

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования
"Московский технический университет связи и информатики"



Методические указания
для проведения лабораторной работы №5

по дисциплине

«СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ»

по теме

«Исследование структуры файловой системы и системных ресурсов»

Направление подготовки:

09.03.01 Информатика и вычислительная техника

Профили

**Программное обеспечение и интеллектуальные системы
Вычислительные машины, комплексы, системы и сети**

Ростов-на-Дону
2019

УДК 681.3.06 (076)
ББК 32.07

Чикалов А.Н. Системное программное обеспечение. Исследование структуры файловой системы и системных ресурсов. Методические указания для проведения лабораторной работы №5. Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2019.- 24 с.

В пособии изложены методические рекомендации и содержательные материалы для проведения занятий изучению системных ресурсов и алгоритмов работы файловой системы современных ОС, а также утилит для изучения, контроля и восстановления данных в файловой системе. Кроме того рассматриваются механизмы управления файловой системой программными средствами с помощью интерфейса прикладного программирования.

Пособие содержит необходимые справочные материалы.

Методические указания предназначены для студентов, обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника, профиля Вычислительные машины, комплексы, системы и сети, Программное обеспечение и интеллектуальные системы.

Пособие предназначено для использования при изучении дисциплин Системное программное обеспечение, а также может быть использовано преподавателями и студентами при изучении родственных дисциплин и в процессе самостоятельной работы.

Учебное пособие обсуждено и одобрено на заседании кафедры ИВТ
Протокол №1 от 26.08.2019 г.

Рецензент Зав. кафедрой ИВТ д.т.н. профессор Соколов С.В.

Лабораторная работа 5. Исследование структуры файловой системы и системных ресурсов.

Цель

1. Выработать практические умения использования системных утилит для восстановления информации на диске.
2. Исследовать структуру системных ресурсов.
3. Приобрести навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных результатов, точного и лаконичного представления докладов по техническим вопросам.

Учебные вопросы

1. Исследование структуры системных ресурсов с помощью утилит.
2. Исследование структуры магнитного диска.

Литература для подготовки к занятию

1. Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение. Питер. 2001. с.146-166.
2. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. Питер. 2001. с.270-294.
3. Финогенов К.Г. Самоучитель по системным функциям MS-DOS. М. Радио и связь. Энтроп. 1995. с.56-98.

Дополнительная литература

1. Нортон П. Персональный компьютер фирмы IBM и операционная система MS-DOS. М.: Радио и связь, 1991.
2. Скляр В.А. Применение ПЭВМ. Кн.1. Операционные системы ПЭВМ. Практическое пособие. - М.: ВШ, 1992.
3. Крамм Р. Нортонские утилиты изнутри. М.: Мир, 1992.
4. Дайсон П. Овладеваем пакетом Norton Utilities 6. М.: Мир, 1993.
5. Нортон П., Джорден. Работа с жестким диском IBM PC. М.: Мир, 1992.
6. Данкан Р. Профессиональная работа в MS-DOS. – М.: Мир, 1993. с.142-156.
7. Войников Н.А. Системное программирование для ПРАВЕЦ-16. – София: Техника, 1990. –с.43-99.

Содержание отчета

1. Название лабораторной работы и учебные вопросы.
2. Последовательно для каждого из разделов исследования:
 - задания и отчетные сведения в соответствии с порядком исследования и руководством;

- схемы и рисунки, раскрывающие структуру исследуемых параметров и схемы операций выполняемых действий (разрешается дополнение своего конспекта соответствующей лекции);

- пояснения к исследуемым системным таблицам и параметрам.

3. Краткие ответы на те контрольные вопросы, которые ещё не нашли своего отражения в отчете.

