ФГБОУ «Национальный исследовательский ядерный университет «МИФИ»

Кафедра «Управляющие интеллектуальные системы»

Группа К5-291

|  |
| --- |
| Заказчик |
| М.Н. Петухов |
| 2012 года |

Модель файловой системы ОС реального времени

Монитор обработки команд файловой системы

Описание программы

ГОСТ 19.402-78

Исполнитель С.С. Богатыренко

Принял И.В. Алексеенко

Москва 2012

1. **Аннотация**

В данном документе содержаться общие сведения о программе «Монитор обработки команд программы «Модель файловой системы ОС реального времени»», описание ее логической структуры и функционального назначения, входных и выходных данных, а также описание используемых технических средств.

Перед прочтением данного документа рекомендуется ознакомиться с пояснительной запиской, техническим заданием и текстом программы к учебному проекту «Монитор обработки команд программы «Модель файловой системы ОС реального времени»».

1. **Содержание**

[**1.** **Аннотация** 1](#_Toc341926874)

[**2.** **Содержание** 2](#_Toc341926875)

[**3.** **Общие сведения** 2](#_Toc341926876)

[**3.1** **Обозначение и наименование программы.** 3](#_Toc341926877)

[**3.2** **Программное обеспечение, необходимое для функционирования программы.** 3](#_Toc341926878)

[Для успешного выполнения программы требуется ОС Windows XP и выше с установленными драйверами для монитора и клавиатуры. 3](#_Toc341926879)

[**3.3** **Языки программирования, на которых написана программа.** 3](#_Toc341926880)

[Программа написана в среде Microsoft Visual Studio 2010, на языке С++. 3](#_Toc341926881)

[**4.** **Функциональное назначение** 3](#_Toc341926882)

[**5.** **Описание логической структуры** 3](#_Toc341926883)

[**5.1** **Алгоритм программы.** 3](#_Toc341926884)

[**5.2** **Используемые методы.** 4](#_Toc341926885)

[**5.3** **Структура программы** 6](#_Toc341926886)

[**5.4** **Связь программы с другими программами** 6](#_Toc341926887)

[**6.** **Используемые технические средства** 6](#_Toc341926888)

[**7.** **Вызов и загрузка** 6](#_Toc341926889)

[**8.** **Входные данные** 7](#_Toc341926890)

[**9.** **Выходные данные** 7](#_Toc341926891)

1. **Общие сведения**
   1. **Обозначение и наименование программы.**

Полное наименование программы: «Монитор обработки команд программы «Модель файловой системы ОС реального времени»», краткое наименование «МК ».

* 1. **Программное обеспечение, необходимое для функционирования программы.**

Для успешного выполнения программы требуется ОС Windows XP и выше с установленными драйверами для монитора и клавиатуры.

* 1. **Языки программирования, на которых написана программа.**

Программа написана в среде Microsoft Visual Studio 2010, на языке С++.

1. **Функциональное назначение**

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

МК - основной модуль программы «Модель файловой системы ОС реального времени». Он предоставляет пользователю набор команд для вызова подпрограмм, имитирующих работу файловой системы, проверяет правильность использования этих команд, обеспечивает корректность передаваемых в вызываемые подпрограммы параметров и контролирует результаты их выполнения. Помимо этого, МК предоставляет пользователю справку по работе программы и по отдельным командам. Кроме того, МК обеспечивает системным программистам интерфейс для вывода информационных сообщений и сообщений об ошибках.

1. **Описание логической структуры**
   1. **Алгоритм программы.**

Работа программы заключается в считывании команд пользователя, вызове по ним необходимых подпрограмм и передаче пользователю сообщений о работе этих подпрограмм либо о некорректности введенных команд.

Для выполнения этих задач в программе имеется цикл, в котором поочередно производится ввод команды пользователем, вызов необходимой процедуры и обработка результатов исполнения процедуры. На этапе ввода команды производится разбор и предварительный анализ содержимого командной строки. При несоответствии содержимого шаблону (по размеру, по введенным символам и т.д.) выводится сообщение об ошибке, и ввод начинается заново. При корректности вводимой пользователем информации производится поиск имени введенной команды в имеющемся перечне команд и, при удачном поиске, производится окончательный анализ введенных параметров на соответствие данной команде. При корректных параметрах производится, если необходимо, преобразование их типов и вызов подпрограммы, соответствующей данной команде с передачей параметров и последующим сохранением возвращаемого кода. Если этапе окончательного анализа данных обнаруживается некорректность данных (введена неизвестная команда или параметры не подходят для данной команды), то вместо возвращаемого кода подпрограммы устанавливается специальный код ошибки. На последнем этапе происходит проверка возвращенного подпрограммой кода (или установленного кода ошибки) и вывод необходимых сообщений пользователю. При вводе пользователем команды выхода производится выход из цикла и завершение программы.

