

Федеральное агентство связи
Сибирский государственный университет телекоммуникаций и
информатики

Кафедра прикладной математики и кибернетики

Отчёт по лабораторной работе №1

Выполнил: студент группы ИП-811
Адов А.С.
Проверил: Малков Е.А.

Новосибирск 2020

Цель: проверка знаний , необходимых для усвоения курса «Операционные системы».

Задание: написать программу для манипуляции данными на основе рекурсивных структур, реализовать функции вставки, удаления и навигации для списка, реализовать сериализацию списка.

Код программы:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#define ERROR_OPEN_FILE 0b00000001
```

```
#define ERROR_EMPTY_FILE 0b00000010
```

```
#define DONT_FIND_NOTE 0b00000011
```

```
#define DONT_DELETE_NOTE 0b00000100
```

```
#define FILE_NAME "journal.bin"
```

```
#define CLEAR_SCREEN printf("\E[H\E[J");
```

```
#define MESSAGE_INCORRECT_CHOICE \
```

```
printf("You choice incorrect paragraph menu\n");
```

```
#define HEADER_OF_THE_TABLE printf("Surname\tAverage mark\n");
```

```
#define CHECK_OPEN_FILE(file) \
```

```
if (file == NULL) { \
```

```
    return ERROR_OPEN_FILE; \
```

```
}
```

```
#define CHECK_EMPTY_FILE(file) \
```

```
fseek(file, 0, SEEK_END); \
```

```
if (ftell(file) <= 0) { \
```

```
    return ERROR_EMPTY_FILE; \
```

```
} else { \
```

```
    fseek(file, 0, SEEK_SET); \
```

```
}
```

```
#define DRAW_MENU \
```

```
printf("1) Print all journal\n"); \
```

```
printf("2) Fill journal\n"); \
```

```
printf("3) Save journal\n"); \
```

```
printf("4) Load journal\n"); \
```

```
printf("5) Find note\n"); \
```

```
printf("6) Delete note\n"); \
```

```
printf("9) Clear screen\n"); \
```

```
printf("0) Exit\n");
```

```
typedef struct Pouple {
```

```
int average_mark;
```

```
int length_surname;
```

```
char* surname;
```

```
} pouple;
```

```
typedef struct List {
```

```
    pouple data;
```

```
    struct List* next;
```

```
} list;
```

```
void Print_All_Journal(list* journal)
```

```
{
```

```
    HEADER_OF_THE_TABLE;
```

```
    while (journal != NULL) {
```

```
        for (short int i = 0; i < journal->data.length_surname; i++) {
```

```
            printf("%c", journal->data.surname[i]);
```

```
        }
```

```
        printf("\t\t%d\n", journal->data.average_mark);
```

```
        journal = journal->next;
```

```
    }
```

```
}
```

```
int Add_Note_In_Journal(list** journal)

{

    list* temp_head = NULL;

    list* buffer = NULL;

    temp_head = (*journal);

    if (*journal != NULL) {

        temp_head = (*journal);

        while (temp_head->next != NULL) {

            temp_head = temp_head->next;

        }

    }

    buffer = (list*)malloc(1 * sizeof(list));

    buffer->next = NULL;

    printf("Write need length surname\n");

    scanf("%d", &(buffer->data.length_surname));
```

```
printf("Write need surname\n");
```

```
scanf(" ");
```

```
buffer->data.surname
```

```
    = (char*)calloc(buffer->data.length_surname, sizeof(char)); //
```

```
fgets(buffer->data.surname, (buffer->data.length_surname + 1), stdin);
```

```
printf("Write need average mark\n");
```

```
scanf("\n%d", &buffer->data.average_mark);
```

```
if (*journal == NULL) {
```

```
    *journal = buffer;
```

```
    return 1;
```

```
} else {
```

```
    temp_head->next = buffer;
```

```
    return 2;
```

```
}
```

```
}
```



```
int Save_Journal(list* journal)

{

    FILE* file = NULL;

    file = fopen(FILE_NAME, "wb");

    CHECK_OPEN_FILE(file);

    while (journal != NULL) {

        fwrite(&journal->data.length_surname, sizeof(int), 1, file);

        for (short int i = 0; i < journal->data.length_surname; i++) {

            fwrite(&journal->data.surname[i], sizeof(char), 1, file);

        }

        fwrite(&journal->data.average_mark, sizeof(int), 1, file);

        if (journal->next != NULL) {

            putc('\n', file);

        }

        journal = journal->next;

    }

}
```

```
fclose(file);

return 0;

}

int Load_Journal(list** journal)

{

    int i;

    FILE* file = NULL;

    list* temp_note = NULL;

    list* buffer = NULL;

    free(*journal);

    *journal = NULL;

    file = fopen(FILE_NAME, "rb");

