

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт № 8
«Информационные технологии и прикладная математика»

Курсовой проект
по курсу «Вычислительные системы»

Семестр 2

Задание 9

Студент: Хайруллина Ясмин Алмазовна

Группа: М8О-103Б-21

Руководитель: Севастьянов Виктор Сергеевич

Дата сдачи: 13.05.22

Москва, 2022

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ	3
2 ЗАДАЧИ	4
3 ФОРМУЛИРОВКА ЗАДАНИЯ	5
4 АЛГОРИТМ ПРОГРАММЫ	6
5 ОПИСАНИЕ ПРОГРАММЫ	7
6 ПРОГРАММА	8
7 РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ	13
8 ЗАКЛЮЧЕНИЕ	16

ВВЕДЕНИЕ

Данная курсовая работа направлена на изучение сортировки на языке программирования Си. В ходе работы полученные с помощью предоставленной информации и самостоятельного изучения материала будет составлена программа для решения предложенной задачи.

ЗАДАЧИ

1. Изучить материал по данной теме, поискать дополнительную информацию в сторонних источниках.
2. Составить программу сортировки таблицы указанным методом и двоичный поиск данных по ключу.
3. Производить работу в режиме меню.
4. Предоставить отчет.

ФОРМУЛИРОВКА ЗАДАНИЯ

Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице. Программа должна производить сортировку в трех случаях:

1. Элементы таблицы с самого начала упорядочены;
2. Элементы таблицы расставлены в обратном порядке;
3. Элементы таблицы не упорядочены.

Метод сортировки: быстрая сортировка Хоара (рекурсивный вариант).

Тип ключа: целый.

Содержание файла: ключ и фрагмент стихотворения.

АЛГОРИТМ ПРОГРАММЫ

1. В отдельном файле создать структуру таблицы, написать функции для обработки таблицы.
2. Написать функцию для считывания данных из файла, содержащего данные для сортировки.
3. Реализовать бинарный поиск по ключу.
4. Реализовать рекурсивный вариант сортировки Хоара.
5. Реализовать работу с программой через меню.

ОПИСАНИЕ ПРОГРАММЫ

Входные данные: файл ключами и фрагментом стихотворения.

Выходные данные: отсортированная таблица по ключу.

Используемые функции и структуры:

1. row – структура ряда (одной строки стихотворения)
2. quickSort — функция для быстрой сортировки Хоара (рекурсивный вариант)
3. printTable — функция для вывода таблицы
4. getRow – функция для считывания строки из файла и записи ее в структуру
5. swapRows – функция для перестановки двух рядов
6. binSearch – бинарный поиск по ключу
7. isSorted – проверка на отсортированность таблицы

ПРОГРАММА

Файл Makefile

```
laba: table.o main.o
    gcc table.o main.o
table.o : table.h table.c
    gcc -c table.c
main.o : table.h main.c
    gcc -c main.c
```

Файл table.h

```
#ifndef TABLE_H
#define TABLE_H

#include <stdio.h>

typedef struct _row
{
    long int _key;
    char _str[120];
} row;

void printTable(row *arr, const int size);
void getRow(FILE *stream, char *str, const int size);
void swapRows(row *r1, row *r2);
int binSearch(const row *arr, const int size, const long int key);
int isSorted(const row *arr, const int size);
void quickSort(row *arr, int left, int right);
#endif
```

Файл table.c

```
#include <time.h>
#include <stdlib.h>
#include "table.h"

void printTable(row *arr, const int size)
{
```



```

printf("┌──────────┐\n");
printf(" │ Ключ │ Значение │\n");

printf("└──────────┘\n");
for (int i = 0; i < size; i++)
    printf(" │ %9ld │ %50s\n", arr[i]._key, arr[i]._str);

printf("└──────────┘\n");
}

```

```

void getRow(FILE *file, char *str, const int size)
{
    int cnt = 0;
    char ch;
    while ((ch = getc(file)) != '\n' && cnt < size - 1)
    {
        str[cnt++] = ch;
    }
    str[cnt] = '\0'; //признак конца строки
}

```

```

void swapRows(row *r1, row *r2)
{
    row tmp;
    tmp = *r1;
    *r1 = *r2;
    *r2 = tmp;
}

```

```

int binSearch(const row *arr, const int size, const long int key)
{
    int start = 0, end = size - 1, mid;
    if (size <= 0)
        return -1;
    while (start < end)
    {
        mid = start + (end - start) / 2;

