МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

КАФЕДРА ПРОГРАМНОЙ ИНЖЕНЕРИИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

на тему: «Проектирование гипотетической операционной системы»

по курсу: «Теория операционных систем»

|  |  |
| --- | --- |
| Руководитель: | Выполнила: |
| \_\_\_\_\_\_\_\_ каф. ПИ | ст. гр. ПИ-16Б  Мамутова В.А. |
| Чернышева А.В. | “\_\_”\_\_\_\_\_\_ 2018 г. |
| “\_\_”\_\_\_\_\_\_ 2018 г. |  |
|  |  |
|  |  |

Донецк – 2018Реферат

Отчет по курсовому проекту содержит: страницы, рисунков, таблиц, приложения, источников.

Объект исследования – операционные системы, их проектирование и эмуляция работы.

Задача исследования – изучить принципы работы операционных систем.

Цель исследования – выполнить обоснованный проект операционной системы, удовлетворяющей ряду специальных требований, приведенных в варианте задания, выполнить эмуляцию работы отдельных модулей спроектированной операционной системы.

Результат – проект операционной системы, программа на языке программирования C#, которая эмулирует работу файловой системы и межпроцессного взаимодействия спроектированной операционной системы.

Данная работа может быть использована в учебных целях для демонстрации работы файловой системы.

ОПЕРАЦИОННАЯ СИСТЕМА, многопользовательская зашита, межпроцессное взаимодействие, КОМАНДНЫЙ ИНТЕРПРеТАТОР, ФАЙЛОВАЯ СИСТЕМА, ntfs, Эмуляция

Содержание

[Введение 5](#_Toc528595030)

[1 Структура файловой системы 6](#_Toc528595031)

[1.1 Общая организация файловой системы 6](#_Toc528595032)

[1.1.1. Структура записи Master File Table 9](#_Toc528595033)

[1.1.2. Структура служебного файла Volume 12](#_Toc528595034)

[1.1.3. Структура записи корневого каталога ($.) 13](#_Toc528595035)

[1.1.4. Структура битовой карты свободных и занятых кластеров 13](#_Toc528595036)

[1.1.5. Структура списка пользователей системы (Users) 14](#_Toc528595037)

[1.1.6. Область файлов и директорий 14](#_Toc528595038)

Введение

Структура проектируемой файловой системы

1.1 Общая организация файловой системы

1.2 Виртуальные страницы

1.3 Команды для работы с ФС

1.4 Системные вызовы для работы с ФС

1.5 Способы организации файлов

1.6 Алгоритмы работы некоторых системных вызовов ФС.

2 Процессы в ОС

2.1 Команды для работы с процессам

2.2 Системные вызовы управления процессами

2.3 Диаграмма состояний процесса

2.4 Приоритеты процессов

2.5 Межпроцессное взаимодействие.

2.6 Выбор дисциплины обслуживания планировщика процессов. Алгоритм работы планировщика процессов в соответствии с выбранной дисциплиной обслуживания.

2.7 Свопинг процессов.

3 Режимы работы проектируемой ОС

3.1 Мультипрограммный режим работы ОС

3.2 Многопользовательская защита

3.3 Интерактивный режим работы ОС

3.4 Пакетный режим работы

4 Структура операционной системы

4.1 Общая структура проектируемой ОС

4.2 Структура ядра проектируемой ОС.

Основные функции и назначение файловой подсистемы, подсистемы управления памятью и процессами, подсистемы управления устройствами.

4.3 Структура управляющих блоков базы данных ОС

5 Разработка программ эмуляции ОС

5.1 Описание программных средств

5.2 Разработка ФС

5.3 Разработка командного интерпретатора

5.4 Эмуляция планирования (или межпроцессного взаимодействия)

6 Тестирование программы.

Анализ результатов

Выводы

Перечень ссылок

Приложение А. Таблица трассировки

Приложение Б. Экранные формы, отображающие результаты работы программ эмуляции

Приложение В. Листинг программ

# Введение

Операционная система (ОС) представляет собой комплекс управляющих и обрабатывающих программ, которые выполняют такие функции, как загрузка программ в оперативную память, их выполнение, управление оперативной памятью, управление доступом к данным. Также ОС обеспечивает стандартный доступ к периферийным устройствам, эффективно распределяет ресурсы вычислительной системы между процессами и т.д.

Предшественником ОС следует считать служебные программы (загрузчики и мониторы), а также библиотеки часто используемых подпрограмм, начавшие разрабатываться с появлением универсальных компьютеров 1-го поколения (конец 1940-х годов). Служебные программы минимизировали физические манипуляции оператора с оборудованием, а библиотеки позволяли избежать многократного программирования одних и тех же действий (осуществления операций ввода-вывода, вычисления математических функций и т.п.). В 1950-60-х годах сформировались и были реализованы основные идеи, определяющие функциональность ОС: пакетный режим, разделение времени и многозадачность, разделение полномочий, реальный масштаб времени, файловые структуры и файловые системы.