    CHECK_OPEN_FILE(file);

    CHECK_EMPTY_FILE(file);

    while (!feof(file)) {
```

```
if ((*journal) == NULL) {

    (*journal) = (list*)malloc(1 * sizeof(list));

    (*journal)->next = NULL;

    fread(&((*journal)->data.length_surname), sizeof(int), 1, file);

    (*journal)->data.surname = (char*)calloc(

        (*journal)->data.length_surname, sizeof(char));

    for (i = 0; i < (*journal)->data.length_surname; i++) {

        fread(&((*journal)->data.surname[i]), sizeof(char), 1, file);

    }

    fread(&((*journal)->data.average_mark), sizeof(int), 1, file);

    fgetc(file);

    temp_note = (*journal);

} else {

    buffer = (list*)malloc(1 * sizeof(list));

    buffer->next = NULL;

    fread(&(buffer->data.length_surname), sizeof(int), 1, file);
```

```
buffer->data.surname
```

```
    = (char*)calloc(buffer->data.length_surname, sizeof(char));
```

```
    for (i = 0; i < buffer->data.length_surname; i++) {
```

```
        fread(&(buffer->data.surname[i]), sizeof(char), 1, file);
```

```
    }
```

```
    fread(&(buffer->data.average_mark), sizeof(int), 1, file);
```

```
    fgetc(file);
```

```
    temp_note->next = buffer;
```

```
    temp_note = buffer;
```

```
    }
```

```
}
```

```
fclose(file);
```

```
return 0;
```

```
}
```

```
int Find_Note_In_Journal(list* journal)
```

```
{

list* temp_journal = NULL;

char* find_surname = NULL;

int length_find_surname = 0;

temp_journal = journal;

printf("Write quantity letter in need surname\n");

scanf("%d", &length_find_surname);

printf("Write need surname\n");

find_surname = (char*)calloc(length_find_surname, sizeof(char));

scanf(" ");

fgets(find_surname, length_find_surname + 1, stdin);

while (temp_journal != NULL) {

    if (strcmp(find_surname, temp_journal->data.surname) == 0) {

        printf("%s\t%d\n", find_surname, temp_journal->data.average_mark);

        return 0;

    }

}
```

```
temp_journal = temp_journal->next;
```

```
}
```

```
return DONT_FIND_NOTE;
```

```
}
```

```
void Delete_Element(list* temp_journal, list** journal)
```

```
{
```

```
list* temp = NULL;
```

```
temp = (*journal);
```

```
while (temp->next != temp_journal) {
```

```
temp = temp->next;
```

```
}
```

```
temp->next = temp_journal->next;
```

```
free(temp_journal);
```

```
}
```

```
int Delete_Note_In_Journal(list** journal)

{

    list* temp_journal = NULL;

    int delete_number;

    temp_journal = *journal;

    printf("Write need number for delete");

    scanf("%d", &delete_number);

    int i = 1;

    while (temp_journal != NULL && i != delete_number) {

        i++;

        temp_journal = temp_journal->next;

    }

    if (temp_journal == NULL) {

        return DONT_DELETE_NOTE;

    } else {

        if (i == 1) {
```

```
*journal = temp_journal->next;
```

```
free(temp_journal);
```

```
} else {
```

```
    Delete_Element(temp_journal, journal);
```

```
}
```

```
}
```

```
return DONT_DELETE_NOTE;
```

```
}
```

```
int main()
```

```
{
```

```
list* journal = NULL;
```

```
int error_flag = 0b0;
```

```
char key_choice = '1';
```

```
while (key_choice != '0') {
```

```
    DRAW_MENU;
```



```
scanf(" %c", &key_choice);

switch (key_choice) {

case ('1'): {

    Print_All_Journal(journal);

    break;

}

case ('2'): {

    error_flag = Add_Note_In_Journal(&journal);

    break;

}

case ('3'): {

    error_flag = Save_Journal(journal);

    break;

}

case ('4'): {

    error_flag = Load_Journal(&journal);
```

```
        break;

    }

    case ('5'): {

        error_flag = Find_Note_In_Journal(journal);

        break;

    }

    case ('6'): {

        error_flag = Delete_Note_In_Journal(&journal);

        break;

    }

    case ('9'): {

        CLEAR_SCREEN;

        break;

