**Московский Авиационный Институт**

(Национальный Исследовательский Университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Курсовой проект

по дисциплине «Практикум программирования»

2 семестр

Задание 6.

Тема: «Обработка последовательной файловой структуры на языке Си»

|  |  |
| --- | --- |
| Студент: | Доянова В. А. |
| Группа: | М8О-111Б-22 |
| Преподаватель: | Аносова Н. П. |
| Подпись: |  |
| Оценка: |  |

Москва, 2023

Содержание:

**Задание3**

Формулировка3

**Работа4**

Код программы4

Описание алгоритма7

Протокол выполнения программы8

**Вывод9**

**Задание**

**Формулировка**

Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си. Составить программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей.

Распечатать содержимое сгенерированного файла в виде таблицы и выполнить над ним заданное действие для 3–5 значений параметров запроса ***p*** и распечатать результат.

Структуры данных и константы, совместно используемые программами, следует вынести в отдельный заголовочный файл.

Содержимое и структура файла:Сведения о составе комплектующих личных ПЭВМ в студенческой группе: фамилия владельца,число и тип процессоров, объём памяти, тип видеоконтроллера (встроенный, внешний, видео-шина) иобъём видеопамяти, тип (SAS/SATA), число и ёмкость винчестеров, количество интегрированныхконтроллеров и внешних (периферийных) устройств, операционная система.

Действие: перечислить все компьютеры студентов группы, платформа которых отлична от WINTEL.

**Работа**

**Код программы**

Файл comp.h: #pragma once

#ifndef compbase

#define compbase

#include <string>

using namespace std;

enum processor {

INTEL, AMD, ARM

};

enum videocontrol {

INTERNAL, EXTERNAL, VIDEOBUS

};

enum disk {

SAS, SATA

};

struct comp {

char name[25];

int processornum;

string processor\_str;

processor processor;

int memory;

string videocontrol\_str;

videocontrol videocontrol;

int videomemory;

string disk\_str;

disk disk;

int disknum;

int diskcap;

int integcontrol;

int exterdev;

char os[25];

};

processor get\_processor\_from\_string(const string& str);

videocontrol get\_videocontrol\_from\_string(const string& str);

disk get\_disk\_from\_string(const string& str);

void read\_comp\_data(ifstream& input, comp& c);

void write\_comp\_data(ofstream& output, const comp& c);

void print\_comp\_data(const comp& c);

#endif

Файл comp.cpp: #include "comp.h"

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

void read\_comp\_data(ifstream& input, comp& c) {

input >> c.name >> c.processornum >> c.processor\_str >> c.memory >> c.videocontrol\_str

>> c.videomemory >> c.disk\_str >> c.disknum >> c.diskcap >> c.integcontrol >> c.exterdev >> c.os;

c.processor = get\_processor\_from\_string(c.processor\_str);

c.videocontrol = get\_videocontrol\_from\_string(c.videocontrol\_str);

c.disk = get\_disk\_from\_string(c.disk\_str);

}

void write\_comp\_data( ofstream& output, const comp& c) {

output.write(reinterpret\_cast<const char\*>(&c), sizeof(comp));

}

void print\_comp\_data(const comp& c) {

if (string(c.os) == "WINTEL") {

return;

}

cout << left << "| Фамилия " << "| Число процессоров " << " | Процессор\t" << "| Память " << "| Видеоконтроллер " << "| Видеопамять " << "| Тип " << "| Число винчестеров " << "| Ёмкость " << "| Интегрированные контроллеры " << "| Внешние устройства " << "| ОС\t |" << endl;

cout << "|----------|--------------------|---------------|---------|------------------|--------------|-------|--------------------|----------|------------------------------|---------------------|-------|\n";

cout << left << "| " << c.name << " | " << c.processornum << " | " << c.processor\_str << "\t | " << c.memory

<< " | " << c.videocontrol\_str << " | " << c.videomemory << " | " << c.disk\_str