Для решения задач управления ресурсами разные ОС используют различные алгоритмы, особенности которых в конечном счете и определяют облик ОС. В число требований, предъявляемых сегодня к сетевым ОС, входят: функциональная полнота и эффективность управления ресурсами, модульность и расширяемость, переносимость и многоплатформенность, совместимость на уровне приложений и пользовательских интерфейсов, надежность отказоустойчивость, безопасность и производительность.

Целью данной курсовой работы является проектирование ОС, а также создание эмулятора, который будет демонстрировать работу файловой системы и реализовать межпроцессное взаимодействие процессов.

# 1 Структура файловой системы

Файловая система - это часть операционной системы, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и обеспечить совместное использование файлов несколькими пользователями и процессами.

В широком смысле понятие "файловая система" включает:

* совокупность всех файлов на диске,
* наборы структур данных, используемых для управления файлами, такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске,
* комплекс системных программных средств, реализующих управление файлами, в частности: создание, уничтожение, чтение, запись, именование, поиск и другие операции над файлами.

Если рассматривать диск с точки зрения операционной системы, то он представляет собой набор кластеров (логических единиц хранения данных). Драйверы файловой системы организуют кластеры в файлы и каталоги (реально являющиеся файлами, содержащими список файлов в этом каталоге). Эти же драйверы отслеживают, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.

## Общая организация файловой системы

При проектировании файловой системы, в качестве основы была выбрана система NTFS (от англ. New Technology File System — «файловая система новой технологии»).

NTFS – файловая система, которая разрабатывалась Microsoft специально для Windows NT. Прежде всего, Windows NT – это ОС корпоративного уровня. Предыдущие файловые системы FAT и HPFS (ФС для OS/2) были способны удовлетворить потребностям рядовых пользователей, но не имели достаточных возможностей для корпоративной среды, таких как защищенность файлов, отказоустойчивость и восстанавливаемость. Устранить перечисленные недостатки призвана была новая операционная система NTFS. Это восстанавливаемая, защищенная и надежная файловая система, поддерживающая диски и файлы большого объёма. В этой файловой системе обеспечиваются множественные потоки данных, имена в UNICODE, универсальное средство индексации, переназначение плохих кластеров и поддержка POSIX. NTFS заимствовала лучшие свойства FAT и HPFS. От FAT система NTFS заимствовала философию “простота рождает эффективность”. От HPFS система NTFS унаследовала методы для повышения быстродействия и гибкости в файловой системе.

Из недостатков можно отметить существенные требования к оперативной памяти и то, что медленные диски и контроллеры сильно снижают быстродействия.

Несмотря на все вышеизложенные плюсы и минусы, определение структуры файловой системы было сделано на основе NTFS.

Файловая система NTFS представляет собой выдающееся достижение структуризации: каждый элемент системы представляет собой файл - даже служебная информация. Самый главный файл в NTFS называется MFT (англ. Master File Table – «общая таблица файлов»). Он размещается в MFT-зоне и представляет собой централизованный каталог всех остальных файлов диска и себя самого. MFT-зона – служебное пространство, которое занимает первые 12% диска. MFT-зона всегда держится пустой - это делается для того, чтобы самый главный, служебный файл (MFT) не фрагментировался при своем росте. Остальные 88% диска представляют собой обычное пространство для хранения файлов.

MFT поделен на записи фиксированного размера (обычно 1 Кбайт), и каждая запись соответствует, какому-либо файлу (в общем смысле этого слова). Первые 16 файлов носят служебный характер и недоступны операционной системе - они называются метафайлами, причем самый первый метафайл - сам MFT. Эти первые 16 элементов MFT - единственная часть диска, имеющая фиксированное положение. Интересно, что вторая копия первых трех записей для надежности хранится ровно посередине диска. Остальной MFT-файл может располагаться, как и любой другой файл, в произвольных местах диска - восстановить его положение можно с помощью его самого, взяв за основу первый элемент MFT.

При проектировании файловой системы структура NTFS была упрощена. Для большей наглядности и разъяснения, данная структура будет представлена в таблице 1.1. А также будут сделаны пояснения к каждому из вышеперечисленных пунктов.

