Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Курсовая работа**

**по курсу «Языки и методы программирования»**

**ll Семестр**

**Задание 8**

**Обработка последовательной файловой структуры на языке Си**

|  |  |
| --- | --- |
| Студент: | Жерлыгин М.А. |
| Группа: | М8О-108Б-18 |
| Преподаватель: | Поповкин А.В. |
|  |  |
| Оценка: |  |
| Дата: |  |

**Москва, 2018 г.**

**Содержание**

1. Постановка задачи………………………………………………….3
2. Общий метод решения……………………………………………..3
3. Общие сведения о программе……………………………………...4
4. Описание логической структуры………………………………….4
5. Таблица переменных и констант…………………………………..5
6. Программа на Си…………………………………………………...5
7. Демонстрация работы программы………………………………...13
8. Вывод ……………………………………………………………….17
9. Список используемых источников ……..………………………...17

**Постановка задачи:**

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). Выполнить над файлом заданные действия: проверить суммы баллов абитуриентов на попадание в границы введённых параметров *P1* и *P2* и вывести их имена*.*

Действие по выборке данных из файла оформить в виде отдельной программы с параметрами запроса, вводимыми из стандартного входного тестового файла, или получаемыми из командной строки UNIX. Параметры задаются с помощью ключей -f (печать) и -p (параметры конкретного варианта задания).

Вариант задания: №8

Содержимое и структуры файла:

Сведения о составе комплектующих личных ПЭВМ в студенческой группе: фамилия владельца, число и тип процессоро, объём памяти, тип видеоконтроллера и объём видеопамяти, тип число и ёмкость винчестеров, количество интегрированных контроллеров и внешних устройств, операционная система.

**Действиe:**

Составить список мультимедийных систем и бездисковых компьютеров.

**Общий метод решения:**

Работа идет с двумя прогрммами СИ, одной библиотекой (содержит описание структуры), текстовым файллм (хранятся данные) и бинарным файлом (туда записываются данные из текстовых файлов).

Первая программа читает данные из текстового файла и записывает их в бинарный.

Вторая программа производит печать данных из бинарного файла в виде таблицы (если указан ключ -f), или результат выполнения поставленной вариантом задачи (ключ -p).

**Общие сведения о программе:**

Программное и аппаратное обеспечения для запуска данной программы на ПК не ограничено в выборе. Операционная система семейства Unix — Ubuntu (upsystem for Windows). Язык программирования Си.

Имена файлов:

1) computers.h;

2) selection.cpp;

3) makebinfile.cpp;

4) computers.bin;

**Описание логической структуры:**

makebinfile.cpp – программа, читающая файл computers.h, объявлённый в коде, и записывающий по нему бинарный файл computers.dat с распечаткой получившейся таблицы данных.

selection.cpp – программа, которая по ключу -f производит вывод таблицы данных в терминал. По ключу -p производит анализ таблицы и вывод мультимедийных систем и бездисковых компьютеров

**Таблица переменных и констант:**

|  |  |
| --- | --- |
| char name[32]; | Фамилия |
| int cpucount; | Количество процессоров |
| char cputype[10]; | Тип процессора |
| Int memvol; | Количество оперативной памяти |
| char videotype[10]; | Тип видеокарты |
| int videomemvol; | Количество памяти видеокарты |
| char hddtype[5]; | Вид винчестера |
| int hddcount; | Количество винчестеров |
| int hddvol; | Количество памяти винчестера |
| int contrcount; | Количество контроллеров |
| int perifcount; | Количество переферийных устройств |
| char os[10] | Операционная система |

Таблица 1. Описание переменных.

В программах selection.cpp и makebinfile.cpp происходит работа со структурой computers.h, значительно новых переменных не добавилось.

**Программа на Си:**

**computers.h:**

#define COMPUTERS \

"Обыденкова", 1, "i7", 16, "PCI-E", 16, \

"ATA", 1, 1, 5, 3, \

"Windows",\

"Овечкин", 1, "i7", 24, "ISA", 4,\

"ATA", 2, 4, 4, 3, \

"Linux", \

"Пупкин", 2, "i3", 2, "PCI-E", 8,\

"SATA", 3, 2, 4, 4, \

"Linux", \

"Каснер", 1, "i5", 32, "PCI", 4, \

"SATA", 0, 0, 3, 3, "Win\

dows",\

"Рыженко", 1, "i5", 8, "AGP", 8,\

"ATA", 1, 4, 3, 4, \

"Linux", \

"Меладзе", 2, "i3", 8, "PCI-E", 16\

, "IDE", 1, 1, 2, 4, \

"Linux", \

"Дудочкин", 1, "i7", 16, "PCI-E", 4, \

"SATA", 1, 4, 3, 5, "Linux",\

\

"Братишкин", 2, "i3", 8, "AGP", 2, \

"IDE", 1, 2, 6, 4, "Lin\

ux", \

"Лолкин", 2, "i5", 32, "ISA", 8,\

"ATA", 2, 3, 4, 5, \

"Windows"