<< " | " << c.disknum << " | " << c.diskcap << " | " << c.integcontrol

<< " | " << c.exterdev << " | " << c.os << "|\n";

cout << "|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_|\n";

}

processor get\_processor\_from\_string(const string& str) {

if (str == "INTEL") {

return processor::INTEL;

}

else if (str == "AMD") {

return processor::AMD;

}

else {

return processor::ARM;

}

}

videocontrol get\_videocontrol\_from\_string(const string& str) {

if (str == "INTERNAL") {

return videocontrol::INTERNAL;

}

else if (str == "EXTERNAL") {

return videocontrol::EXTERNAL;

}

else {

return videocontrol::VIDEOBUS;

}

}

disk get\_disk\_from\_string(const string& str) {

if (str == "SAS") {

return disk::SAS;

}

else {

return disk::SATA;

}

}

Файл binary.cpp: #include "comp.h"

#include <iostream>

#include <fstream>

int main() {

setlocale(LC\_ALL, "Russian");

ifstream input("comp\_inf.txt");

ofstream output("comp\_data.bin", ios::binary);

comp c;

while (input) {

read\_comp\_data(input, c);

write\_comp\_data(output, c);

}

input.close();

output.close();

std::ifstream bin\_input("comp\_data.bin", ios::binary);

while (bin\_input.read(reinterpret\_cast<char\*>(&c), sizeof(comp))) {

print\_comp\_data(c);

cout << endl;

}

bin\_input.close();

return 0;

}

**Описание алгоритма**

Файл comp.h: я создаю классы с перечислением возможных процессоров, типов видеоконтроллеров, дисков. Я создаю структуру comp, в которой перечисляю то, чем будет заполнена таблица, вместе с типами данных и размерами в некоторых случаях. Также я определяю функции, которые буду использовать.

Файл comp.cpp: это файл с функциями.

* read\_comp\_data: эта функция считывает данные из текстового файла для последующей записи в бинарный файл.
* get\_processor\_from\_string, get\_videocontrol\_from\_string, get\_disk\_from\_string: эти функции выбирают нужный процессор, видеоконтроллер и диск, соответственно, из списка. Они нужны для правильной записи строковых значений в бинарный файл.
* write\_comp\_data: функция для записи данных в бинарный файл.
* print\_comp\_data: функция для вывода данных из бинарного файла на экран. Сначала она проверяет, является ли операционная система Wintel или нет. Если нет, то она выводит соответствующие строки. Иначе программа завершается.

Файл binary.cpp: это файл с частью main.

**Протокол выполнения программы**

1. Входные данные:

IVANOV 1 INTEL 300 INTERNAL 400 SATA 2 30 1 2 WINTEL

DADADAD 2 ARM 100 VIDEOBUS 250 SATA 2 61 1 5 WINDOWS

PETROV 3 AMD 250 EXTERNAL 670 SATA 4 75 2 3 WINTEL

NONONO 2 INTEL 300 VIDEOBUS 230 SATA 5 345 1 2 MACOS

Вывод программы:

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

1. Входные данные:

IVANOV 1 INTEL 300 INTERNAL 400 SATA 2 30 1 2 WINTEL

LAPIDUS 2 ARM 100 VIDEOBUS 250 SATA 2 61 1 5 WINTEL

PETROV 3 AMD 250 EXTERNAL 670 SATA 4 75 2 3 WINTEL

DOYANOVA 2 INTEL 300 VIDEOBUS 230 SATA 5 34 1 2 MACOS

Вывод программы:

Изображение выглядит как снимок экрана, программное обеспечение, текст, Мультимедийное программное обеспечение

Автоматически созданное описание

**Вывод**

Во время работы над данным курсовым проектом я написала программу для перевода текстового файла в бинарный файл и последующего вывода этих данных из файла. Таким образом, я разобралась с работой с файлами, поняла, как осуществлять ввод и вывод данных, учитывая необходимое действие.