

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Программирование»**  
**Тема: Обработка файлов .csv**

Студент гр. 6304

\_\_\_\_\_

Запевалов А.И.

Преподаватель

\_\_\_\_\_

Берленко Т.А.

Санкт-Петербург

2017

# ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

**Студент** Запевалов Алексей Ильич

**Группа** 6304

**Тема работы (проекта):** Двухнаправленные списки

Исходные данные:

Вариант 6

Требуется написать программу, на вход которой подаются две таблицы, которые хранятся в csv файлах. Первая таблица содержит информацию о ФИО студента и его Github аккаунте, email'e и номере группы. Вторая таблица содержит фамилию, имя, количество баллов за экзамен. Программа должна составить несколько новых таблиц (по количеству номеров групп), в которых содержится ФИО, Github аккаунт, email, оценка. Номер группы должен быть в начале каждой таблицы, таблицы разделяются двумя символами перевода строки. Результат должен быть сохранен в новой файле.

Программа получает параметры из входного потока. Параметры:

- input\_file\_1 - csv файл
- input\_file\_2 - csv файл

В случае, если программа получила некорректные параметры, то:

- не создается выходного файла
- выводится сообщение об ошибке "Fail with <имя параметра>".

Получить csv файл можно, сохранив таблицу (Excel/Google Таблицы/LibreOffice Calc) в формате csv. Откройте файл в текстовом редакторе, он выглядит примерно следующим образом:

Имя,Фамилия,Возраст

Иван,Иванов,22

Петр,Иванов,23

Антон,Иванов,24

Разделитель (в данном случае ',') в общем случае может быть другим. Мы будем использовать ','.

**Содержание пояснительной записки:**

- Содержание
- Введение
- Теоретические обоснования двухнаправленных списков
- Описание необходимых для работы со списком функций.
- Описание работы с git и github
- Заключение
- Код программы

**Предполагаемый объем пояснительной записки:**

Не менее 20 страниц.

**Дата выдачи задания:** 28.04.2017

**Дата сдачи реферата:**

**Дата защиты реферата:**

Студент

Запевалов А.И.

Преподаватель

Берленко Т.А.

## **АННОТАЦИЯ**

Задача была считать данные из двух файлов с разрушением .csv и создать новый файл, собрав эти данные.

Во время разработки были использованы текстовый редактор gedit, IDE Code::Blocks, LibreOffice Writer, LibreOfficeCalc.

## СОДЕРЖАНИЕ

Введение.....	6
1. Теория.....	7
1.1.Файлы .csv.....	7
1.2.Пример табличных данных, записанных в формате .csv.....	7
Исходный текст:.....	7
Результирующая таблица:.....	7
2.Создание ветки.....	7
3.Методы работы.....	8
3.1.Считывание файла в массив строк.....	8
3.2.Создание отсортированного массива с номерами групп.....	8
3.3.Поиск студентов соответствующей группы.....	9
3.4.Вывод данных.....	10
3.5 Код программы полностью.....	11
3.6 Примеры работы программы.....	16
3.7 Отправка файлов на github.com.....	18
список использованных источников.....	20

## **ВВЕДЕНИЕ**

### **Цель работы:**

Написать программу, на вход которой подаются две таблицы, которые хранятся в csv файлах. Программа должна составить несколько новых таблиц. Результат должен быть сохранен в новом файле.

## 1. ТЕОРИЯ

### 1.1. Файлы .csv

CSV (от англ. Comma-Separated Values — значения, разделённые запятыми) — текстовый формат, предназначенный для представления табличных данных. Ячейки каждой строки разделяются запятыми, строки разделяются символами переноса строки.

### 1.2. Пример табличных данных, записанных в формате .csv

**Исходный текст:**

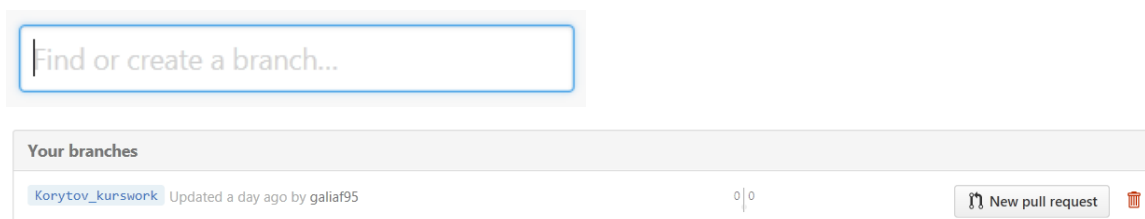
```
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""",4900.00
1996,Jeep,Grand Cherokee,"MUST SELL! air, moon roof, loaded",4799.00
```