**selection.cpp:**

#include <stdio.h>

#include <stdlib.h>

#include <sys/stat.h>

#include <string.h>

#include "computers.h"

#include "person.h"

int load(char \* filename);

int query(char \* filename);

int main(int argc, char \*argv[])

{

char\* filename = (char\*)"computers.dat";

if (argc != 2) {

printf("Программа используется с ключами: -f распечатка файла, -p\

анализ\n");

exit(1);

}

if (argv[1][0] == '-' && argv[1][1] == 'f') { load(filename); exit(0); }

if (argv[1][0] == '-' && argv[1][1] == 'p') { query(filename); exit(0); } \

else {printf("Программа используется с ключами: -f распечатка файла, -p параметр\\

n"); exit(1);}

//load(filename);

//query(filename);

return 0;

}

// загрузка из файла структуры

int load(char \* filename)

{

FILE \* fp;

char \*c, i;

struct person \*ptr, \*startptr;

if ((fp = fopen(filename, "rb")) == NULL)

{

perror("Error occured while opening file");

return 1;

}

// узнаем размер файла и выделим память

struct stat statistics;

if (fstat(fileno(fp), &statistics) != -1) {

ptr = (struct person\*)malloc(statistics.st\_size);

startptr = ptr;

}

// устанавливаем указатель на начало блока выделенной памяти

c = (char \*)ptr;

// считываем посимвольно из файла

while ((i = (char)getc(fp)) != EOF)

{

\*c = i;

c++;

}

fclose(fp);

// вывод на консоль загруженной структуры

int sizestr = sizeof(struct person);

long int reccount = statistics.st\_size / sizestr;

printf("Фамилия \tчисло CPU Тип CPU Объем памяти Тип видео Объем видеопамя\

ти Тип HDD Число HDD Емкость HDD Количество контроллеров Периферия ОС\n\n");

while (reccount > 0)

{

printf("%-16s\t %3d \t %-10s %-3d %-10s %-3d \

%-5s %-3d %-3d %-3d %-3d %-10s\n",

ptr->name,

ptr→cpucount,

ptr->cputype,

ptr->memvol,

ptr->videotype,

ptr->videomemvol,

ptr->hddtype,

ptr->hddcount,

ptr->hddvol,

ptr->contrcount,

ptr->perifcount,

ptr->os

);

ptr++;

reccount--;

}

free(startptr);

return 0;

}

int query(char \* filename)

{

struct param1

{

char name[32];

int cpucount;

char cputype[10];

int memvol;

char videotype[10];

int videomemvol;

char hddtype[5];

int hddcount;

int hddvol;

int contrcount;

int perifcount;

char os[10];

};

// загрузка из файла структуры

FILE \* fp;

char \*c;

struct person \*ptr, \*startptr;

char i; // для считывания одного символа

if ((fp = fopen(filename, "rb")) == NULL)

{

perror("Error occured while opening file");

return 1;

}

// узнаем размер файла и выделим память

struct stat statistics;

if (fstat(fileno(fp), &statistics) != -1) {

ptr = (struct person\*)malloc(statistics.st\_size);

startptr = ptr;

}

// устанавливаем указатель на начало блока выделенной памяти

c = (char \*)ptr;

// считываем посимвольно из файла

while ((i = (char)getc(fp)) != EOF)

{

\*c = i;

c++;

}

// количество записей в файле

long int equality = 0, reccounter = statistics.st\_size / sizeof(struct pe\

rson);

fclose(fp);

printf("\nФамилия \tчисло CPU Тип CPU Объем памяти Тип видео Объем видеоп\

амяти Тип HDD Число HDD Емкость HDD Количество контроллеров Периферия ОС\n\n");

ptr = startptr;

while (reccounter > 0)

