**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 11

по дисциплине «Операционные системы»

на тему: «Управление файловой системой»

Выполнил: студент гр. ИТП-11

Иванов Д.А.

Принял: преподаватель

Карась О.В.

Гомель 2022

**Цель:** изучить принципы работы файлового менеджера и утилиты для работы с ним.

**Задача:**

Разработать утилиту в соответствии со своим вариантом.

**Вариант 11:**

Управление доступом.

**Задание**

Для реализации этой утилиты был написана программа на языке программирования *C*++. При первом запуске программы пользователя просят ввести логин и пароль для доступа к файлам (рис.1). Далее пользователя просят ввести количество нужной памяти для файлов (рис.2). Затем открывается меню с основными функциями программы такие как: добавить файл, показать файлы, удалить файл и выход из программы (рис.3).

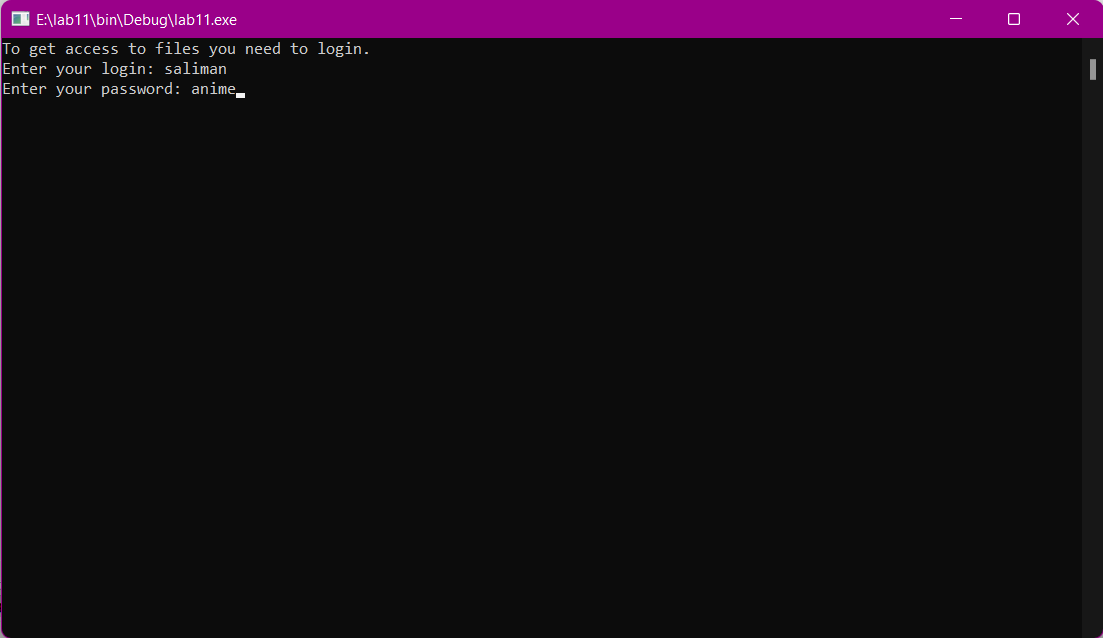


Рисунок 1 – Доступ к файлам

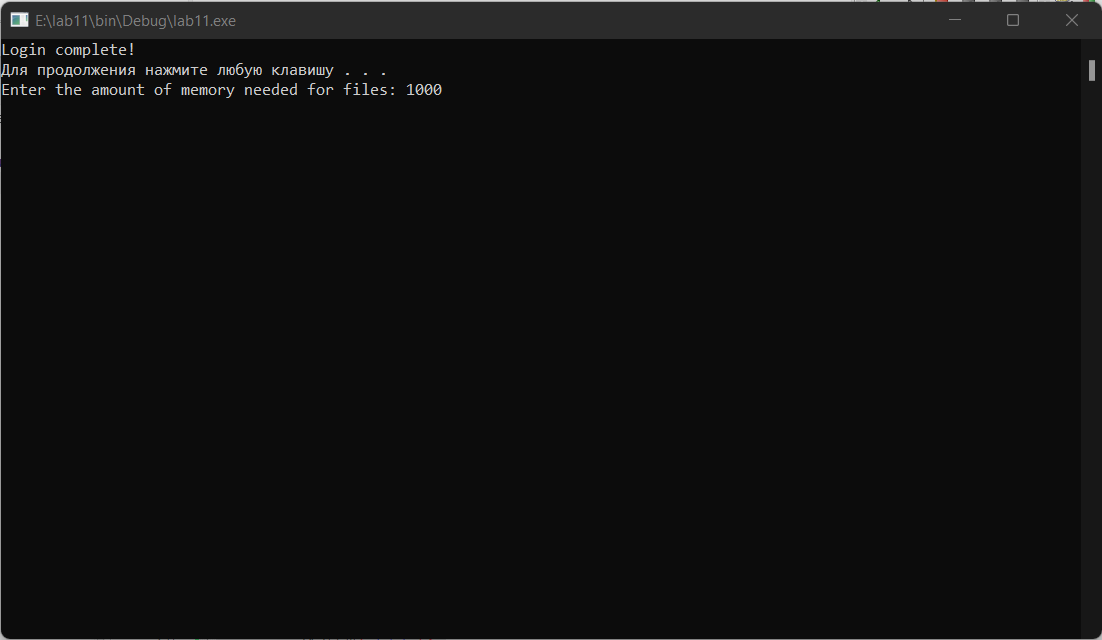


Рисунок 2 – Ввод объёма памяти

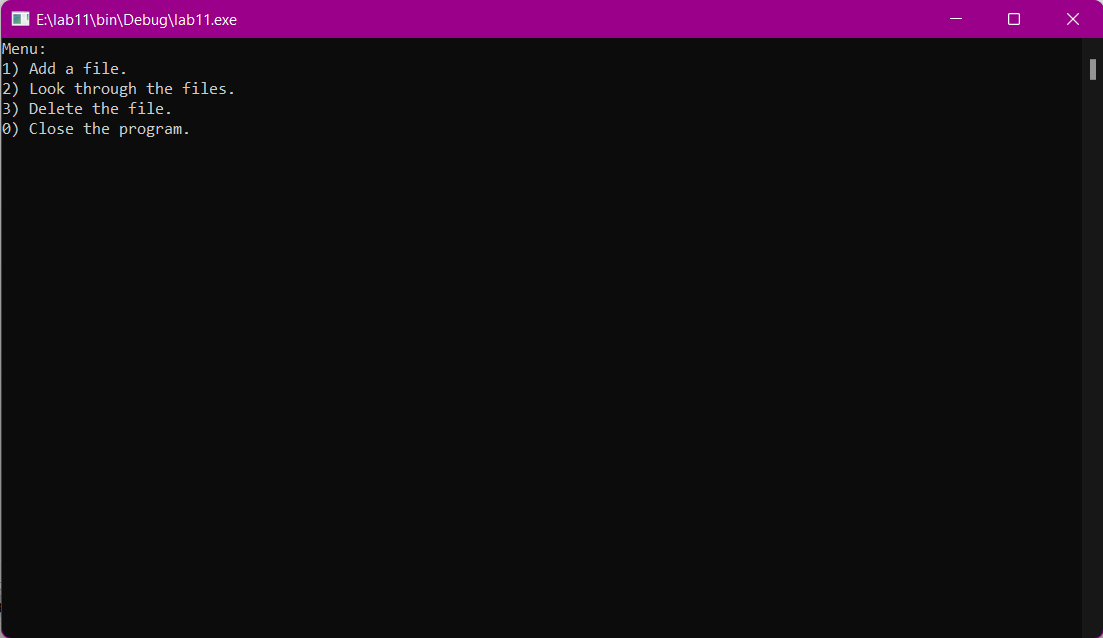


Рисунок 3 – Главное меню

После этого пользователь может создать файл. В новом меню предлагается вести название файла и объем памяти, который этот файл будет занимать на диске (рис.4). Если объем памяти будет меньше, чем количество свободного места на диске то файл не будет добавлен. Если будет предложен файл с уже существующим именем, то программа даст выбор: заменить файл с таким именем на новый (рис.5) или создать файл с таким же именем, но с хештегом “#” после (рис.6).

