

Московский Авиационный Институт
(Национальный Исследовательский Институт)

Факультет прикладной математики

КУРСОВАЯ РАБОТА

по курсам

«Инструментальные средства UNIX, алгоритмы и
структуры данных»

Задание VI

«Обработка последовательной файловой структуры
на языке Си»

Студент: Пивницкий Д.

Группа: М8о-101Б-19

Преподаватель: Сорокин С.

Оценка: _____

Дата: _____

Москва, 2019

Оглавление

Введение	3
Общий метод решения	4
Общие сведения о программе	5
Функциональное назначение	6
Описание программы	7
Описание функции программы	8
Используемые переменные	9
Входные значения	10
Протокол	12
Заключение	15

Введение

Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом. Составить программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей (15-20). Распечатать содержимое сгенерированного файла в виде таблицы и выполнить над ним заданное действие для 2-3 значений параметров запроса *p*, распечатав результат. Действие по выборке данных из файла оформить с параметрами запроса, вводимыми из стандартного входного текстового файла, или получаемых из стандартной строки ввода UNIX. Получение параметров из командной строки производится с помощью стандартных библиотечных функций `argc` и `argv`. Число наборов текстов должно быть не менее трех.

Содержимое и структура данных:

Сведения о вступительных экзаменах абитуриентов: фамилия, инициалы, пол, номер школы, наличие медали, оценки в баллах и зачет/незачёт по сочинению

Действия:

Найти абитуриентов-медалистов, не набравших проходной балл *p*

Общий метод решения

Создаем 3 текстовых файла, в которые записываем все исходные данные. (по 15 студентов на файл).

Считываем файл, указанный в 1 аргументе командной оболочки, в обусловленном формате, выводим в виде таблицы, создаём условие, фильтруем студентов, подходящих под условие и выводим их в виде таблицы. Повторяем эту операцию трижды.

Общие сведения о программе

Аппаратное обеспечение: ноутбук Lenovo Z570

ОС: Linux MINT Tessa

Язык и система программирования: GNU C

Стандарт языка: C99

Число строк программы: 52

Компиляция программы в терминале: gcc main6kr.c -o 6kr

Вызов программы: ./6kr (тестовый файл)

Функциональное назначение

Задача программы состоит в считывании текстового файла в корректном формате; записи данных в память программы и фильтрации студентов по критериям. В файле содержатся данные по студентам: каждый студент представлен в файле строкой, где строки — это набор данных через пробел, а именно фамилия, инициалы, пол, номер школы, есть ли медаль(+ или -), оценки за предметы: математика, физика, литература и зачет/незачет по сочинению(в виде +/-).

Описание программы

1. Подключаем библиотеки **stdio.h**, **stdlib.h**
2. Объявление структуры студента
3. Объявление функции **main** с аргументами **argc**, **argv**
4. Определение имени файла, данного пользователем в качестве первого аргумента, либо заданного по умолчанию, если пользователь не указал названия
5. Задание количества студентов в файле в виде константы **count** типа **int**
6. Создание массивов студентов: исходных и отфильтрованных по *p*
7. Считывание данных из файла в указанном формате и создание объектов студента
8. Запрос у пользователя переменной *p*
9. Вывод заголовка таблицы
10. Проход по студентам и их фильтрация + запись подходящих студентов в массив и вывод их на экран в виде таблицы
11. Если никого не нашлось, вывести «NO DATA»
12. Вывести закрывание таблицы

Описание функции программы

Функция/выражение	Описание
FILE *file	Объявление дескриптора файла
fopen(filename, "r");	Открывает файл в режиме считывания текста
calloc(count, sizeof(Student));	Возвращает указатель на 1 байт выделенной памяти для массива с элементами Student
feof(file)	Проверяет, достиг ли файл конца
fscanf	Считывает данные из файла в указанном формате
printf	Печатает текст и переменные в указанном формате
puts	Печатает только текст
fclose(file)	Закрывает файл