**Результирующая таблица:**

1997	Ford	E350	ac, abs, moon	3000
1999	Chevy	Venture «Extended Edition»		4900
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799

## 2. СОЗДАНИЕ ВЕТКИ

Открыт репозиторий [.com/moevm/pr1-2016-2304](https://github.com/moevm/pr1-2016-2304). На нём с помощью поля “Find or create branch” создана ветка Alexey\_Zapevalov\_kurswork\_sem2



### 3. МЕТОДЫ РАБОТЫ

#### 3.1. Считывание файла в массив строк

```
void read(char* filename, char** array) //считываются строки из
файла
{
    char buf[BUFSZ];
    int j=0;
    FILE *file = fopen(filename,"r");
    while(!feof(file))
    {
        fgets(buf, BUFSZ-1, file);
        strcpy(array[j], buf);
        j++;
    }
    memset(array[j-1], 0, N*sizeof(char)); //последняя строка
считывается дважды, заполняем ее нулями
    fclose(file);
}
```

#### 3.2. Создание отсортированного массива с номерами групп

char\*\* returnNumbersOfGroups(char\*\* array) //нахождение номеров групп и их сортировка

```
{
    int i, j=0, m=0, flagNumberOfGroup=0;
    char** numbers = createArrayOfStrings(P, L);
    i=11;
    while (array[i][0])
    {
        while(numbers[j][0]!=0)
        {
            if (strcmp(numbers[j], array[i])==0)
            {
                flagNumberOfGroup=1;
                break;
            }
            j++;
        }
        if (flagNumberOfGroup==0)
        {
            strcpy(numbers[m], array[i]);
            m++;
        }
        flagNumberOfGroup=0;
        j=0;
        i+=6;
    }
}
```



```

    if (m==0)
    {
        printf("Fail with group numbers\n");
        exit(10);
    }
    i=0;
    qsort(numbers, m, L*sizeof(char), compare);
    return numbers;
}

```

Переменная `flagNumberOfGroup` остается нетронутой, если номера текущей группы нет в массиве. Если номеров не было вовсе, программа выдает ошибку. Переменная `i` изменяется увеличивается на 6 при каждой итерации, так как в одной исходной строке 6 видов данных.

### 3.3. Поиск студентов соответствующей группы

```

void findThem(char* number, char** arrayOneNoCommas, char**
arrayTwoNoCommas)
{//поиск студентов одной группы, нахождение соответствия со второй
таблицей
    int i=6, k=3, findIt=0;
    printTitle(arrayOneNoCommas, arrayTwoNoCommas);
    while(i<N-6)
    {
        if (strcmp(arrayOneNoCommas[i+5], number)==0)
        {
            while(k<N)
            {
                if (strcmp(arrayTwoNoCommas[k],
arrayOneNoCommas[i])==0&&strcmp(arrayTwoNoCommas[k+1],
arrayOneNoCommas[i+1])==0)
                {
                    findIt=1;
                    printThem(arrayOneNoCommas, i, arrayTwoNoCommas, k);
                    break;
                }
                k+=3;
            }
            if (findIt==0)
            {
                printf("Fail with Name or Surname comparison");
                remove(output_file);
                exit(5);
            }
            findIt=0;
        }
    }
}

```

```

        i+=6;
    }
    printGap();
}

```

Сначала сравниваются номера групп из первой таблицы с нужным номером группы, после чего имена и фамилии из первой таблицы сравниваются с именами и фамилиями из второй таблицы. В случае, если имена или фамилии не совпали, выдается ошибка и файл удаляется. Иначе данные записываются в файл, выводятся названия данных и пропуск строк между группами.

