|  |
| --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| федеральное государственное автономное образовательное учреждение  высшего профессионального образования |
| **«Национальный исследовательский ядерный университет «МИФИ»** |
| ФАКУЛЬТЕТ КИБЕРНЕТИКИ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ |
| КАФЕДРА УПРАВЛЯЮЩИХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ (№ 29) |

|  |
| --- |
| Заказчик |
| М.Н. Петухов |
| 2014 года |

Модель файловой системы ОС реального времени

Модуль записи метки тома и имени владельца

Текст описания программы

ГОСТ 19.402-78

ГОСТ 19.401-78

Исполнитель Богатенков А.

Кириченко А.

Принял И.В. Алексеенко

Москва 2014

Оглавление

[1. Аннотация 3](#_Toc385282852)

[2. Общие сведения 3](#_Toc385282853)

[2.1. Функциональное назначение 3](#_Toc385282854)

[2.2. Вызов и загрузка 3](#_Toc385282855)

2.3 Входные и выходные параметры…………………………………………...4

[3. Алгоритм программы 3](#_Toc385282856)

[3.1. Используемые классы и методы 4](#_Toc385282857)

[4. Модульное тестирование 5](#_Toc385282858)

[5. Исходный код модуля 7](#_Toc385282859)

# Аннотация

В данном документе представлен текст программного модуля «модуль записи метки тома и владельца для модели файловой системы ОС реального времени, разрабатываемой в рамках курса Технологии Программирования» входящего в состав разрабатываемой модели «Демонстрационная модель файловой системы ОС реального времени». Помимо исходного кода программы документ содержит описание класса файловой системы и его методов.

# Общие сведения

## 

## Функциональное назначение

Программа «Модель файловой системы ОС реального времени» предназначена для изучения устройства и функционирования файловых систем.

Модуль записи метки тома и владельца – модуль программы «Модель файловой системы ОС реального времени». Он предоставляет пользователю возможность записи метки тома и владельца и проверяет правильность вводимых параметров.

# Вызов и загрузка

Модуль записи метки тома и владельца вызывается монитором команд, если пользователю необходимо изменить метку тома или имя владельца.

**2.3 Входные и выходные параметры**

|  |  |  |
| --- | --- | --- |
| Имя команды | Входные данные | Ограничения |
| writeVolumeLabelAndOwner | -метка тома | не может превышать 12 символа |
| -имя владельца | не может превышать 12 символов |

Возможные коды завершения функции:

|  |  |
| --- | --- |
| Код | Значение |
| 0 | успешное завершение |
| 1 | Недопустимое имя владельца |
| 2 | Недопустимая метка тома |

# Алгоритм программы

Модуль записи метки тома и владельца вызывается монитором команд при вызове команды записи метки тома и владельца и производит запись метки тома и имени владельца.

Модуль записи метки тома и имени владельца при вызове производит проверку корректности вводимых данных (имя тома и имя пользователя), в случае неправильных входных параметров монитору команд выдается код ошибки. Затем происходит запись метки тома и имени владельца в открытый файл файловой системы.

## Используемые классы и методы

Модуль записи метки тома и имени владельца использует следущие классы и структуры данных:

* Класс файловой системы. Содержит открытый поток файла и методы, реализующие функции, вызываемые монитором команд.

|  |
| --- |
| class FileSystem  {  public:  fstream memory;  void createFile(string); //открытие существующего физического файла или создание  int format(string,string,string,string); //форматирование файловой системы  Block readBlock(int); //считывание блока, заданного номером  void writeBlock(Block,int); //запись блока в заданную позицию файла  FileDescriptor getRecord(string); //извлечение описателя файла по имени  int writeRecord(FileDescriptor); //сохранение записи о файле в конец  int deleteRecord(string); //удаление записи о файле  void closeFileSystem();  int showTableOfContent();  int showContentInAlphavit();  int createNewFile();  int showInformation();  int deleteFile();  int reNameFile();  int addInformationToFile();  int changeFileType();  int compressionOfFileSystem();  int writeVolumeLabelAndOwner();  }; |

# Модульное тестирование

Модульное тестирование функции форматирования производилось с использованием фреймворка для модульного тестирования Google C++ Test Framework.

Для запуска тестов необходимо запустить fileSystem.exe с передаваемым параметром runtest.