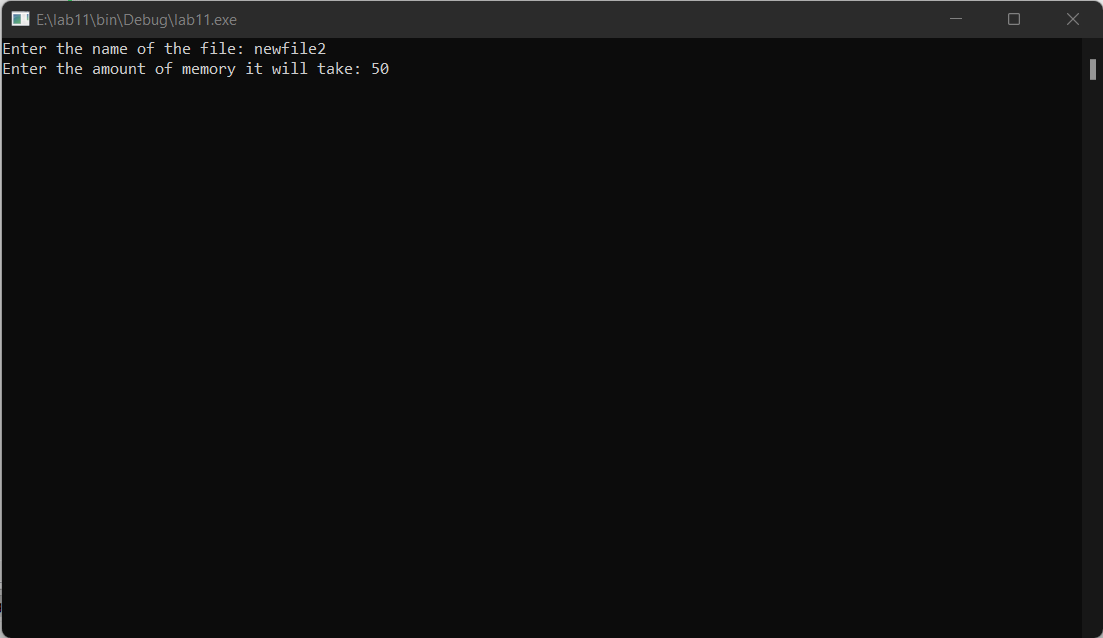


Рисунок 4 – Добавление нового файла

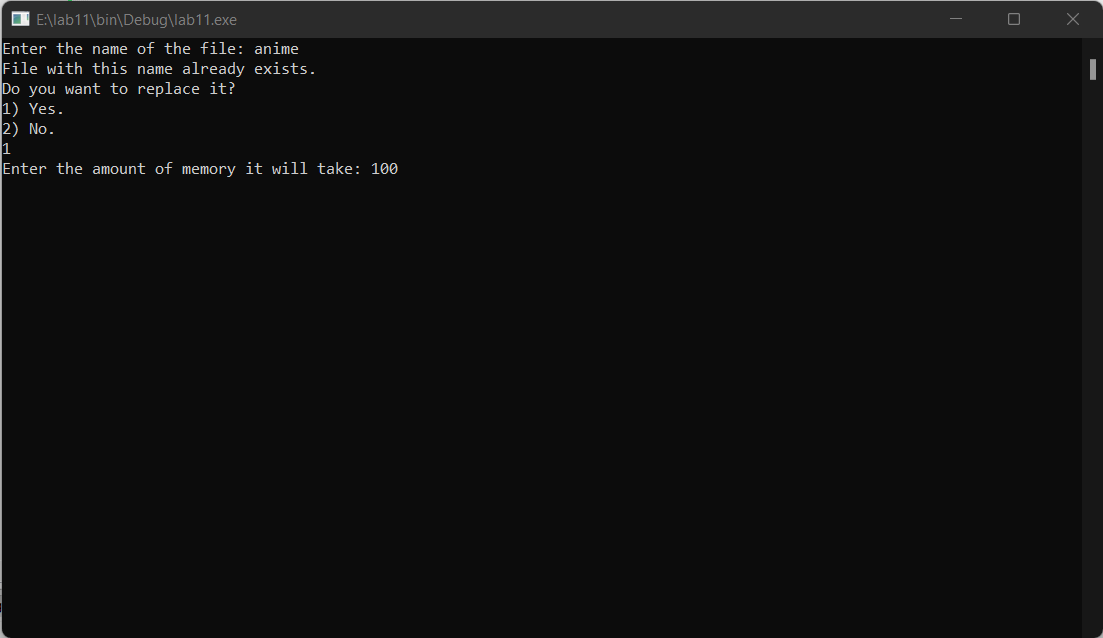


Рисунок 5 – Замена файла

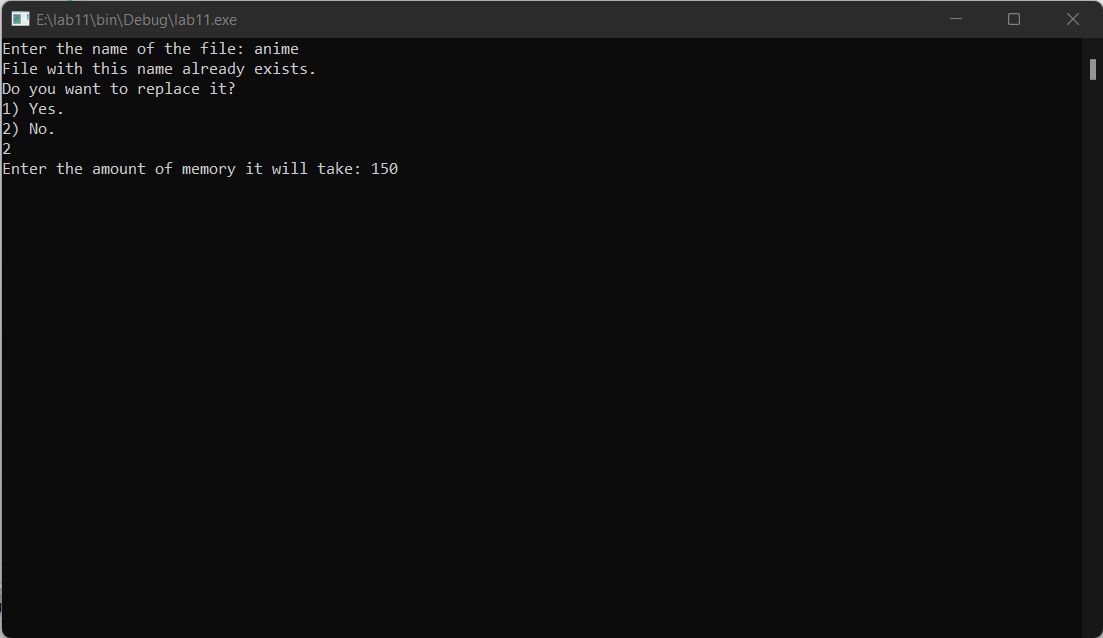


Рисунок 6 – Создание нового файла с хештегом

После добавления нескольких файлов пользователь может открыть список файлов, где указаны названия файлов и их объем. А также количество свободной памяти (рис.7).

