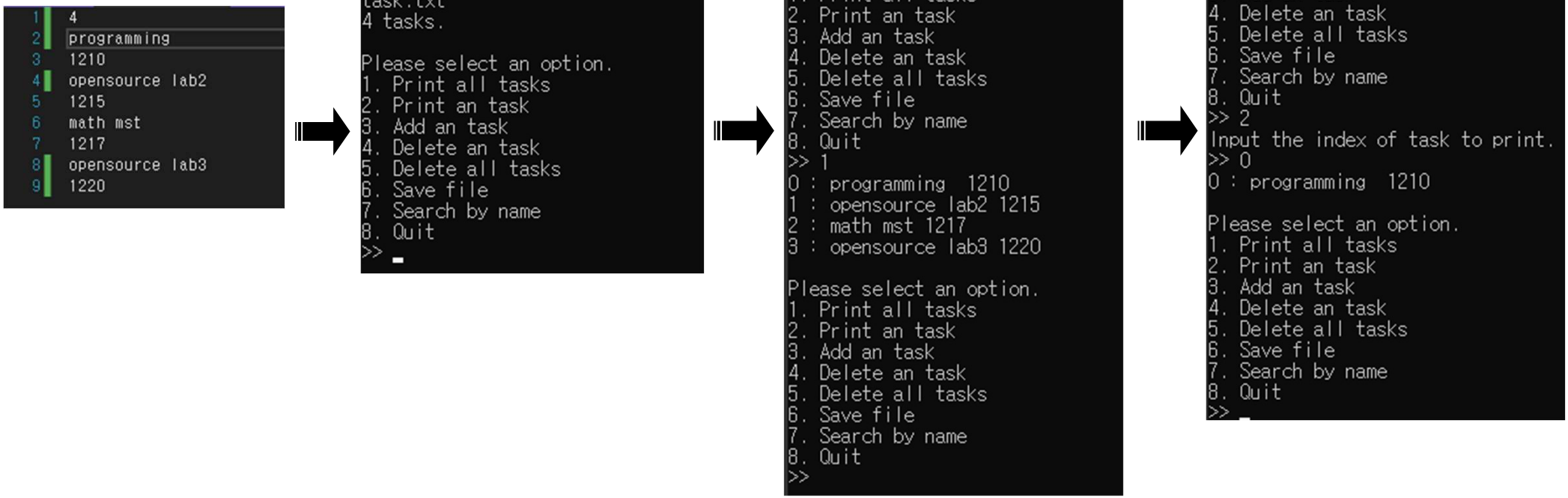
```c
void delete_all(task_ptr* head_ptr){
    if (*head_ptr == NULL){
        printf("Nothing to delete.₩n");
        return;
    }

    task_ptr search = *head_ptr;
    task_ptr tmp = NULL;

    int count = 0;
    while (search != NULL){
        printf("%s is deleted.₩n", search->name);
        tmp = search->next;
        free(search);
        search = tmp;
        count++;
    }

    *head_ptr = NULL;
    printf("%d tasks deleted.₩n", count);
}
```

```c
void write_file(task_ptr head){
    char filename[SIZE] = { 0, };

    printf("Please input filename to read and press Enter.₩n");
    printf(">> ");

    if (scanf("%[^₩n]%*c", filename) != 1){
        printf("Wrong input.₩n");
        exit(1);
    }

    FILE* file = fopen(filename, "w");

    if (file == NULL){
        printf("Error: Cannot open file. ₩n");
        exit(1);
    }

    int count = 0;

    fprintf(file, "%d₩n", (int)count_tasks(head));

    task_ptr node = head;

    while (node != NULL){
        fprintf(file, "%s₩n", node->name);
        fprintf(file, "%d₩n", node->date);

        node = node->next;

        count++;
    }

    fclose(file);

    assert(count == (int)count_tasks(head));

    printf("%d tasks have been saved to the file.₩n", count);
}
```

```c
void search_task(task_ptr head){
    printf("Please input name to search.₩n");
    printf(">> ");

    char name[SIZE] = { 0, };
    if (scanf("%[^₩n]%*c", name) != 1){
        printf("Wrong input.₩n");
        return;
    }

    task_ptr node = head;

    int count = 0;
    while (node != NULL){
        if (strcmp(node->name, name) == 0) break;
        node = node->next;
        count++;
    }

    if (node == NULL){
        printf("No task found : %s₩n", name);
        return;
    }

    printf("%d : %s %d₩n", count, node->name, node->date);
}
```

