ГУАП

КАФЕДРА № 83

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ст. препод. |  |  |  | К.А. Жиданов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
|  |
| по курсу: ЯЗЫКИ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3145 |  |  |  | Я.Р.Резанова |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Вариант 1**

Отсортировать строки по заданной колонке

**Цель работы**

В .CSV файле хранятся значения зарплат. Требуется написать консольное приложение, которые считает данные и проведёт их обработку. Имя файла с данными и параметры задаются в командной строке.

**Ход работы**

1. Реализовала на языке Си структуру, описывающую строку в таблице:

typedef struct data {

char name[30];

char surname[30];

char gender[30];

char occupation[30];

char salary[30];

} data;

1. Реализовала на языке Си функцию вывода:

void printdata(data \*struc) {

printf("%s\n",struc->name);

printf("%s\n",struc->surname);

printf("%s\n",struc->gender);

printf("%s\n",struc->occupation);

printf("%s\n",struc->salary);

}

1. Реализовала операторы сравнения для всех столбцов:

int comparator1(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->name, ps1->name);

}

int comparator2(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->surname, ps1->surname);

}

int comparator3(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->gender, ps1->gender);

}

int comparator4(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->occupation, ps1->occupation);

}

int comparator5(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->salary, ps1->salary);

}

**4)**Удаление реализовано через отвязывание узла и переноса хвоста или головы списка на следующий элемент:

void\* popFront(DblLinkedList \*list) {

Node \*prev;

void \*tmp;

if (list->head == NULL) {

exit(2);

}

prev = list->head;

list->head = list->head->next;

if (list->head) {

list->head->prev = NULL;

}

if (prev == list->tail) {

list->tail = NULL;

}

tmp = prev->value;

free(prev);

list->size--;

return tmp;

}

void\* popBack(DblLinkedList \*list) {

Node \*next;

void \*tmp;

if (list->tail == NULL) {

exit(4);

}

next = list->tail;

list->tail = list->tail->prev;

if (list->tail) {

list->tail->next = NULL;

}

if (next == list->head) {

list->head = NULL;

}

tmp = next->value;

free(next);

list->size--;

return tmp;

}

**5)**Реализовала меню, с помощью которого можно отсортировать по нужному столбцу:

switch (mode)

{

case 1:

qsort(arr, n, sizeof(data),comparator1);

break;

case 2:

qsort(arr, n, sizeof(data),comparator2);

break;

case 3:

qsort(arr, n, sizeof(data),comparator3);

break;

case 4:

qsort(arr, n, sizeof(data),comparator4);

break;

case 5:

qsort(arr, n, sizeof(data),comparator5);

break;

default:

break;

}

**6)**Полный код программы:

#include <stdio.h>

#include <assert.h>

#include <malloc.h>

#include "string.h"

#include "stdlib.h"

typedef struct data {//структура строки таблицы

char name[30];

char surname[30];

char gender[30];

char occupation[30];

char salary[30];

} data;

void printdata(data \*struc) {//функция вывода строки

printf("%s\n",struc->name);

printf("%s\n",struc->surname);

printf("%s\n",struc->gender);

printf("%s\n",struc->occupation);

printf("%s\n",struc->salary);

}

int comparator1(const void\* p0, const void\* p1)//операторы сравнения

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->name, ps1->name);

}

int comparator2(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->surname, ps1->surname);

}

int comparator3(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->gender, ps1->gender);

}

int comparator4(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->occupation, ps1->occupation);

}

int comparator5(const void\* p0, const void\* p1)

{

struct data\* ps0 = (struct data\*) p0;

struct data\* ps1 = (struct data\*) p1;

return strcmp( ps0->salary, ps1->salary);

}

int main() {

char path[100]="data.csv";

FILE \*file = fopen(path, "r");//открываем файл для заполнения массива

assert(file != NULL);

char line[100];

int n=0;

while (fgets(line, sizeof(line), file) != NULL) {

n++;

}

fclose(file);

file = fopen(path, "r");

data\* arr=malloc(n\*sizeof(data));

int i=0;

fgets(line, sizeof(line), file);

while (fgets(line, sizeof(line), file) != NULL) {//копируем все данные в массив

strcpy(arr[i].name,strtok(line, ",\""));

strcpy(arr[i].surname,strtok(NULL, ",\""));

strcpy(arr[i].gender,strtok(NULL, ",\""));

strcpy(arr[i].occupation,strtok(NULL, ",\""));

strcpy(arr[i].salary,strtok(NULL, ",\""));

i++;

}

fclose(file);

printf("enter mode(1-name,2-surname,3-gender, 4-occupation, 5-salary):");

int mode=0;

scanf("%d",&mode);

switch (mode)

{//меню для выбора столбца сортьировки

case 1:

qsort(arr, n, sizeof(data),comparator1);

break;

case 2:

qsort(arr, n, sizeof(data),comparator2);

break;

case 3:

qsort(arr, n, sizeof(data),comparator3);

break;

case 4:

qsort(arr, n, sizeof(data),comparator4);

break;

case 5:

qsort(arr, n, sizeof(data),comparator5);

break;

default:

break;

}

for (int j=0;j<n;j++){

printdata(arr + j);//вывод массива в консоль

printf("\n");

}

remove ("path");

FILE \*f = fopen(path, "w");

if (f == NULL) return -1;

fprintf(f, "\"First Name\",\"Last Name\",\"Gender\",\"Occupation\",\"Salary\"");

for (int j=0;j<n;j++) {

fprintf(f, "\"%s\",\"%s\",\"%s\",\"%s\",\"%s\"\n", arr[j].name, arr[j].surname, arr[j].gender,arr[j].occupation,arr[j].salary); //вывод массива в файл

}

fclose(f);

free(arr);

}

**Вывод**

Изучила методы работы с txt файлами в Си.