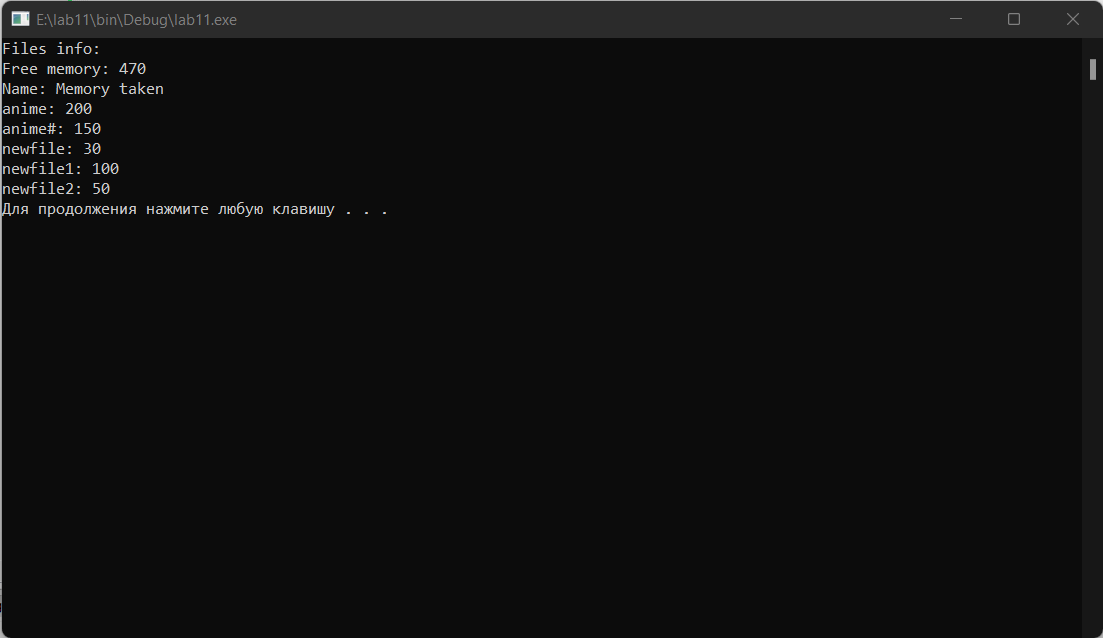


Рисунок 7 – Список файлов

Один из пунктов меню это удаление файла. При выборе этого пункта будет открываться новое окно куда пользователь указывает название файла, который нужно удалить (рис.8). Если файла с таким названием нет, то выведет ошибку.

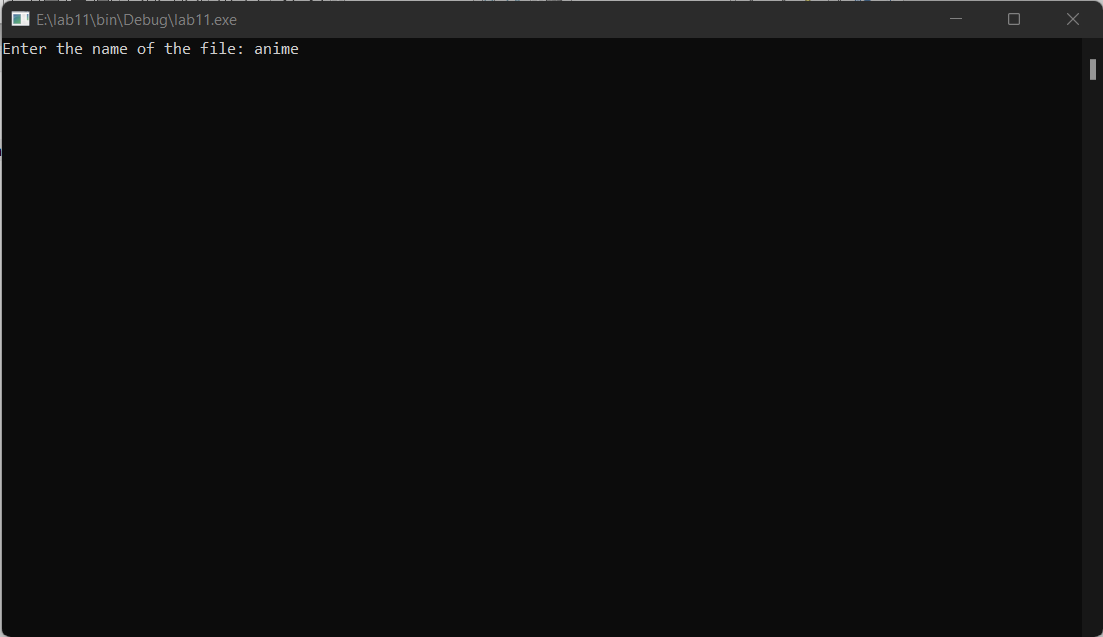


Рисунок 8 – Удаление файла

После удаления файла список файлов примет такой вид (рис.9).