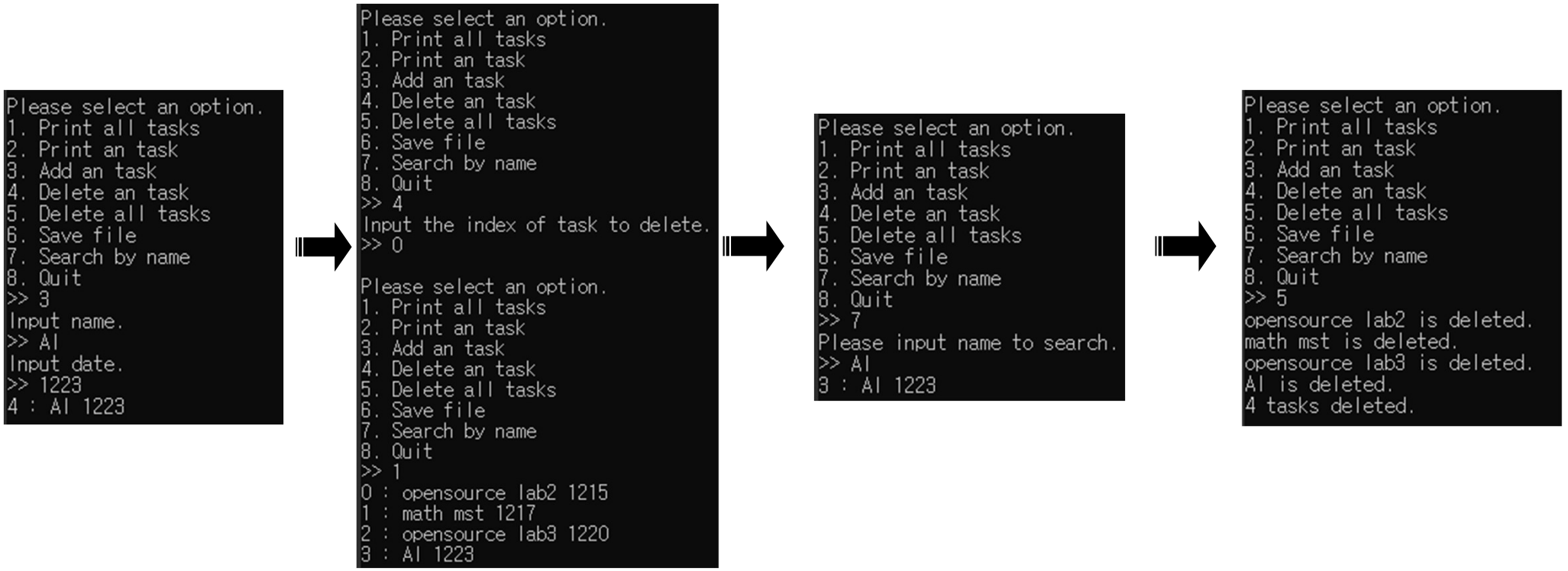
본 론

2. 결과 화면


4
programming
1210
opensource lab2
1215
math mst
1217
opensource lab3
1220
Please input filename to read.
task.txt
4 tasks.
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> _
Please input filename to read.
task.txt
4 tasks.
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 1
0 : programming 1210
1 : opensource lab2 1215
2 : math mst 1217
3 : opensource lab3 1220
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>>
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 2
Input the index of task to print.
>> 0
0 : programming 1210
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> _

본 론

2. 결과 화면


Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 3
Input name.
>> AI
Input date.
>> 1223
4 : AI 1223
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 4
Input the index of task to delete.
>> 0
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 1
0 : opensource lab2 1215
1 : math mst 1217
2 : opensource lab3 1220
3 : AI 1223
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 7
Please input name to search.
>> AI
3 : AI 1223
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 5
opensource lab2 is deleted.
math mst is deleted.
opensource lab3 is deleted.
AI is deleted.
4 tasks deleted.

2. 결과 화면

```
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 6
Please input filename to read and press Enter.
>> task.txt
0 tasks have been saved to the file.
```

```
Please select an option.
1. Print all tasks
2. Print an task
3. Add an task
4. Delete an task
5. Delete all tasks
6. Save file
7. Search by name
8. Quit
>> 8
Good Bye
Please input filename to read and press Enter.
>> task.txt
0 tasks have been saved to the file.
```

결론

결 론

발전 방향

1. 현재 날짜를 입력해서 남은 시간을 계산

2. 날짜만이 아닌 다른 세부정보 입력

3. 과제 세부정보 수정, 우선순위에 따라 중간에 삽입

4. 검색할 때 이름 전체가 아닌 일부 내용으로 검색 가능

5. 날짜나 우선순위로 검색 가능

THANK YOU
FOR LISTENING