### 3.4. Вывод данных

```

void printThem(char** arrayOne, int i, char** arrayTwo, int k)//Вывод
данных из строк

```

```

{
    FILE *file;
    file=fopen(output_file, "a");
    int c=0;
    for (c=0; c<5; c++)
    {
        fprintf(file, "%s,", arrayOne[i+c]);
    }
    fprintf(file, "%s", arrayTwo[k+2]);
    fclose(file);
}

```

```

void printNumber(char* number)//Вывод номера группы

```

```

{
    FILE *file;
    file=fopen(output_file, "a");
    fprintf(file, "%s", number);
    fclose(file);
}

```

```

void printTitle(char** arrayOne, char** arrayTwo)//Вывод заголовка

```

```

{
    int i=0;
    FILE *file;
    file=fopen(output_file, "a");
    for (i=0; i<5; i++)
    {
        fprintf(file, "%s,", arrayOne[i]);
        if (i==4)
            fprintf(file, "%s", arrayTwo[2]);
    }
    fclose(file);
}

```

```

}

void printGap()//печать пробела между группами
{
    FILE *file;
    file=fopen(output_file, "a");
    fprintf(file, "\n\n");
    fclose(file);
}

```

Каждый элемент дописывается в файл без удаления предыдущих.

### 3.5 Код программы полностью.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>

#define input_file_1 "./Worksheet 1.csv"
#define input_file_2 "./Worksheet 2.csv"
#define output_file "./Worksheet 3.csv"
#define BUFSZ 1000
#define N 10000
#define P 30
#define L 4

int compare(const void** a, const void** b)//сравнение
{
    return strcmp(*(char**)a, *(char**)b);
}

int checkWriteableFile(char* sourcefile) //Проверка на возможность записи
в файл
{
    FILE* file = fopen(sourcefile, "a");
    if (errno)
    {
        printf("Problem with file %s\n", sourcefile);
        return 0;
    }
    else
        return 1;
    fclose(file);
}

int checkReadableFile(char* sourcefile) //Проверка на наличие исходных
файлов

```

```

{
    FILE* file =fopen(sourcefile, "r");
    if (errno)
    {
        printf("Problem with file %s\n", sourcefile);
        return 0;
    }
    else
        return 1;
    fclose(file);
}

void printThem(char** arrayOne, int i, char** arrayTwo, int k)//ВЫВОД
данных из строк
{
    FILE *file;
    file=fopen(output_file, "a");
    int c=0;
    for (c=0; c<5; c++)
    {
        fprintf(file, "%s,", arrayOne[i+c]);
    }
    fprintf(file, "%s", arrayTwo[k+2]);
    fclose(file);
}

void printNumber(char* number)//ВЫВОД номера группы
{
    FILE *file;
    file=fopen(output_file, "a");
    fprintf(file, "%s", number);
    fclose(file);
}

void printTitle(char** arrayOne, char** arrayTwo)//ВЫВОД заголовка
{
    int i=0;
    FILE *file;
    file=fopen(output_file, "a");
    for (i=0; i<5; i++)
    {
        fprintf(file, "%s,", arrayOne[i]);
        if (i==4)
            fprintf(file, "%s", arrayTwo[2]);
    }
    fclose(file);
}

```

```

    }

    void printGap()//печать пробела между группами
    {
        FILE *file;
        file=fopen(output_file, "a");
        fprintf(file, "\n\n");
        fclose(file);
    }

    void freeArray(char** arr)
    {
        int i=0;
        while (arr[i])
        {
            free(arr[i]);
            i++;
        }
        free(arr);
    }

    void findStudents(char* number, char** arrayOneNoCommas, char**
arrayTwoNoCommas)
    {
        //поиск студентов одной группы, нахождение соответствия со второй
таблицей
        int i=6, k=3, findIt=0;
        printTitle(arrayOneNoCommas, arrayTwoNoCommas);
        while(i<N-6)
        {
            if (strcmp(arrayOneNoCommas[i+5], number)==0)
            {
                while(k<N)
                {
                    if (strcmp(arrayTwoNoCommas[k],
arrayOneNoCommas[i])==0&&strcmp(arrayTwoNoCommas[k+1],
arrayOneNoCommas[i+1])==0)
                    {
                        findIt=1;
                        printThem(arrayOneNoCommas, i, arrayTwoNoCommas, k);
                        break;
                    }
                    k+=3;
                }
                if (findIt==0)
                {

```

```

        printf("Fail with Name or Surname comparison");
        remove(output_file);
    }
    findIt=0;
}
i+=6;
k=3;
}
printGap();
}

```

```

char** createArrayOfStrings(int rows, int columns)//создается массив
{
    int i;
    char** array = (char**)malloc(rows*sizeof(char*));
    for (i=0; i<rows; i++)
    {
        array[i]=(char*)malloc(columns*sizeof(char));
    }
    return array;
}

```

их сортировка