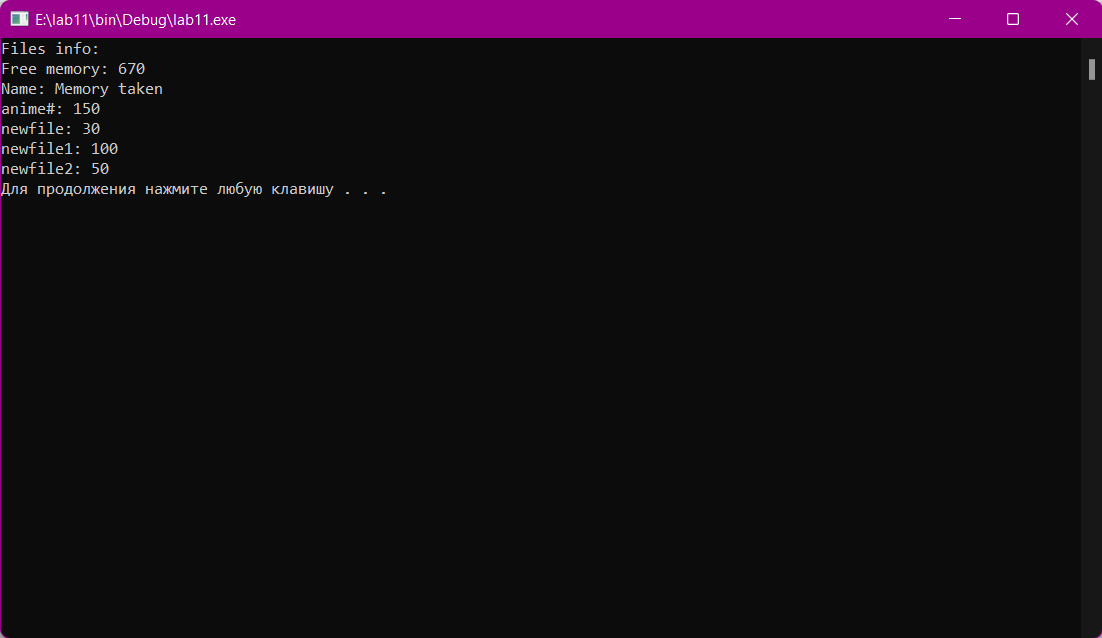


Рисунок 9 – Результат удаления файла

**Вывод**: в ходе лабораторной работы были изучены основные утилиты для работы с файловым менеджером и разработана программа с управлением доступа на языке программирования *C*++.

**Листинг программы**

#include <bits/stdc++.h>

using namespace std;

map<string, int> files;

string login = "saliman", password = "anime";

int mem, mem\_took = 0;

int main()

{

string temp = "";

while (1) {

cout << "To get access to files you need to login." << endl << "Enter your login: ";

cin >> temp;

if (login == temp) {

cout << "Enter your password: ";

cin >> temp;

if (password == temp) {

system("cls");

cout << "Login complete!" << endl;

system("pause");

break;

}

system("cls");

cout << "Incorrect password" << endl;

}

system("cls");

cout << "Incorrect login" << endl;

}

cout << "Enter the amount of memory needed for files: ";

cin >> mem;

int p = -1;

while (p != 0) {

system("cls");

cout << "Menu:" << endl << "1) Add a file." << endl << "2) Look through the files." << endl << "3) Delete the file." << endl << "0) Close the program." << endl;

cin >> p;

if (p >= 1 && p <= 3) {

string name;

int mem\_temp;

switch (p) {

case 1:

{

system("cls");

cout << "Enter the name of the file: ";

cin.ignore();

getline(cin, name);

if (files.find(name) != files.end()) {

cout << "File with this name already exists." << endl << "Do you want to replace it?" << endl << "1) Yes." << endl << "2) No." << endl;

int choise;

cin >> choise;

cout << "Enter the amount of memory it will take: ";

cin >> mem\_temp;

if (choise == 1) {

if ((mem\_took - files[name] + mem\_temp) <= mem) {

mem\_took -= files[name];

files[name] = mem\_temp;

mem\_took += mem\_temp;

}

else {

system("cls");

cout << "Out of memory";

system("pause");

}

}

else {

if (mem\_took + mem\_temp <= mem) {

while (files.find(name) != files.end()) name += "#";

files[name] = mem\_temp;

mem\_took += mem\_temp;

}

else {

system("cls");

cout << "Out of memory";

system("pause");

}

}

}

else {

cout << "Enter the amount of memory it will take: ";

cin >> mem\_temp;

if (mem\_took + mem\_temp <= mem) {

files[name] = mem\_temp;

mem\_took += mem\_temp;

}

else {

system("cls");

cout << "Out of memory";

system("pause");

}

}

break;

}

case 2:

{

system("cls");

cout << "Files info:" << endl << "Free memory: " << mem - mem\_took << endl << "Name: Memory taken" << endl;

for (auto& i : files) cout << i.first << ": " << i.second << endl;

system("pause");

break;

}

case 3:

{

system("cls");

cout << "Enter the name of the file: ";

cin.ignore();

getline(cin, name);

if (files.find(name) != files.end()) files.erase(name);

else {

cout << "There are no files with this name." << endl;

system("pause");

}

break;

}

}

}

}

}