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

Руководство системного программиста

ГОСТ 19.503-79

Исполнитель С.С. Богатыренко

Принял И.В. Алексеенко

Москва 2012

1. **Аннотация**

Документ содержит общую информацию по работе программы «Монитор обработки команд программы «Модель файловой системы ОС реального времени»»

Приводится описание структуры программы, настройки, проверки и сообщений системному программисту.

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](#_Toc341939597)

[**2.** **Содержание** 2](#_Toc341939598)

[**3.** **Общие сведения о программе** 3](#_Toc341939599)

[**4.** **Структура программы** 3](#_Toc341939600)

[**5.** **Настройка программы** 6](#_Toc341939601)

[**6.** **Проверка программы** 6](#_Toc341939602)

[**7.** **Дополнительные возможности** 6](#_Toc341939603)

[**8.** **Выводимые сообщения** 7](#_Toc341939604)

1. **Общие сведения о программе**

Полное наименование программы: «Монитор обработки команд программы «Модель файловой системы ОС реального времени»», краткое наименование «МК ». Программа «Модель файловой системы ОС реального времени» предназначена для изучения устройства и функционирования файловых систем.

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.

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

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.

Каждому из уровней соответствует свой модуль среди файлов КП:

Исходный код- Code.cpp

Класс монитора команд class Monitor-Monitor.h

Класс файловой системы FS-FS.h

На уровне исходного кода возможно редактирование порядка вызова методов класса монитора команд и внесение дополнительных возможностей в программу.

Класс монитора команд предоставляет в распоряжение программиста следующие методы:

* 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.

Класс файловой системы предоставляет в распоряжение программиста следующие методы:

* 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

В модуле FS.h объявляются глобальные буферы сообщений stringstream screen, err и флаги для работы с буферами bool screenflag,errflag. Эти буферы могут быть использованы для вывода сообщений с уровня класса файловой системы: метод класса файловой системы записывает информацию в буфер и устанавливает флаг буфера, метод класса монитора команд выводит сообщение из буфера и снимает флаг.

* 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. **Настройка программы**

Настройка программы сводится к инициализации объекта класса монитора команд при помощи конструктора класса. При этом инициализируется список команд монитора, инициализируются нулевыми значениями прочие поля объекта, создается пустой объект класса файловой системы. Для дальнейшей работы модели необходимо отформатировать файловую систему – выделить место под каталог с записями о файлах и создать там необходимые записи. Для этого используется метод int format(unsigned long size, char\* NT, char\* NO) файловой системы.

1. **Проверка программы**

Проверка программы реализуется выполнением различных команд монитора и проверкой получаемых результатов. Описание проверки можно найти в документе: Программа и методика испытаний к учебному проекту ««Монитор обработки команд программы «Модель файловой системы ОС реального времени»»

1. **Дополнительные возможности**

Дополнительные возможности в данном комплексе программ не предусмотрены.