```

```

    if (arr[mid]._key == key)
        return mid;
    else if (arr[mid]._key < key)
        start = mid + 1;
    else
        end = mid;
}
if (arr[end]._key == key)
    return end;
return -1;
}

int isSorted(const row *arr, const int size)
{
    for (int i = 0; i < size - 1; i++)
        if (arr[i]._key > arr[i + 1]._key)
            return 0;
    return 1;
}

void quickSort(row *arr, int left, int right)
{
    int pivot;
    row p;          // разрешающий элемент
    int l_hold = left; //левая граница
    int r_hold = right; // правая граница
    pivot = arr[left]._key;
    p = arr[left];
    while (left < right) // пока границы не сомкнутся
    {
        while ((arr[right]._key >= pivot) && (left < right))
            right--;    // сдвигаем правую границу пока элемент [right] больше
[pivot]
        if (left != right) // если границы не сомкнулись
        {
            //arr[left]._key = arr[right]._key;
            //*arr[left]._str = *arr[right]._str; // перемещаем элемент [right] на место
разрешающего
            //printf("%s\t%s\t", arr[left]._str, arr[right]._str);
            swapRows(&arr[left], &arr[right]);
            left++; // сдвигаем левую границу вправо
        }
    }
}

```

```

while ((arr[left]._key <= pivot) && (left < right))
    left++; // сдвигаем левую границу пока элемент [left] меньше [pivot]
if (left != right) // если границы не сомкнулись
{
    //arr[right]._key = arr[left]._key;
    //*arr[right]._str = *arr[left]._str; // перемещаем элемент [left] на место
[right]
    swapRows(&arr[right], &arr[left]);
    right--; // сдвигаем правую границу влево
}
}
arr[left]._key = pivot;
// *arr[left]._str = *arr[l_hold]._str;
pivot = left;
left = l_hold;
right = r_hold;
if (left < right) // Рекурсивно вызываем сортировку для левой и правой части
массива
    quickSort(arr, left, pivot - 1);
if (right > pivot)
    quickSort(arr, pivot + 1, right);
}

```

Файл main.c

```

#include <stdio.h>
#include <stdlib.h>
#include "table.h"

void menu()
{
    printf("Меню:\n");
    printf("1) Вывести таблицу\n");
    printf("2) Найти элемент в таблице по ключу\n");
    printf("3) Отсортировать таблицу\n");
    printf("4) Меню\n");
    printf("5) Выход\n");
}

```

```

int main(void)
{
    const int N = 50;
    int i, cnt, rule;
    long key;
    char ch;
    row arr[N];
    char filename[100];
    printf("Введите название файла:\n");
    scanf("%s", filename);
    FILE *file = fopen(filename, "r");
    if (file == NULL)
    {
        printf("Файл не существует\n");
        return 0;
    }
    i = 0;
    while (i < N && fscanf(file, "%ld", &arr[i]._key) == 1)
    {
        fscanf(file, "%c", &ch);
        getRow(file, arr[i]._str, sizeof(arr[i]._str));
        i++;
    }
    fclose(file);
    cnt = i;
    menu();
    while (1)
    {
        scanf("%d", &rule);
        switch (rule)
        {
            case 1:
                printTable(arr, cnt);
                break;
            case 2:
                if (!isSorted(arr, cnt))
                {
                    printf("Таблица неотсортирована\n");
                }
                else
                {
                    printf("Введите ключ\n");
                    scanf("%ld", &key);
                }
            }
        }
    }
}

```

```

    i = binSearch(arr, cnt, key);
    if (i > -1)
    {
        printf("По ключу %ld найден элемент \"%s\"\n", key, arr[i]._str);
    }
    else
        printf("Элемент с данным ключом не найден\n");
}
break;
case 3:
    quickSort(arr, 0, cnt - 1);
    printf("Таблица отсортирована\n");
    break;
case 4:
    menu();
    break;
case 5:
    return 0;
    break;
default:
    printf("Выберите корректную команду из меню\n");
    break;
}
}
return 0;
}

```

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

```
jasmin@ubuntu:~$ make
```

```
gcc -c table.c
```

```
gcc -c main.c
```

```
gcc table.o main.o
```

```
jasmin@ubuntu:~$ ./a.out
```

Введите название файла:

input

Меню:

- 1) Вывести таблицу
- 2) Найти элемент в таблице по ключу
- 3) Отсортировать таблицу
- 4) Меню
- 5) Выход

1

Ключ	Значение	
7	Дым табачный воздух выел.	
6	Комната -	
5	глава в крученыховском аде.	
4	Вспомни -	
3	за этим окном	
2	впервые	
1	руки твои, иступленный, гладил.	

3

Таблица отсортирована

1

Ключ	Значение
1	руки твои, исступленный, гладил.
2	впервые
3	за этим окном
4	Вспомни -
5	глава в крученокховском аде.
6	Комната -
7	Дым табачный воздух выел.

2

Введите ключ

5

По ключу 5 найден элемент "глава в крученокховском аде."

5

jasmin@ubuntu:~\$

ЗАКЛЮЧЕНИЕ

В ходе данной работы была составлена программа на языке Си для реализации рекурсивного варианта быстрой сортировки Хоара, изучен бинарный поиск, проработан ранее полученный материал работы с файлами. Полученные знания, практика и опыт в поиски нужной информации в сторонних источниках привнесли большой вклад в мое развитие и будут помогать мне в дальнейшей моей работе.