Вопросы для самопроверки

1. Что входит в физический и логический формат магнитного диска?
2. Запишите физический адрес размещения загрузочной записи.
3. Какова структура загрузочной записи диска.
4. Каково содержание записи в таблице каталога.
5. Поясните принципы построения FAT – таблицы.
6. Поясните назначение (.) и (..) в отдельных записях каталогов.
7. Что показывает номер кластера в записи файла и каталога в корневом каталоге (дочерних каталогах)?
8. Как изменяется запись о файле в каталоге при его удалении? Какие еще изменения происходят ?
9. Какая цифра записывается в FAT для обозначения последнего кластера в списке индексов?
10. Сформулируйте последовательность действий при восстановлении утраченной информации на магнитном диске.
11. Поясните, что собой представляет HEX-формат вывода на экран.
12. Каков диапазон номеров кластеров, используемый для представления логической структуры дискеты 3,5"? Какая файловая система создается при ее форматировании?
13. Найдите коды окончания строки.
14. Что и где записывается на диск при сохранении, создании, удалении, закрытии файла?
15. Сколько кластеров на диске A:\?

Актуальность занятия

1. Файловая система управляет организацией и размещением файлов в ОС, определяет эффективность доступа и объемы хранимой информации на магнитных носителях. Задачи, решаемые файловой системой – основные задачи ОС.

2. Умения работы с системными ресурсами файловой системы – это возможность сохранения и восстановления по каким-либо причинам утраченной информации.

3. Файловая система FAT – это пример широко применяемой в настоящее время файловой системы. Изучение ее структуры поможет глубже понять принципы построения ОС и логику выполнения основных ее операций.

4. Занятие имеет явно выраженную практическую направленность и связано с решением задачи сохранения информации и использования типовых системных средств.

Задачи файловой системы

Основная задача ОС в части организации хранения данных на внешних устройствах – удобство обмена с диском. Для этого ОС подменяет физическую структуру диска и данных некоторой удобной пользователю логической моделью. Такой моделью является – дерево каталогов с символическими именами (рис.1.1). Файлами в терминах этой модели оперируют так называемые менеджеры типа Norton Commander, Мой компьютер, Проводник и т.д. Файловая система (ФС) при этом обеспечивает долговременное хранение информации и совместное использование данных.

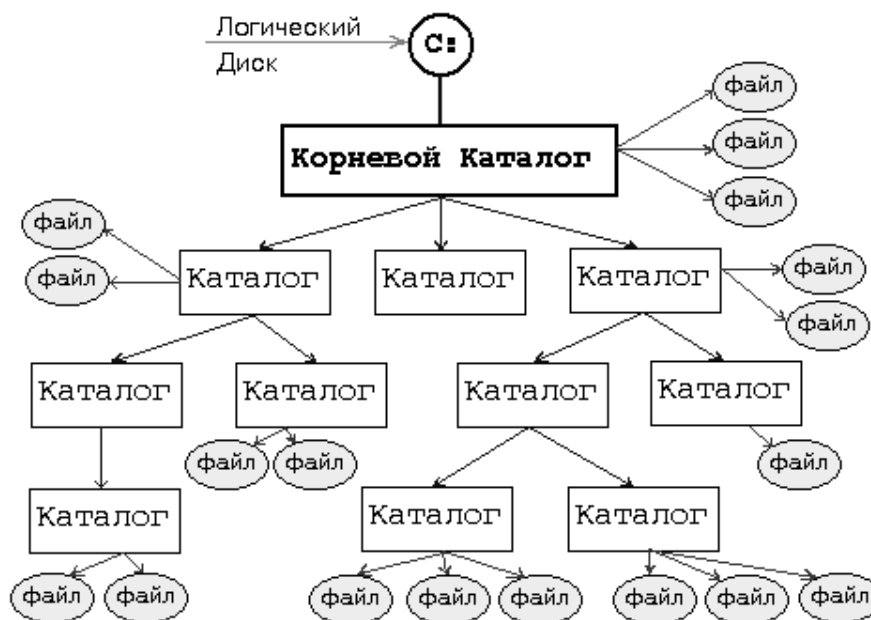


Рис.1.1. Логическая модель ФС

Файловая система включает:

- совокупность всех файлов на диске, непосредственно содержащих хранящиеся данные;
- набор системных структур для управления ими (каталоги, дескрипторы, таблицы распределения свободного и занятого дискового пространства);
- комплекс системных программных средств реализации файловых операций (создание файла, удаление, чтение, запись, переименование, поиск и т.д.). Эти программы являются неотъемлемой частью ОС.