Используемые переменные

Переменная	Тип	Значение
filename	char*	Название файла для считывания
argc	int	Кол-во переданных аргументов программе
argv	char**	Сами аргументы
count	const int	Кол-во студентов в файле
file	FILE*	Файл для считывания
students	Student*	Студенты из базы данных
result	Student*	Отфильтрованные студенты
i	int	Индекс, чтобы проходиться по студентам
ires	int	Индекс, чтобы проходиться по массиву отфильтрованных студентов
s	Student	Студент, обрабатываемый в данный момент
p	int	P из условия
j	int	Счётчик для студентов

Входные значения

test.txt:

Whjherx EX female 1899 - 3 3 4 +
Mblpqvkcg WA female 855 + 4 3 3 +
Kwhhgb EK female 2355 + 4 4 2 -
Bzuulhvwxu QA female 692 - 5 2 3 +
Qctzqrzzb OE female 1485 - 2 2 5 +
Zmdmrae GU female 727 + 3 5 5 +
Eealawtynw FQ female 1176 - 4 3 2 -
Xcckkdlpe DG female 2956 - 5 4 3 +
Vwsoucv ZA male 1268 - 4 4 2 +
Mkxocxnm GO male 1345 - 5 3 5 -
Gvrwqtbuwg GQ female 234 - 5 5 4 -
Ajcqt AC female 997 + 3 3 4 +
Rvyjyxma OH female 1317 - 5 3 2 +
Biyncrqr UC female 1023 + 3 4 2 -
Hxlyou AK female 1961 - 2 3 5 +

test2.txt

Xdsnmhdr CX female 152 + 5 3 5 +
Fysji SW female 304 - 2 2 5 -
Qxapqmgzys VM female 1528 - 5 3 3 -
Mruaffg ZF male 1700 + 5 4 2 +
Wdulkvb ZU male 2918 - 3 2 2 +
Yzkpzzo MX female 940 - 2 3 2 -
Nexbkmylh JW female 2269 - 2 4 3 +
Oakpuhfv MR male 1666 + 3 3 3 +
Fmxmp IW female 167 + 4 3 5 -
Oxiklzmwxv HZ female 1738 - 4 5 5 +
Wvvelk RR male 1961 + 5 2 5 -
Vnjzmibw XC male 1426 - 2 5 4 + 10
Pyumtgj BG male 2460 + 4 3 2 -
Orlxzhvsb IR female 1714 + 3 3 2 +
Sanbbj EI female 1264 - 4 4 5 -

test3.txt

Ccsrfabtp VW female 2883 - 3 5 5 -
Gbvlcxswjc UH female 2244 - 4 5 3 -
Jihbi EM female 491 - 3 5 5 +
Zuhcus TT female 205 - 3 4 3 -
Fzxxises IO female 1131 - 3 5 5 +
Twaiprv FA female 1902 + 3 2 5 +
Xfoddd XK female 2774 + 3 4 5 -
Ogbfvazxbt RP male 367 - 2 3 4 +
Heyzlxucf FZ male 1207 - 3 2 5 +

Efqwwgymic FU female 2407 - 2 4 2 -
Qargpw XT female 253 - 2 4 5 +
Uomdwdstcw ZW male 1230 + 3 4 4 +
Htokswaxqs HN female 1231 + 3 3 5 +
Qiqftkqg FZ female 950 - 4 5 4 -
Ihsjo UZ male 1371 - 4 5 2 -

Протокол

(py37) → ~ gcc main6kp.c -o 6kp

(py37) → ~ ./6kp test.txt

What's a p?

>>> 3

|Surname |In|Sex|School|Medal|Math|Phys|Lit|Writing|

| Kwhhgb|EK|female| 2355| +| 4| 4| 2| -|
| Biyncrqr|UC|female| 1023| +| 3| 4| 2| -|

(py37) → ~ ./6kp test.txt

What's a p?