Таблица 1.1 - Логическая структура диска

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MFT-пространство | | | | | | |  | | |
| MFT-файл | | | | | |  |
| $MFT | $MFTMirr | $volume | $. (корневой каталог) | $BitMap | $Users | … | Область данных | $MFTMirr (Копия первых записей $MFT) | Область данных |

Пояснения для служебных областей:

1. $MFT - сам MFT;
2. $MFTMirr - зеркальное отображение наиболее важных частей MFT. Используется для гарантирования доступа к MFT в случае повреждений в начале диска, в разрабатываемой файловой системе это копия первых 6 служебных записей MFT, размещенная посередине диска;
3. $Volume содержит служебную информацию – метка тома, версия файловой системы, состояние тома (повреждён/не повреждён);
4. $. (корневой каталог);
5. $BitMap – карта свободного места тома;
6. $Users – список пользователей.

Размер раздела – 400 Мб (419 430 400 Б)

Размер кластера – 4096 Б

Количество кластеров – 102 400 ед.

Рассмотрим более подробно служебные файлы.

### Структура записи Master File Table

Сам MFT состоит из записей фиксированного размера по 1024 Б.

Условно поделим структуру записи на 5 логических блоков: Signature, Standard Information, File Name, Security Descriptor, Data (см. табл. 1.2).

Таблица 1.2 – Структура записи файла $MFT, определяющей небольшой файл или каталог

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signature | Standard Information | File Name | Security Descriptor | Data |

Signature. Начало записи идентифицируется его сигнатурой - признаком записи, который указывает, является ли запись файлом, описывает ли она каталог, хранит ли данные файла либо список управления доступа к файлу или не используется вообще.

Standard information. В блоке со стандартной информацией о файле содержатся стандартная информация о файле: его атрибуты, расширение, размер, время создания и модификации файла, идентификатор владельца.

File Name. В данном поле определяется имя файла.

Security Descriptor. Дескриптор безопасности хранит номер записи MFT, в которой хранится список управления доступом к файлу.

Data. Здесь содержатся все данные файла.

Структура записи представлена в таблице 1.3.

Таблица 1.3 – Пример структуры записи $MFT

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Поле | Размерность | Назначение |
| enum | signature | 1 Б | Признак записи |
| byte | attributes | 1 Б | Атрибуты |
| string | extension | 5 Б | Расширение |
| int | size | 4 Б | Размер |
| short | creatDate | 5 Б | Дата создания |
| short | modifDate | 5 Б | Дата модификации |
| byte | userID | 1 Б | User ID |
| string | fileName | 25 Б | Имя файла |
| int | securityDescriptor | 4 Б | Дескриптор безопасности |
| byte | dataAttributes | 1 Б | Заголовок Data |

Все указанные поля (первые логические четыре блока) в сумме занимают 52 байта, значит под данные (Data) в одной записи выделяется 972 байта.

В разрабатываемой операционной системе MFT-пространство будет занимать 10 % от общего размера диска, то есть 40 Мб = 41 943 040 Б. Количество кластеров под MFT-зону – 10240 ед.

Возможные значения перечисления Признака записи (signature) представлены в таблице 1.4.

Таблица 1.4 – Флаги для перечисления Признак записи

|  |  |
| --- | --- |
| Значение | Имя |
| 0 | NOT\_USED |
| 1 | IN\_USE |
| 2 | IS\_ACCESS\_CONTROL\_LIST |
| 3 | IS\_DATA |

Предопределённые атрибуты файла представлены в таблице 1.5.

Таблица 1.5 – Атрибуты файла

|  |  |
| --- | --- |
| Бит | Описание |
| 7 | Зарезервировано |
| 6 | Зарезервировано |
| 5 | Зарезервировано |
| 4 | Зарезервировано |
| 3 | directory |
| 2 | hidden |
| 1 | system |
| 0 | readOnly |

Если значение бита равно 1, файл обладает данным атрибутом. Если значение бита равно 0, файл не обладает данным атрибутом.

Если запись файла содержит список управления доступом (ACL – Access Control List), то её структура данных будет отличаться. Каждая заявка будет состоять из идентификатора пользователя (UID) или группы и предоставленных ему/ей прав (root) (см. таблицу 1.6).

Таблица 1.6 – Структура заявки в списке управления доступом

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Поле | Размерность | Назначение |
| byte | UID | 1 Б | Идентификатор пользователя (группы) |
| byte | Rights | 1 Б | Права доступа |

Всего одна заявка занимает 3 байта. В системе предусмотрена возможность создания 15 аккаунтов. Следовательно, список управления доступом будет занимать 45 байтов.

Поле «Права доступа» содержит набор битовых полей, структура которых описана в таблице 1.7.