    }

    case ('0'): {

        return error_flag;
```

```
}
```

```
default: {
```

```
    MESSAGE_INCORRECT_CHOICE;
```

```
    break;
```

```
}
```

```
}
```

```
}
```

```
return error_flag;
```

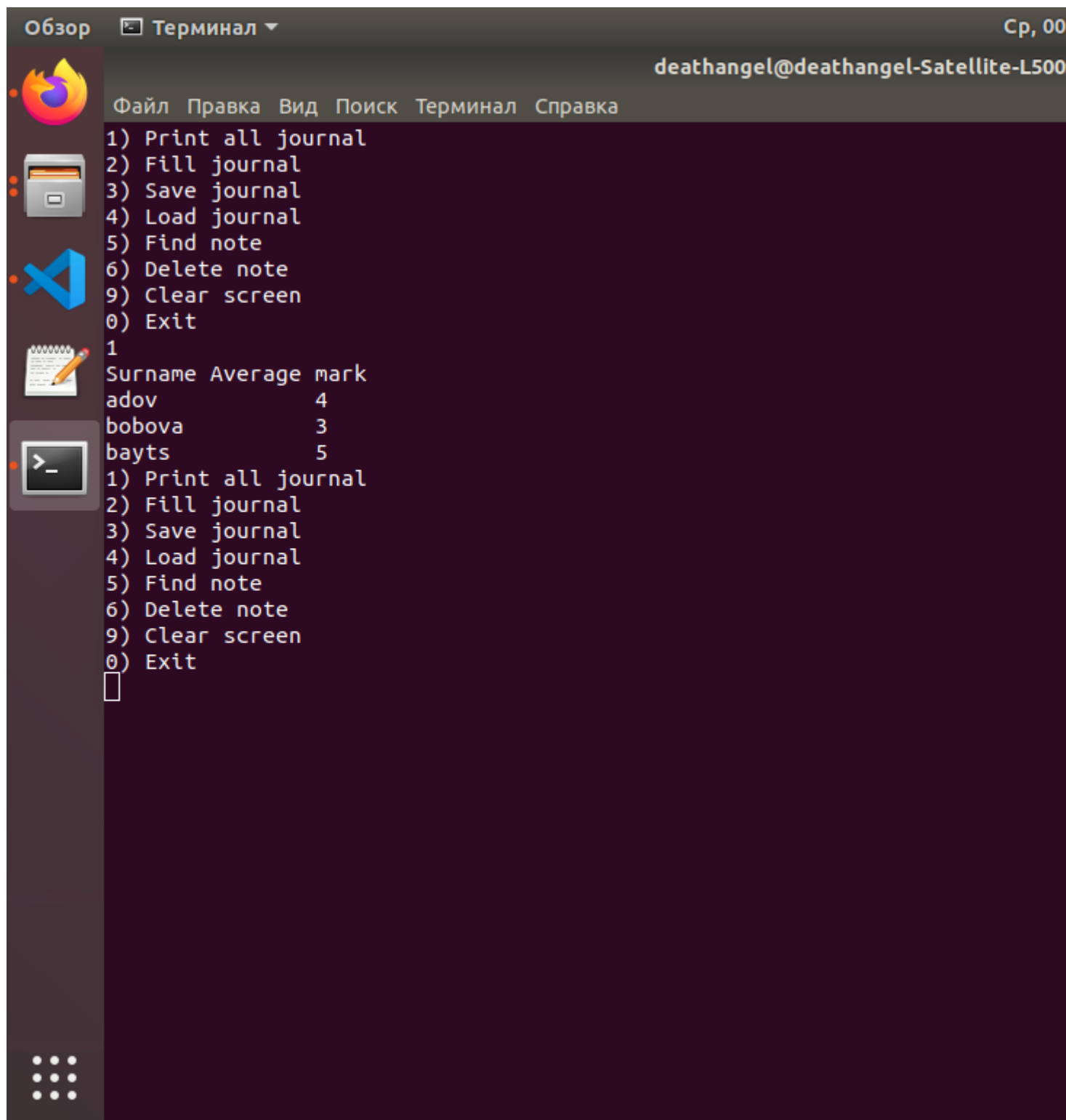
```
}
```

Обзор Терминал Cp, 00 deathangel@deathangel-Satellite-L500

Файл Правка Вид Поиск Терминал Справка

2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
2
Write need length surname
6
Write need surname
bobova
Write need average mark
3
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
2
Write need length surname
5
Write need surname
bayts
Write need average mark
5
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit

Заполнение списка



Вывод всего списка в терминал

Обзор Терминал

deathangel@deathangel-Satellite-L500

Файл Правка Вид Поиск Терминал Справка

4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit

3

1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit

4

1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit

1

Surname Average mark
adov 4
bobova 3
bayts 5

1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit

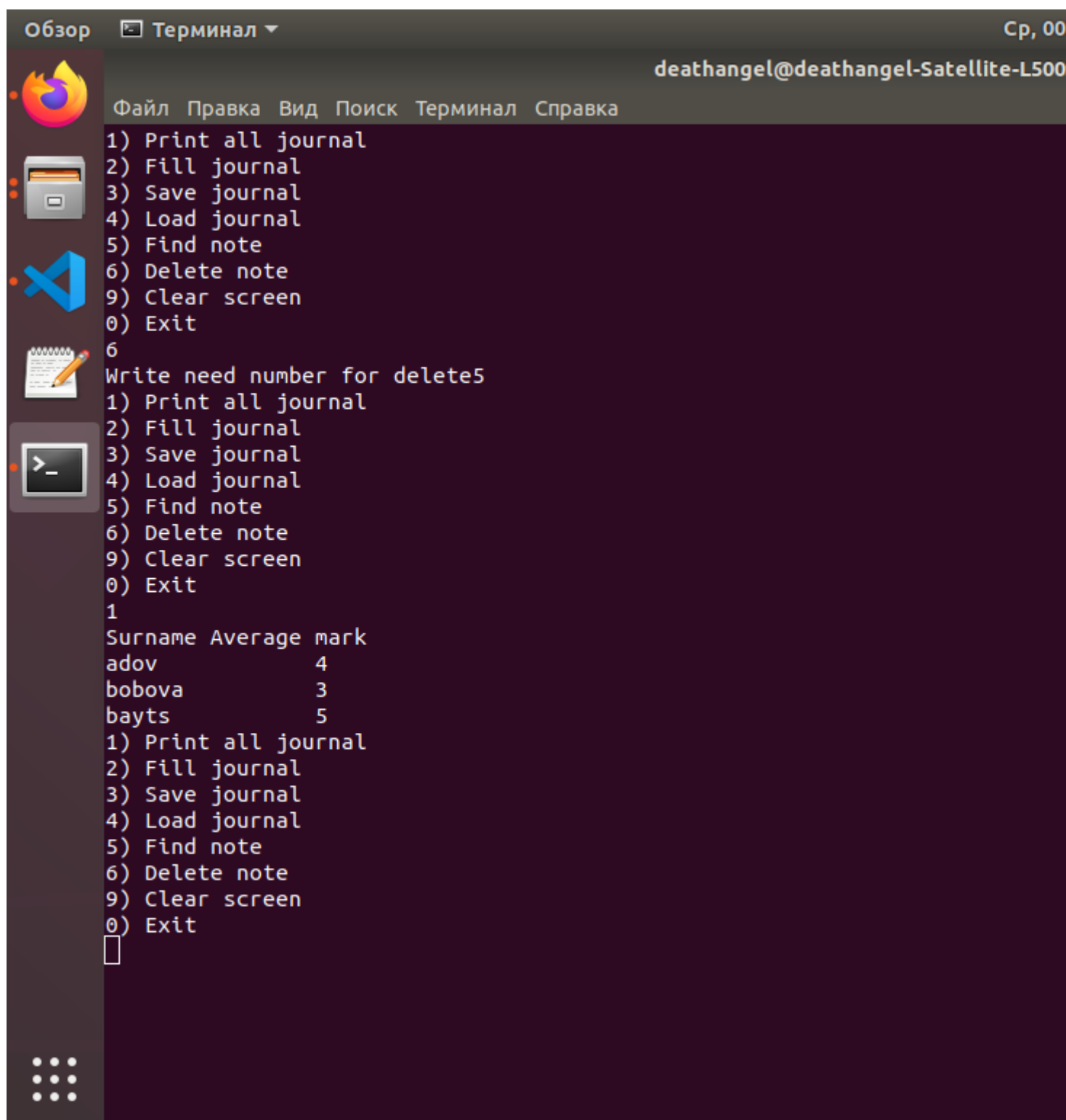
Запись списка в бинарный файл и считывание из него

Обзор Терминал Cp, 00 deathangel@deathangel-Satellite-L500

Файл Правка Вид Поиск Терминал Справка

1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
5
Write quantity letter in need surname
4
Write need surname
goga
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
5
Write quantity letter in need surname
4
Write need surname
adov
adov 4
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
█

Поиск существующего и не существующего элементов списка



Попытка удаления несуществующего элемента списка

Обзор Терминал Cp, 00

deathangel@deathangel-Satellite-L500

Файл Правка Вид Поиск Терминал Справка

9) Clear screen
0) Exit
1
Surname Average mark
adov 4
bobova 3
bayts 5
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
6
Write need number for delete3
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
1
Surname Average mark
adov 4
bobova 3
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
[]

Удаление третьего элемента в списке

Обзор Терминал deathangel@deathangel-Satellite-L500

Файл Правка Вид Поиск Терминал Справка

5) Find note
6) Delete note
9) Clear screen
0) Exit
1
Surname Average mark
adov 4
bobova 3
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
6
Write need number for delete1
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit
1
Surname Average mark
bobova 3
1) Print all journal
2) Fill journal
3) Save journal
4) Load journal
5) Find note
6) Delete note
9) Clear screen
0) Exit

Удаление первого элемента в списке