```

char** returnNumbersOfGroups(char** array) //нахождение номеров групп и
{
    int i, j=0, m=0, flagNumberOfGroup=0;
    char** numbers = createArrayOfStrings(P, L);
    i=11;
    while (array[i][0])
    {
        while(numbers[j][0]!=0)
        {
            if (strcmp(numbers[j], array[i])==0)
            {
                flagNumberOfGroup=1;
                break;
            }
            j++;
        }
        if (flagNumberOfGroup==0)
        {
            strcpy(numbers[m], array[i]);
            m++;
        }
        flagNumberOfGroup=0;
    }
}

```

```

        j=0;
        i+=6;
    }
    if (m==0)
    {
        printf("Fail with group numbers\n");
    }
    i=0;
    qsort(numbers, m, L*sizeof(char), compare);
    return numbers;
}

void read(char* filename, char** array) //считываются строки из файла
{
    char buf[BUFSZ];
    int j=0;
    FILE *file = fopen(filename,"r");
    while(!feof(file))
    {
        fgets(buf, BUFSZ-1, file);
        strcpy(array[j], buf);
        j++;
    }
    memset(array[j-1], 0, N*sizeof(char)); //последняя строка считывается
    дважды, заполняем ее нулями
    fclose(file);
}

void checkCompleteness(char** array, int commas) //проверяем количество
запятых в строках
{
    int i=0, j=0, counter=0;
    while(array[i][j]!=0)
    {
        while(array[i][j])
        {
            if (array[i][j]=='(',')')
            {
                counter++;
            }
            j++;
        }
        j=0;
        if (counter!=commas)
        {
            printf("Fail with commas(Line %d)", i+1);

```

```

        }
        counter=0;
        i++;
    }
}

char** arrayNoCommas(char** array) //разделяем строки с данными на
тоkens запятыми и заполняем массив
{
    char** arrNoCommas=createArrayOfStrings(N, P);
    int i=0, j=0;
    char delim[ ]=",";
    for (j=0; j<N; j++)
    {
        char* p=strtok(array[j], delim);
        while (p!=NULL)
        {
            strcpy(arrNoCommas[i], p);
            p=strtok(NULL, delim);
            i++;
        }
    }
    return arrNoCommas;
}

void findGroups(char** arrayOne, char** arrayTwo) //разделяем данные,
находим номера групп
{
    int i=0;
    char** arrOneNoCommas=arrayNoCommas(arrayOne);
    char** arrTwoNoCommas=arrayNoCommas(arrayTwo);
    char** groupNumbersArray = returnNumbersOfGroups(arrOneNoCommas);
    while (groupNumbersArray[i][0]!=0)
    {
        printNumber(groupNumbersArray[i]);
        findStudents(groupNumbersArray[i], arrOneNoCommas,
arrTwoNoCommas);
        i++;
    }
    i=0;
    freeArray(arrOneNoCommas);
    freeArray(arrTwoNoCommas);
    freeArray(groupNumbersArray);
}

```