Написанные тесты для модуля форматирования:

|  |
| --- |
| #include "FileSystem.h"  #include <gtest\gtest.h>  class TestVolumeLabelAndOwner : public ::testing::Test {  public:  void SetUp() // инициализация тестируемого класса  {  f=new FileSystem;  f->createFile("3");  f->openFile("3");  f->format("23","tom","otherString","last");  FileDescriptor d;  FileDescriptor d1;  d.firstBlockNumber=f->toString(20,16);  d.blockCount=9;  d.descriptorType="002000";  d.fileName="name";  f->writeRecord(d,5);  d.fileName="ExistName";  f->writeRecord(d);  }  FileSystem \*f;  };  TEST\_F(TestVolumeLabelAndOwner, correctVolumeLabelAndOwner) //успешное задание метки тома и имени владельца  {  stringstream s;  s<<"LocalDriveD"<<endl;  s<<"Anonymous"<<endl;  //cout.rdbuf(s.rdbuf());  cin.rdbuf(s.rdbuf());  ASSERT\_EQ(f->writeVolumeLabelAndOwner(),0);  };  TEST\_F(TestVolumeLabelAndOwner, longVolumeLabel) //длинная метка тома и подходящее имя владельца  {  stringstream s;  s<<"LocalDriveD\_Doubled"<<endl;  s<<"Anonymous"<<endl;  //cout.rdbuf(s.rdbuf());  cin.rdbuf(s.rdbuf());  ASSERT\_EQ(f->writeVolumeLabelAndOwner(),1);  };  TEST\_F(TestVolumeLabelAndOwner, correctVolumeLabelAndLongOwner) //успешное задание метки тома и длинное имя владельца  {  stringstream s;  s<<"LocalDriveD"<<endl;  s<<"AnonymousDoubled"<<endl;  //cout.rdbuf(s.rdbuf());  cin.rdbuf(s.rdbuf());  ASSERT\_EQ(f->writeVolumeLabelAndOwner(),2);  };  TEST\_F(TestVolumeLabelAndOwner, LongVolumeLabelAndLongOwner) //длинная метка тома и имя владельца  {  stringstream s;  s<<"LocalDriveDDoubled"<<endl;  s<<"AnonymousDoubled"<<endl;  //cout.rdbuf(s.rdbuf());  cin.rdbuf(s.rdbuf());  ASSERT\_EQ(f->writeVolumeLabelAndOwner(),1);  };  int main(int argc, char \*\*argv)  {  if(argc > 1)  {  for (int i=1;i<argc;i++)  {  string tmp(argv[i]);  if (tmp.compare("runtest")==0)  {  ::testing::InitGoogleTest(&argc, argv);  RUN\_ALL\_TESTS();  getch();  }  }  }  return 0;  } |

Результаты прогонки тестов следующие:

|  |
| --- |
| [----------] 4 tests from TestVolumeLabelAndOwner  [ RUN ] TestVolumeLabelAndOwner.correctVolumeLabelAndOwner  Enter Volume Label  Enter the name of owner  [ OK ] TestVolumeLabelAndOwner.correctVolumeLabelAndOwner (109 ms)  [ RUN ] TestVolumeLabelAndOwner.longVolumeLabel  Enter Volume Label  [ OK ] TestVolumeLabelAndOwner.longVolumeLabel (110 ms)  [ RUN ] TestVolumeLabelAndOwner.correctVolumeLabelAndLongOwner  Enter Volume Label  Enter the name of owner  [ OK ] TestVolumeLabelAndOwner.correctVolumeLabelAndLongOwner (109 ms)  [ RUN ] TestVolumeLabelAndOwner.LongVolumeLabelAndLongOwner  Enter Volume Label  [ OK ] TestVolumeLabelAndOwner.LongVolumeLabelAndLongOwner (140 ms)  [----------] 4 tests from TestVolumeLabelAndOwner (468 ms total) |

# Исходный код модуля

|  |
| --- |
| #include "FileSystem.h"  int FileSystem::writeVolumeLabelAndOwner()  {  cout<<"Enter Volume Label"<<endl;  string VolName;  cin>>VolName;  if (VolName.length()>12) return 1;  cout<<"Enter the name of owner"<<endl;  string Owner;  cin>>Owner;  if (Owner.length()>12) return 2;  Block temp\_Block;  temp\_Block.InsertString(472,VolName);  temp\_Block.InsertString(484,Owner);  memory.write((char\*)temp\_Block.byteMassive, 512);  return 0;  } |