>>> 4

|Surname |In|Sex|School|Medal|Math|Phys|Lit|Writing|

Mblpqvkc	WA	female	855	+	4	3	3	+
Kwhhgb	EK	female	2355	+	4	4	2	-
Zmdmrae	GU	female	727	+	3	5	5	+
Ajcqt	AC	female	997	+	3	3	4	+
Biyncrqr	UC	female	1023	+	3	4	2	-

(py37) → ~ ./6kp test2.txt

What's a p?

>>> 3

|Surname |In|Sex|School|Medal|Math|Phys|Lit|Writing|

Mruaffg	ZF	male	1700	+	5	4	2	+
Wvvelk	RR	male	1961	+	5	2	5	-
BG	ma	le	2460	+	4	3	2	-
Orlxzhvsb	IR	female	1714	+	3	3	2	+

(py37) → ~ ./6kp test2.txt

What's a p?

>>> 2

|Surname |In|Sex|School|Medal|Math|Phys|Lit|Writing|

| NO DATA |

(py37) → ~ ./6kp test3.txt

What's a p?

>>> 3

```
-----  
|Surname |In|Sex|School|Medal|Math|Phys|Lit|Writing|  
-----
```

```
|   Twaiprv|FA|female| 1902| +|  3|  2| 5|    +|  
-----
```

```
(py37) → ~ ./6kp test3.txt
```

```
What's a p?
```

```
>>> 4  
-----
```

```
|Surname |In|Sex|School|Medal|Math|Phys|Lit|Writing|  
-----
```

```
|   Twaiprv|FA|female| 1902| +|  3|  2| 5|    +|
```

```
|   Xfoddd|XK|female| 2774| +|  3|  4| 5|    -|
```

```
| Uomdwdstcw|ZW| male| 1230| +|  3|  4| 4|    +|
```

```
| Htokswaxqs|HN|female| 1231| +|  3|  3| 5|    +|  
-----
```

```
(py37) → ~ cat main6kp.c
```

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

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

```
typedef struct{  
    char surname[30+1];  
    char initials[2+1];  
    char sex[6+1];  
    int schoolNumber;  
    char haveMedal;  
    int math, phys, lit;  
    char passedWriting;  
} Student;
```

```
int main(int argc, char* argv[]){  
    char *filename;  
    if(argc < 2){  
        filename = "test.txt";  
    } else{  
        filename = argv[1];  
    }  
    const int count = 15;  
    FILE *file = fopen(filename, "r");  
    Student *students = (Student*)calloc(count, sizeof(Student));  
    Student *result = (Student*)calloc(count, sizeof(Student));  
    int i = 0, ires = 0;  
  
    while(!feof(file)){  
        Student s;
```

```

        fscanf(file, "%s %2s %s %d %c %d %d %d %c", s.surname, s.initials,
s.sex, &s.schoolNumber, &s.haveMedal, &s.math, &s.phys, &s.lit,
&s.passedWriting);
        students[i++] = s;
    }

    int p;
    printf("What's a p?\n>>> ");
    scanf("%d", &p);
    puts("-----");
    puts("|Surname |In|Sex|School|Medal|Math|Phys|Lit|Writing|");
    puts("-----");
    for (int j = 0; j < count; ++j){
        Student s = students[j];
        if(s.haveMedal == '+' && (s.math < p || s.phys < p || s.lit < p)){
            printf("|%12s|%2s|%6s|%6d|%2c|%4d|%4d|%3d|%7c|\n",
s.surname, s.initials, s.sex, s.schoolNumber, s.haveMedal, s.math, s.phys, s.lit,
s.passedWriting);
            result[ires++] = s;
        }
    }
    if(ires == 0){
        puts("                NO DATA                |");
    }
    puts("-----");
    fclose(file);
}%
(py37) → ~

```

Заключение

Я научился работать с базами данных в виде текстовых файлах, научился делать простейшие запросы к этой базе и применять к ней фильтры по значениям, в частности, я реализовал запросы, в которых нужно выбрать студента, у которого есть медаль и оценка по какому-либо из предметов ниже р.