Таблица 1.7 – Структура поля «Права доступа»

|  |  |  |
| --- | --- | --- |
| Тип прав | Бит | Описание |
|  | 7 | Зарезервирован |
|  | 6 | Зарезервирован |
|  | 5 | Зарезервирован |
|  | 4 | Зарезервирован |
| Полный доступ (Full Control) | 3 | Предоставляет полный доступ к файлу/папке, то есть допускаются все действия, предусмотренные всеми перечисленными ниже разрешениями. Дополнительно позволяется становиться владельцем файла/папки и изменять ее разрешения. |
| Изменение (Modify) | 2 | Разрешает все действия, которые позволяют выполнять «Чтение» и «Запись», а также разрешает удаление файла/папки. |
| Чтение (Read) | 1 | Разрешается чтение файла/просмотр вложенных папок и файлов, а также просмотр информации о них. |
| Запись (Write) | 0 | Разрешается перезапись файла/создание и размещение внутри папки новых файлов и подпапок, а также изменение информации о них. |

Если значение бита равно 1, доступ по соответствующему критерию разрешён. Если значение бита равно 0, доступ запрещён.

Если данные файла не помещаются в одну запись MFT, то это отображается в заголовке атрибута Data (табл. 1.8), который в данном случае будет содержать признак того, что атрибут является нерезидентным (не помещается в одну запись и находится в других местах раздела).

Таблица 1.8 – Значения заголовка Data

|  |  |
| --- | --- |
| Значение | Описание |
| 0 | резидентный атрибут (в данном поле содержатся данные файла) |
| 1 | нерезидентный атрибут (в данном поле содержится адресная информация следующих блоков данных) |

В качестве адресной информации для больших файлов в атрибуте Data будем хранить номера записей MFT для служебных файлов или номера блоков данных (для пользовательских файлов).

Пример размещения большого файла приведён на рисунке 1.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IN\_USE | Standard Information | File Name | Security Descriptor | Data  N1, N2, N3 |
| … | … | | | |
| … | … | | | |
| IS\_ACCESS\_ CONTROL\_LIST |  | | | |
| … | … | | | |
| IS\_DATA | Отрезок данных 1 | | | |
| … | … | | | |
| IS\_DATA | Отрезок данных 2 | | | |
| … | … | | | |
| … | … | | | |
| IS\_DATA | Отрезок данных 3 | | | |

Рисунок 1 – Размещение большого файла

### Структура служебного файла Volume

Структура служебного файла $Volume представлена в таблице 1.10.

Таблица 1.10 – Структура $Volume

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Поле | Размерность | Назначение |
| string | name | 1 Б | Метка тома |
| string | version | 9 Б | Версия ФС |
| byte | state | 1 Б | Состояние |

### Структура записи корневого каталога ($.)

Запись корневого каталога имеет следующую структуру, представленную в табл. 1.11.

Таблица 1.11 – Структура записи корневого каталога

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Поле | Размерность | Назначение |
| string | fileName | 24 Б | Имя файла |
| byte | attributes | 1 Б | Атрибуты |
| string | extension | 5 Б | Расширение |
| int | size | 4 Б | Размер |
| short | modifDate | 5 Б | Дата модификации |
| int | number | 4 Б | Номер записи в MFT |

Каждая запись, находящаяся в каталоге, занимает 43 байта.

Учитывая то, что количество выделяемых кластеров под область файлов 100 единиц, то размер корневого каталога = 100 \* 4096 = 409 600 байтов.

### Структура битовой карты свободных и занятых кластеров

Битовая карта — служебная структура в сложных файловых системах, хранящая информацию о наличии и расположении свободного места. Битовая карта будет представлена битовым массивом, в котором каждому кластеру будет соответствовать 2 бита. Возможные значения состояния кластера представлены в таблице 1.12.

Таблица 1.12 – Значения битовой карты свободных и занятых кластеров

|  |  |
| --- | --- |
| Значение | Описание |
| 00 | свободен |
| 01 | повреждён (Bad-кластер) |
| 10 | служебный |
| 11 | занят |

Всего кластеров в разделе 102 400 ед., значит размер битовой карты – 102 400 \* 2 = 204 800 бит = 25 600 байт (всего 4 блока).

### Структура списка пользователей системы (Users)

Структура списка пользователей представлена в таблице 1.13.

Таблица 1.13 – Структура записи в списке пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Поле | Размерность | Назначение |
| string | login | 20 Б | Логин пользователя |
| string | name | 30 Б | Имя пользователя |
| byte | UID | 1 Б | Уникальный идентификатор |
| string | salt | 24 Б | Соль |
| string | password | 88 Б | Пароль |
| string | homeDirectory | 30 Б | Домашняя директория |

Общий размер одной записи – 193 Б. Всего в системе будет определено 15 пользователей. Тогда суммарный размер списка пользователей – 2925 байтов. Следовательно, данный список может занимать до трёх записей в MFT.

### Область файлов и директорий

В разрабатываемой операционной системе данная область содержит в себе различные файлы, а также их различные расширения. Размерность данной области - 90 % от области диска, то есть 360 Мб = 377 487 360 Б = 92 160 блоков.