```

    int main() //вызываются функции по созданию массивов, считыванию
    файлов и нахождению номеров групп
    {
                                                                    if
    (checkReadableFile(input_file_1)&&checkReadableFile(input_file_2)&&checkWriteabl
    eFile(output_file))
    {
        char** arrayOne = createArrayOfStrings(N, N);
        char** arrayTwo = createArrayOfStrings(N, N);
        read(input_file_1, arrayOne);
        read(input_file_2, arrayTwo);
        checkCompleteness(arrayOne, 5);
        checkCompleteness(arrayTwo, 2);
        findGroups(arrayOne, arrayTwo);
        freeArray(arrayOne);
        freeArray(arrayTwo);
    }
    return 0;
}

```

### 3.6 Примеры работы программы

Открыть ▾	Сохранить
<div>1230</div> <div> <div>Фамилия,Имя,Отчество,Github,e-mail,Оценка</div> <div>Никитина,Антонина,Александров,аккаунт4,e-mail4,4</div> <div>Гордеев,Фрол,Арсеньевич,аккаунт17,e-mail17,2</div> <div>Никитин,Михаил,Филатович,аккаунт18,e-mail18,3</div> <div>Андреева,Лариса,Арсеньевна,аккаунт43,e-mail43,4</div> <div>Дементьев,Мэлор,Мартынович,аккаунт45,e-mail45,2</div> </div>	
<div>1231</div> <div> <div>Фамилия,Имя,Отчество,Github,e-mail,Оценка</div> <div>Федосеев,Иван,Григорьевич,аккаунт7,e-mail7,1</div> <div>Молчанова,Вероника,Парфеньевна,аккаунт10,e-mail10,4</div> <div>Лапин,Валерьян,Богданович,аккаунт19,e-mail19,3</div> <div>Родионова,Надежда,Владислав,аккаунт20,e-mail20,4</div> <div>Власов,Сергей,Пантелеймон,аккаунт40,e-mail40,1</div> <div>Хохлов,Христофор,Яковович,аккаунт41,e-mail41,4</div> <div>Молчанов,Николай,Семёнович,аккаунт47,e-mail47,3</div> </div>	
<div>1232</div> <div> <div>Фамилия,Имя,Отчество,Github,e-mail,Оценка</div> <div>Горбачёва,Зоя,Якуновна,аккаунт3,e-mail3,0</div> <div>Сазонов,Лаврентий,Донатович,аккаунт8,e-mail8,0</div> <div>Гришина,Людмила,Пётровна,аккаунт9,e-mail9,5</div> <div>Романова,Надежда,Пётровна,аккаунт13,e-mail13,1</div> <div>Поляков,Станислав,Ростислав,аккаунт15,e-mail15,4</div> <div>Герасимов,Христофор,Константин,аккаунт21,e-mail21,4</div> <div>Пахомов,Протасий,Ильяович,аккаунт26,e-mail26,1</div> <div>Фокина,Ксения,Андреевна,аккаунт31,e-mail31,4</div> <div>Кононова,Лидия,Якововна,аккаунт33,e-mail33,1</div> <div>Егорова,Надежда,Матвеевна,аккаунт34,e-mail34,1</div> <div>Фадеева,Наина,Арсеньевна,аккаунт37,e-mail37,2</div> </div>	
<div>1233</div> <div> <div>Фамилия,Имя,Отчество,Github,e-mail,Оценка</div> <div>Степанов,Леонид,Николаевич,аккаунт1,e-mail1,2</div> <div>Колесникова,Алла,Романовна,аккаунт2,e-mail2,5</div> <div>Маслов,Филат,Пантелеймон,аккаунт5,e-mail5,4</div> </div>	
CSV ▾	Ширина табуляции: 8 ▾         Стр 1, Стлб 1 ▾         ВСТ

1230

Фамилия	Имя	Отчество	Github	e-mail	Оценка
Никитина	Антонина	Александров	аккаунт4	e-mail4	4
Гордеев	Фрол	Арсеньевич	аккаунт17	e-mail17	2
Никитин	Михаил	Филатович	аккаунт18	e-mail18	3
Андреева	Лариса	Арсеньевна	аккаунт43	e-mail43	4
Дементьев	Мэлор	Мартынович	аккаунт45	e-mail45	2

1231

Фамилия	Имя	Отчество	Github	e-mail	Оценка
Федосеев	Иван	Григорьевич	аккаунт7	e-mail7	1
Молчанова	Вероника	Парфеньевна	аккаунт10	e-mail10	4
Лапин	Валерьян	Богданович	аккаунт19	e-mail19	3
Родионова	Надежда	Владислав	аккаунт20	e-mail20	4
Власов	Сергей	Пантелеймон	аккаунт40	e-mail40	1
Хохлов	Христофор	Яковович	аккаунт41	e-mail41	4
Молчанов	Николай	Семёнович	аккаунт47	e-mail47	3

1232

Фамилия	Имя	Отчество	Github	e-mail	Оценка
Горбачёва	Зоя	Якуновна	аккаунт3	e-mail3	0
Сазонов	Лаврентий	Донатович	аккаунт8	e-mail8	0
Гришина	Людмила	Пётровна	аккаунт9	e-mail9	5
Романова	Надежда	Пётровна	аккаунт13	e-mail13	1
Поляков	Станислав	Ростислав	аккаунт15	e-mail15	4
Герасимов	Христофор	Константин	аккаунт21	e-mail21	4
Пахомов	Протасий	Ильевич	аккаунт26	e-mail26	1
Фокина	Ксения	Андреевна	аккаунт31	e-mail31	4
Кононова	Лидия	Якововна	аккаунт33	e-mail33	1
Егорова	Надежда	Матвеевна	аккаунт34	e-mail34	1
Фадеева	Наина	Арсеньевна	аккаунт37	e-mail37	2

1233

Фамилия	Имя	Отчество	Github	e-mail	Оценка
Степанов	Леонид	Николаевич	аккаунт1	e-mail1	2
Колесникова	Алла	Романовна	аккаунт2	e-mail2	5
Маслов	Филат	Пантелеймон	аккаунт5	e-mail5	4
Нестерова	Марина	Ильевна	аккаунт11	e-mail11	2
Щербаков	Даниил	Куприянович	аккаунт12	e-mail12	3
Пестов	Владимир	Николаевич	аккаунт16	e-mail16	5
Шилов	Федот	Владислав	аккаунт28	e-mail28	0
Лапина	Александра	Аркадьевна	аккаунт30	e-mail30	2
Тарасов	Аристарх	Донатович	аккаунт44	e-mail44	3
Лыткин	Германн	Авдеевич	аккаунт49	e-mail49	4

1234

Фамилия	Имя	Отчество	Github	e-mail	Оценка
Дмитриев	Арсений	Ильевич	аккаунт23	e-mail23	1
Соловьёв	Денис	Павлович	аккаунт24	e-mail24	1
Маслов	Ириней	Егорович	аккаунт27	e-mail27	1
Копылов	Федот	Евгеньевич	аккаунт32	e-mail32	2
Антонова	Зоя	Кондратовна	аккаунт35	e-mail35	1
Николаев	Руслан	Демьянович	аккаунт36	e-mail36	1
Максимов	Пётр	Анатольевич	аккаунт38	e-mail38	5
Некрасова	Пелагея	Никитевна	аккаунт39	e-mail39	2
Прохоров	Альберт	Филатович	аккаунт42	e-mail42	4
Уварова	Валерия	Мэлоровна	аккаунт48	e-mail48	3
Зими́на	Алина	Аксентьевна	аккаунт50	e-mail50	1

1235

Фамилия	Имя	Отчество	Github	e-mail	Оценка
---------	-----	----------	--------	--------	--------

Изменили фамилию в исходных данных только в одной таблице:

Фамилия, Имя, Отчество
Степан, Леонид, Никола
Колесникова, Алла, Ром

Результат:

### 3.7 Отправка файлов на github.com

