**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

Курсовая РАБОТА

**по дисциплине «Программирование»**

Тема: Файлы .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):** Двунаправленные 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'е 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 в 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 двунаправленных 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](#__RefHeading___Toc1330_374830139)

[1. Теория 7](#__RefHeading___Toc1334_374830139)

[1.1.Файлы .csv 7](#__RefHeading___Toc1336_374830139)

[1.2.Пример табличных данных, записанных в формате .csv 7](#__RefHeading___Toc1338_374830139)

[Исходный текст: 7](#__RefHeading___Toc1340_374830139)

[Результирующая таблица: 7](#__RefHeading___Toc1342_374830139)

[2.Создание ветки 7](#__RefHeading___Toc1344_374830139)

[3.Методы работы 8](#__RefHeading___Toc1346_374830139)

[3.1.Считывание файла в массив строк 8](#__RefHeading___Toc1348_374830139)

[3.2.Создание отсортированного массива с номерами групп 8](#__RefHeading___Toc1350_374830139)

[3.3.Поиск студентов соответствующей группы 9](#__RefHeading___Toc1352_374830139)

[3.4.Вывод данных 10](#__RefHeading___Toc1354_374830139)

[3.5 Код программы полностью. 11](#__RefHeading___Toc1356_374830139)

[3.6 Примеры работы программы 16](#__RefHeading___Toc1360_374830139)

[3.7 Отправка файлов на github.com 18](#__RefHeading___Toc1089_1457664996)

[список использованных источников 20](#__RefHeading___Toc1358_374830139)

# **В****ведение**

**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. **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. **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

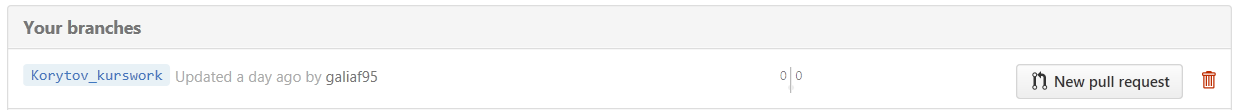
Результирующая таблица:

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

1. **Создание ветки**

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





# **М****етоды работы**

* 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);

}

* 1. **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.

* 1. **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.

* 1. **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>

#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 1000

#define P 30

#define L 4

int compare(const void\*\* a, const void\*\* b)//сравнение

{

return strcmp(\*(char\*\*)a, \*(char\*\*)b);

}

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 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();

}

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;

}

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;

}

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);

exit(2);

}

counter=0;

i++;

}

}

char\*\* arrayNoCommas(char\*\* array) //разделяем строки с данными на токены 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\*\* arrayOneNoCommas=arrayNoCommas(arrayOne);

char\*\* arrayTwoNoCommas=arrayNoCommas(arrayTwo);

char\*\* groupNumbersArray = returnNumbersOfGroups(arrayOneNoCommas);

while (groupNumbersArray[i][0]!=0)

{

printNumber(groupNumbersArray[i]);

findThem(groupNumbersArray[i], arrayOneNoCommas, arrayTwoNoCommas);

i++;

}

}

int main() //вызываются функции по созданию массивов, считыванию файлов и нахождению групп

{

char\*\* arrayOne = createArrayOfStrings(N, N);

char\*\* arrayTwo = createArrayOfStrings(N, N);

read(input\_file\_1, arrayOne);

read(input\_file\_2, arrayTwo);

checkCompleteness(arrayOne, 5);

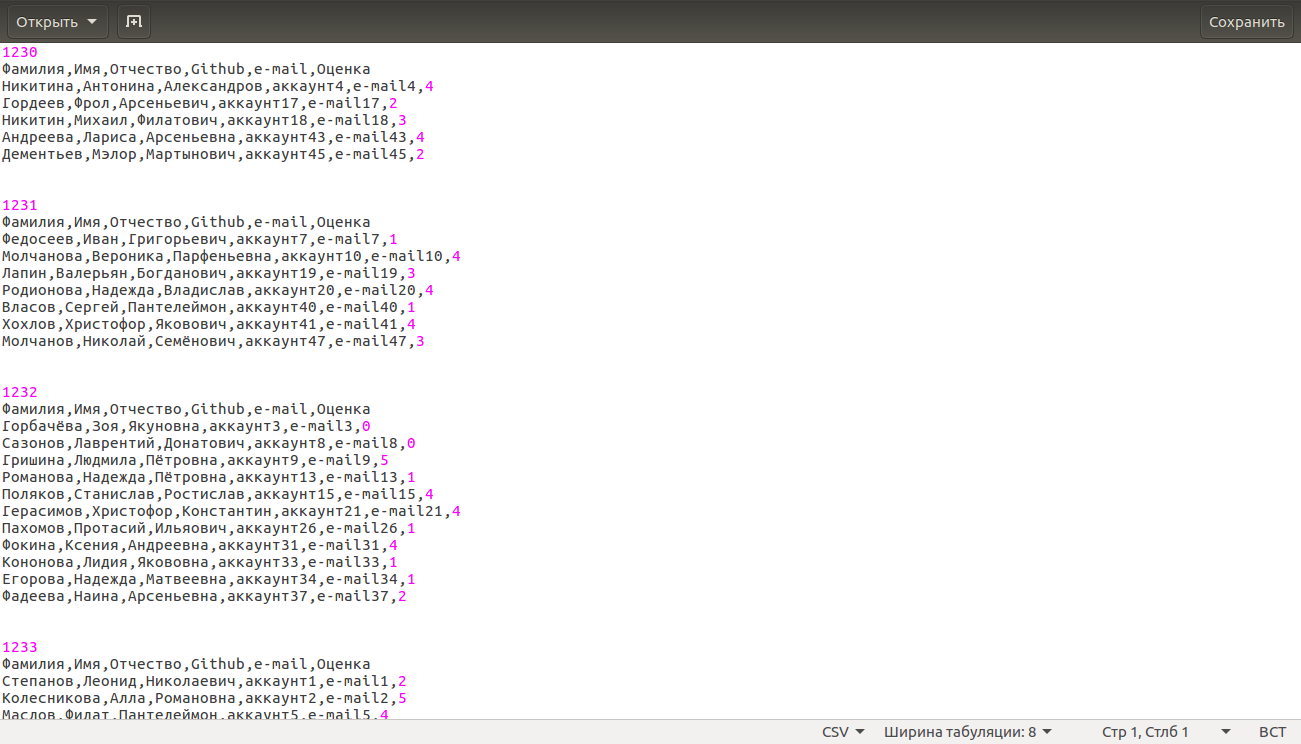
checkCompleteness(arrayTwo, 2);

findGroups(arrayOne, arrayTwo);

return 0;

}

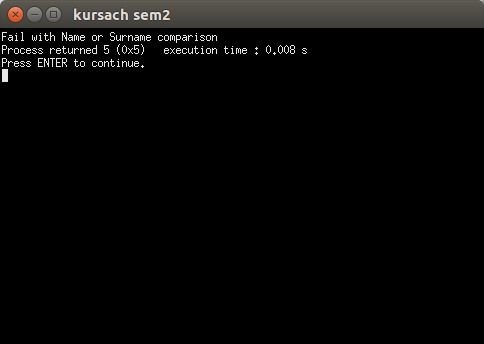
**3.6 Примеры работы программы**

****



Изменили фамилию в исходных данных только в одной таблице:

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