{

// много процессоров(>1) и мало памяти(<4) equality = 2

if (((ptr->cpucount) > 0) && ((ptr->memvol) > 0) && (ptr->videome\

mvol > 0) && (ptr->hddvol > 0)) { equality = 2; goto endchecking; }

// много процессоров(>1) и мало объема hdd(<2) equality = 3

if ((ptr->hddvol == 0) || (ptr->hddcount == 0)) { equality = 3; g\

oto endchecking; }

endchecking:

// Распечатаем, удовлетворяющую запросу запись

if (equality != 0)

{

printf("%-16s\t %3d \t %-10s %-3d %-10s %-3\

d %-5s %-3d %-3d %-3d %-3d %-1\

0s\n",

ptr->name,

ptr->cpucount,

ptr->cputype,

ptr->memvol,

ptr->videotype,

ptr->videomemvol,

ptr->hddtype,

ptr->hddcount,

ptr->hddvol,

ptr->contrcount,

ptr->perifcount,

ptr->os

);

switch (equality)

{

case 2:

printf("Мультимедийные компьютеры\n\n");

break;

case 3:

printf("Бездисковые рабочие станции\n\n");

break;

}

}

equality = 0;

ptr++;

reccounter--;

}

return 0;

}

**makebinfile.c:**

#include <stdio.h>

#include <stdlib.h>

#include <sys/stat.h>

#include <string.h>

#include "computers.h"

#include "person.h"

int save(char \* filename, struct person \*p, int size);

int load(char \* filename);

int main(void)

{

char \* filename = (char\*)"computers.dat";

// Массив комплектаций

struct person people[] = { COMPUTERS };

save(filename, people, sizeof(people));

load(filename);

return 0;

}

// запись данных в файл

int save(char \* filename, struct person \*p, int size)

{

FILE \* fp;

char \*c;

if ((fp = fopen(filename, "wb")) == NULL)

{

perror("Error occured while opening file");

return 1;

}

// устанавливаем указатель на начало структуры

c = (char \*)p;

// посимвольно записываем в файл структуру

for (int i = 0; i < size; i++)

{

putc(\*c++, fp);

}

fclose(fp);

return 0;

}

// загрузка из файла структуры

int load(char \* filename)

{

FILE \* fp;

char \*c;

struct person \*ptr, \*startptr;

char i; // для считывания одного символа

if ((fp = fopen(filename, "rb")) == NULL)

{

perror("Error occured while opening file");

return 1;

}

// узнаем размер файла и выделим память

struct stat statistics;

if (fstat(fileno(fp), &statistics) != -1) {

ptr = (struct person\*)malloc(statistics.st\_size);

startptr = ptr;

}

// устанавливаем указатель на начало блока выделенной памяти

c = (char \*)ptr;

// считываем посимвольно из файла

while ((i = (char)getc(fp)) != EOF)

{

\*c = i;

c++;

}

}

fclose(fp);

printf("Данная программа формирует нетекстовый файл computers.dat, содерж\

ащий следующую информацию:\n\n");

// вывод на консоль загруженной структуры

int sizestr = sizeof(struct person);

long int reccount = statistics.st\_size / sizestr;

printf("Фамилия \tчисло CPU Тип CPU Объем памяти Тип видео Объем видеопам\

яти Тип HDD Число HDD Емкость HDD Количество контроллеров Периферия ОС\n\n");

while (reccount > 0)

{

printf("%-16s\t %3d \t %-10s %-3d %-10s %-3d \

%-5s %-3d %-3d %-3d %-3d %-10s\n",

ptr->name,

ptr->cpucount,

ptr->cputype,

ptr→memvol,

ptr->videotype,

ptr->videomemvol,

ptr->hddtype,

ptr->hddcount,

ptr->hddvol,

ptr->contrcount,

ptr->perifcount,

ptr->os

);

ptr++;

reccount--;

}

free(startptr);

return 0;

}

**Демонстрация работы программы:**

mmaxim2710@DESKTOP-RDPBU3D:~/2sem/kp6$ ls

'#computers.h#' computers.dat computers.h lol lol1 makebinfile.cpp person.h selection.cpp

mmaxim2710@DESKTOP-RDPBU3D:~/2sem/kp6$ ./lol1

Данная программа формирует нетекстовый файл computers.dat, содержащий следующую информацию:

Фамилия число CPU Тип CPU Объем памяти Тип видео Объем видеопамяти Тип HDD Число HDD Емкость HDD Количество контроллеров Периферия ОС