Для вывода подпрограммами информационных сообщений и сообщений об ошибках используются специальные глобальные буферы сообщений и флаги управления ими. При необходимости вывода сообщения подпрограмма устанавливает флаг буфера и выводит сообщение в буфер, затем монитор команд при анализе возвращенного подпрограммой кода проверяет состояние флага и выводит сообщение, записанное подпрограммой в буфер. Такой же механизм вывода сообщений об ошибках (при окончательном анализе параметров команд) используется и самим монитором команд.

* 1. **Используемые методы.**

При построении программы были созданы специальные классы файловой системы и монитора команд. В исходном коде программы вызываются методы класса монитора команд, а они, в свою очередь, вызывают методы класса файловой системы.

Описание методов класса монитора команд class Monitor:

* Monitor() – конструктор класса; вызывает конструктор класса файловой системы для создания пустой файловой системы, инициализирует список команд, инициализирует нулевыми значениями прочие поля в объекте класса
* void readcom() – производит считывание введенной пользователем команды и её предварительный синтаксический анализ, при получении некорректных данных выводит об этом сообщение и начинает считывание снова.
* void execute() – производит окончательный анализ введенной команды и её параметров. При корректных данных вызывает соответствующий метод класса файловой системы и сохраняет возвращенный код, при некорректных - вместо возвращаемого кода подпрограммы устанавливает специальный код ошибки.
* void errcheck() – проверяет возвращенный подпрограммой (или установленный методом execute) код, выводит соответствующие ему сообщения, при необходимости – выводит содержимое буфера информационных сообщений или буфера сообщений об ошибках.

Описание методов класса файловой системы class FS:

* FS() – конструктор класса, создает пустую файловую систему
* int format(unsigned long size, char\* NT, char\* NO) - файловую систему заданного размера, устанавливает её метку тома и имя владельца
* int content() - выводит перечень файлов в системе в том порядке, в котором они там находятся
* int sortcontent() - выводит перечень файлов в системе в алфавитном порядке
* int freespace() - выводит перечень пустых областей в системе
* int create(char\* name, char\* type, unsigned short size) - создает в системе файл с заданными именем и типом, имеющий заданный размер
* int del(char\* name, char\* type) - удаляет из системы файл с заданными именем и типом
* int rename(char\* oldname, char\* type,char\* newname) - меняет имя файла с заданными именем и типом на новое
* int append(char\* name, char\* type,unsigned short size) - увеличивает размер файла с заданными именем и типом на указанную величину
* int changetype(char\* name, char\* oldtype,char\* newtype) - меняет тип файла с заданными именем и типом на новый
* int defrag() - дефрагментирует файловую систему
* int userandlabel(char\* owner,char\* label) - меняет имя владельца и метку тома в системе на новые
* int createFS() - создает на жестком диске файл filesystem.txt со структурой, соответствующей созданной файловой системе
  1. **Структура программы**

Программа может быть разделена на следующие уровни:

* Исходный код. Исполняемый в программе цикл. Предварительно создается объект класса монитор команд, затем в цикле вызываются его методы.
* Класс монитора команд class Monitor. Отвечает за взаимодействие пользователя с моделью файловой системы, представленной объектом класса FS, который содержится в объекте класса Monitor. Методы класса осуществляют считывание и проверку команд пользователя и вызов методов класса FS.
* Класс файловой системы FS. Объект класса представляет собой модель файловой системы. Содержит необходимые информационные поля и методы, которые имитируют работу файловой системы. Методы реализуются отдельными программистами.

* 1. **Связь программы с другими программами**

Вызов данной программы может осуществляться из других процессов средствами операционной системы, при вызове программа не требует каких либо параметров.

1. **Используемые технические средства**

Для успешного выполнения программы требуется IBM – совместимый компьютер под управлением ОС Windows XP и выше, имеющий клавиатуру для ввода информации оператором и монитор для вывода информации и жесткий диск или съемный носитель для записи информации. Компьютер должен иметь процессор с тактовой частотой не менее 233 МГц и объем оперативной памяти не менее 64 Мб.

1. **Вызов и загрузка**

Запуск монитора производится запуском соответствующего exe файла. В соответствии с п.4.5 технического задания к учебному проекту «Монитор обработки команд программы «Модель файловой системы ОС реального времени»» он должен называться ФС.exe и должен находиться в подкаталоге Release каталога ФС

1. **Входные данные**

Программа получает данные от оператора посредством ввода строки с командой и параметрами.

1. **Выходные данные**

Вывод данных осуществляется через консоль в ответ на действия пользователя. Кроме того, предоставляется возможность создания на жестком диске файла со структурой, соответствующей созданной файловой системе.