|  |
| --- |
| 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 создания файловой системы с заданными параметрами

Пояснительная записка

ГОСТ 19.404-79

Исполнитель Крючков А.

Малышев А.

Шевченко С.

Щипанов С.

Принял И.В. Алексеенко

Москва 2014

Оглавление

[1. Введение 4](#_Toc384216870)

# Введение

В данном документе представлен текст программного модуля «модуль создания файловой системы с заданными параметрами (форматирования) 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 классов файловой системы и монитора команд и их методов. Программа написана в среде Microsoft Visual Studio 2010, на языке С++.

# Исходный код модуля

## Текст файла FileSystem.h

Ниже приведен фрагмент текста программы, относящийся к модулю форматирования.

|  |
| --- |
| #include "Segment.h"  #include "FileDescriptor.h"  class FileSystem  {  public:  fstream memory;  int format(string,string,string,string);  void writeBlock(Block,int);  }; |

## Текст файла FileSystem.cpp

|  |
| --- |
| #include "FileSystem.h"  void FileSystem::writeBlock(Block input, int place)  {  memory.seekp((place)\*512);  memory.write((char\*)input.byteMassive, sizeof(input.byteMassive));  }  int FileSystem::format(string version, string tomName, string userName, string systemName)  {  if(version.length()<3||tomName.length()<13||userName.length()<13||systemName.length()<13) return 1;  memory.clear();  Block firstBlock; //загрузочный блок  firstBlock.Clean();  memory.write((char\*)firstBlock.byteMassive, sizeof(firstBlock.byteMassive));  Block systemBlock;  systemBlock.Clean(); //системный блок  systemBlock.InsertString(469,"6");  systemBlock.InsertString(470,version);  systemBlock.InsertString(472,tomName);  systemBlock.InsertString(484,userName);  systemBlock.InsertString(496,systemName);  memory.write((char\*)systemBlock.byteMassive, sizeof(firstBlock.byteMassive));  for (int i=0;i<4;i++)  {  Block emptyBlock; //4 пустых блока  emptyBlock.Clean();  memory.write((char\*)emptyBlock.byteMassive, sizeof(emptyBlock.byteMassive));  }  Segment catalog[31];  for (int i=0;i<30;i++)  {  catalog[i].clean();  catalog[i].segmentCount=31;  catalog[i].nextSegmentNumber=(i+1);  catalog[i].busySegmentCount=0;  catalog[i].write();  catalog[i].blockMassive[1].byteMassive[128]='1';  catalog[i].blockMassive[1].byteMassive[16+128]='5';  memory.write((char\*)catalog[i].blockMassive[0].byteMassive, sizeof(catalog[i].blockMassive[0].byteMassive));  memory.write((char\*)catalog[i].blockMassive[1].byteMassive, sizeof(catalog[i].blockMassive[1].byteMassive));  }  catalog[30].clean();  catalog[30].segmentCount=31;  catalog[30].nextSegmentNumber=(0);  catalog[30].busySegmentCount=0;  catalog[30].write();  memory.write((char\*)catalog[30].blockMassive[0].byteMassive, sizeof(catalog[30].blockMassive[0].byteMassive));  memory.write((char\*)catalog[30].blockMassive[1].byteMassive, sizeof(catalog[30].blockMassive[1].byteMassive));  return 0;  } |