Иванов 1 i3 16 PCI-E 16 SATA 1 1 3 3 Linux

Петров 1 i7 24 ISA 4 IDE 2 4 4 4 Linux

Васечкин 2 i3 2 PCI-E 8 ATA 3 2 5 3 Linux

Мартынова 1 i3 16 PCI 4 SATA 4 4 4 3 Linux

Кренев 1 i3 16 AGP 4 ATA 1 4 3 3 Linux

Кирвалидзе 2 i3 8 PCI-E 16 IDE 1 1 4 4 Linux

Загайнова 1 i7 16 PCI-E 4 SATA 1 4 2 3 Linux

Абдурахманов 2 i3 8 AGP 2 IDE 1 3 2 5 Linux

Сидоров 1 i5 32 ISA 4 SATA 2 4 4 4 Windows

mmaxim2710@DESKTOP-RDPBU3D:~/2sem/kp6$ ./lol

Программа используется с ключами: -f распечатка файла, -p анализ

mmaxim2710@DESKTOP-RDPBU3D:~/2sem/kp6$ ./lol -p

Фамилия число CPU Тип CPU Объем памяти Тип видео Объем видеопамяти Тип HDD Число HDD Емкость HDD Количество контроллеров Периферия ОС

Иванов 1 i3 16 PCI-E 16 SATA 1 1 3 3 Linux

Мультимедийные компьютеры

Петров 1 i7 24 ISA 4 IDE 2 4 4 4 Linux

Мультимедийные компьютеры

Васечкин 2 i3 2 PCI-E 8 ATA 3 2 5 3 Linux

Мультимедийные компьютеры

Мартынова 1 i3 16 PCI 4 SATA 4 4 4 3 Linux

Мультимедийные компьютеры

Кренев 1 i3 16 AGP 4 ATA 1 4 3 3 Linux

Мультимедийные компьютеры

Кирвалидзе 2 i3 8 PCI-E 16 IDE 1 1 4 4 Linux

Мультимедийные компьютеры

Загайнова 1 i7 16 PCI-E 4 SATA 1 4 2 3 Linux

Мультимедийные компьютеры

Абдурахманов 2 i3 8 AGP 2 IDE 1 3 2 5 Linux

Мультимедийные компьютеры

Сидоров 1 i5 32 ISA 4 SATA 2 4 4 4 Windows

Мультимедийные компьютеры

mmaxim2710@DESKTOP-RDPBU3D:~/2sem/kp6$ ./lol -f

Фамилия число CPU Тип CPU Объем памяти Тип видео Объем видеопамяти Тип HDD Число HDD Емкость HDD Количество контроллеров Периферия ОС

Иванов 1 i3 16 PCI-E 16 SATA 1 1 3 3 Linux

Петров 1 i7 24 ISA 4 IDE 2 4 4 4 Linux

Васечкин 2 i3 2 PCI-E 8 ATA 3 2 5 3 Linux

Мартынова 1 i3 16 PCI 4 SATA 4 4 4 3 Linux

Кренев 1 i3 16 AGP 4 ATA 1 4 3 3 Linux

Кирвалидзе 2 i3 8 PCI-E 16 IDE 1 1 4 4 Linux

Загайнова 1 i7 16 PCI-E 4 SATA 1 4 2 3 Linux

Абдурахманов 2 i3 8 AGP 2 IDE 1 3 2 5 Linux

Сидоров 1 i5 32 ISA 4 SATA 2 4 4 4 Windows

mmaxim2710@DESKTOP-RDPBU3D:~/2sem/kp6$

**Заключение**

Базы данных могут содержать различные объекты. Основными объектами любой базы данных являются ее таблицы. Простейшая база данных имеет хотя бы одну таблицу. Соответственно, структура простейшей базы данных тождественно равна структуре ее таблицы. Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в простейшей базе данных являются поля и записи. Если записей в таблице пока нет, значит, ее структура образована только набором полей. Изменив состав полей базовой таблицы (или их свойства), мы изменяем структур базы данных и, соответственно, получаем новую базу данных.  
Данная курсовая работа научила меня работе с базами данных, их анализу, записи данных в бинарный файл.

**Список используемых источников**

1) Методические указания к выполнению курсовых работ. Зайцев В. Е.

2)Работа с текстовыми файлами [Электронный ресурс]. – URL: https://learnc.info/c/text\_files.html

3)Работа с файлами в СИ[Электронный ресурс]. – URL: http://www.sbp-program.ru/c/sbp-file-c.htm

4) Списки о зачислении [Электронный ресурс]. – URL: https://mai.ru/upload/iblock/192/303-Prikaz-o-zachislenii-MAI-ochnoe-03.08.2018-\_303st.pdf