1. **Выводимые сообщения**

|  |  |
| --- | --- |
| Причина появления | Текст сообщения |
| Приветствие | Вас приветствует модель файловой системы ОС реального времени. Данная программа моделирует работу файловой системы и имитирует её основные функции.  Чтобы получить информацию о предоставляемых командах введите “help”. |
| **Сообщения об ошибках** | |
| Превышено максимально допустимое количество символов в команде | Превышена допустимая длина команды! |
| В команде обнаружены русские символы | Русские символы не разрешены! |
| Введено неверное число параметров команды | Неверное число параметров! |
| Введено неверное имя команды | Неверное имя команды! |
| Введен неверный первый параметр команды | Неверный первый параметр! |
| Введен неверный второй параметр команды | Неверный второй параметр! |
| Введен неверный третий параметр команды | Неверный третий параметр! |
| Попытка выполнить операцию с неотформатированной системой | Файловая система не отформатирована! |
| Непредвиденная ошибка во время выполнения команды | Во время выполнения команды произошла непредвиденная ошибка! |
| Попытка получить содержимое пустой файловой системы | Файлы в системе отсутствуют! |
| Попытка получить информацию о свободных областях полностью заполненной системы | Свободные области в системе отсутствуют! |
| Недостаток места в системе при записи/перезаписи файла | Недостаточно места в системе! |
| Недостаток места в каталоге при записи/перезаписи файла | Недостаточно места в каталоге! |
| Попытка создания уже существующего файла | Такой файл уже существует! |
| Попытка обращения к несуществующему файлу | Такого файла нет в системе! |
| Ошибка записи файла на жесткий диск | Не удалось записать файл на жесткий диск! |
| Ошибка в МК | Во время работы программы возникла непредвиденная ошибка. Программа завершает своё выполнение. |
| **Сообщения об успешном выполнении** | |
| format | Файловая система отформатирована |
| content, sortcontent | Содержимое файловой системы:  «содержимое» |
| freespace | Информация о свободных областях:  «информация» |
| create | Файл «имя.тип» с размером «размер» успешно создан |
| del | Файл «имя.тип» удален |
| rename | Файл «имя.тип» переименован в «имя2.тип» |
| append | Размер файл «имя.тип» увеличен на «размер» |
| changetype | Тип файла «имя.тип» изменен на «.тип2» |
| defrag | Файловая система сжата |
| userandlabel | Имя владельца и метка тома изменены на «метка тома» и «имя владельца» |
| createFS | Файл filesystem.txt успешно создан |
| exit | Завершение работы. |
| **Справочные сообщения** | |
| help | Команды в системе имеют следующий формат:  имя\_команды параметр\_1 параметр\_2 параметр\_3  Количество параметров у разных команд может быть от 0 до 3  При задании параметров не допускается использование символов русского алфавита  В системе имеются следующие команды:  format размер\_системы метка\_тома имя\_владельца  content  sortcontent  freespace  create имя\_файла тип\_файла размер\_файла  del имя\_файла тип\_файла  rename старое\_имя\_файла тип\_файла новое\_имя\_файла  append имя\_файла тип\_файла добавляемый\_размер  changetype имя\_файла старый\_тип\_файла новый\_тип\_файла  defrag  userandlabel имя\_владельца метка\_тома  createFS  help  exit  Для получения информации о команде введите её имя с вопросительным знаком в конце, например format? |
| format? | format размер\_системы метка\_тома имя\_владельца  Данная команда создает файловую систему заданного размера, устанавливает её метку тома и имя владельца  Размер задается в блоках, он должен быть больше 9-ти и не должен превышать 65536-ти  Имя владельца и метка тома не должны быть длиннее 12-ти символов |
| content? | content  Данная команда выводит перечень файлов в системе в том порядке, в котором они там находятся |
| sortcontent? | sortcontent  Данная команда выводит перечень файлов в системе в алфавитном порядке |
| freespace? | freespace  Данная команда выводит перечень пустых областей в системе |
| create? | create имя\_файла тип\_файла размер\_файла  Данная команда создает в системе файл с заданными именем и типом, имеющий заданный размер  Имя создаваемого файла не должно быть длиннее 6-ти символов, тип - 4-х  Размер должен быть ненулевым и не должен превышать 65468-ми |
| del? | del имя\_файла тип\_файла  Данная команда удаляет из системы файл с заданными именем и типом  Имя файла не должно быть длиннее 6-ти символов, тип - 4-х |
| rename? | rename старое\_имя\_файла тип\_файла новое\_имя\_файла  Данная команда меняет имя файла с заданными именем и типом на новое  Имя файла не должно быть длиннее 6-ти символов, тип - 4-х |
| append? | append имя\_файла тип\_файла добавляемый\_размер  Данная команда увеличивает размер файла с заданными именем и типом на указанную величину  Имя файла не должно быть длиннее 6-ти символов, тип - 4-х  Размер должен быть ненулевым и не должен превышать 65468-м |
| changetype? | changetype имя\_файла старый\_тип\_файла новый\_тип\_файла  Данная команда меняет тип файла с заданными именем и типом на новый  Имя файла не должно быть длиннее 6-ти символов, тип - 4-х |
| defrag? | defrag  Данная команда дефрагментирует файловую систему |
| userandlabel? | userandlabel имя\_владельца метка\_тома  Данная команда меняет имя владельца и метку тома в системе на новые  Имя владельца и имя тома не должны быть длиннее 12-ти символов |
| createFS? | createFS  Данная команда создает на жестком диске файл filesystem.txt со структурой, соответсвующей созданной файловой системе |
| help? | help  Данная команда выводит список имеющихся команд и их параметров |
| exit? | exit  Данная команда осуществляет выход из приложения |