```
alexey@alexey-Inspiron-3542:~/ok/kursach sem2$ git checkout -b
Alexey_Zapevalov_Coursework_sem2
M zapevalov_alexey_6304/lab1/main.c
Переключено на новую ветку «Alexey_Zapevalov_Coursework_sem2»
alexey@alexey-Inspiron-3542:~/ok/kursach sem2$ git add
./Alexey_Zapevalov_kurswork_sem2/
alexey@alexey-Inspiron-3542:~/ok/kursach sem2$ git add
./Alexey_Zapevalov_kurswork_sem2/
alexey@alexey-Inspiron-3542:~/ok/kursach sem2$ git commit -m "added
coursework code"
[Alexey_Zapevalov_Coursework_sem2 2d5eae4] added coursework code
3 files changed, 338 insertions(+)
create mode 100755 kursach
sem2/Alexey_Zapevalov_kurswork_sem2/Worksheet 1.csv
create mode 100755 kursach
sem2/Alexey_Zapevalov_kurswork_sem2/Worksheet 2.csv
create mode 100644 kursach sem2/Alexey_Zapevalov_kurswork_sem2/main.c
alexey@alexey-Inspiron-3542:~/ok/kursach sem2$ git push
https://github.com/moevm/pr1-2016-6304.git Alexey_Zapevalov_Coursework_sem2
Username for 'https://github.com': almostpurplerain
Password for 'https://almostpurplerain@github.com':
```

```
Подсчет объектов: 23, готово.  
Delta compression using up to 4 threads.  
Сжатие объектов: 100% (16/16), готово.  
Запись объектов: 100% (23/23), 21.93 KiB | 0 bytes/s, готово.  
Total 23 (delta 0), reused 0 (delta 0)  
To https://github.com/moevm/pr1-2016-6304.git  
* [new branch] Alexey_Zapevalov_Coursework_sem2 ->  
Alexey_Zapevalov_Coursework_sem2  
alexey@alexey-Inspiron-3542:~/ok/kursach sem2$
```

## **ЗАКЛЮЧЕНИЕ**

Цель работы достигнута – создана программа, которая записывает исходные данные в файл после их проверки и обработки.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Команда man в терминале Linux

URL: <http://man.he.net/>

Керниган, Ритчи. Язык программирования Си