Физическая структура магнитного диска

Физическая организация диска – это принципы размещения файлов, каталогов и системной информации на устройстве.

Основными критериями эффективности физической организации файлов являются:

- скорость доступа к данным;
- объем адресной информации файла;
- степень фрагментации дискового пространства;
- максимально возможный объем файла.

Дисковой накопитель – устройство, предназначенное для записи и чтения данных, как правило, на магнитных дисках. Сами диски (их может быть несколько в одном пакете) представляют собой стеклянную, металлическую или лавсановую основу с нанесенным магнитным слоем. Конструктивно выделяют следующие понятия.

Поверхность (сторона) диска – образуется магнитным слоем с одной стороны диска. Нумеруется, начиная с нуля. Каждый диск, соответственно, включает две поверхности.

Магнитная головка – устройство, образованное катушкой с сердечником, предназначенное для записи и чтения данных. Количество головок соответствует числу поверхностей. В едином блоке все головки позиционируются с шагом на одну дорожку по радиусу шаговым двигателем. Такая конструкция сокращает время поиска данных.

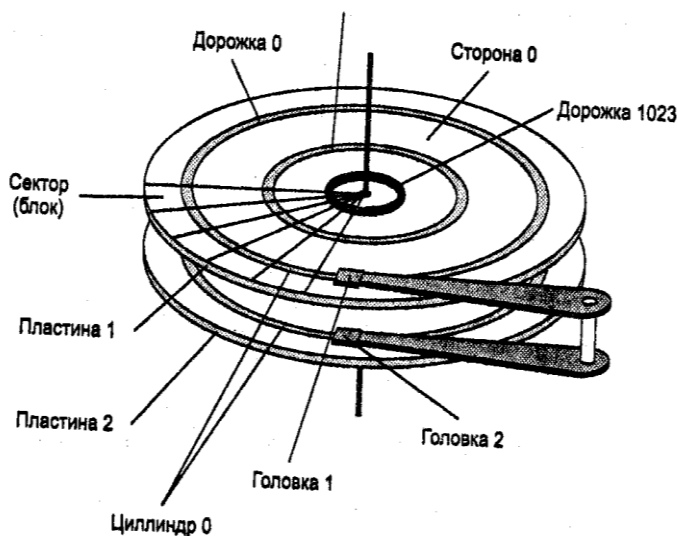


Рис.1.2. Организация дискового накопителя

Дорожка - tracks – концентрические кольца, образуемые головкой на магнитном слое. Нумерация производится от внешнего края к центру, начиная с нуля.

Цилиндр – все дорожки одного радиуса на всех поверхностях всех пластин (дисков). Цилиндры нумеруются, соответственно, так же как и дорожки. Для накопителей на гибких магнитных дисках (НГМД) количество цилиндров порядка 80,

для жестких магнитных дисков (НЖМД) – десятки тысяч.

Сектор (блок) – фрагмент дорожки. Все дорожки имеют равное число секторов (исключения бывают для начальных и используются как резерв) равного размера, объемом, обычно, 512 байт. Нумеруются сектора, начиная с единицы. Плотность записи для дискет 1,44" 18 секторов. Сектор является наименьшей адресуемой единицей обмена диска с оперативной памятью (ОП). Этими категориями оперируют драйверы при работе с накопителями.

Физический адрес сектора образуется номером цилиндра, поверхности и сектора. Адрес самого первого сектора, таким образом, 0-0-1.

Количество сторон и дорожек – это аппаратные характеристики устройств. Дорожки и секторы, их количество и размер – создаются программно при низкоуровневом (физическом) форматировании. Тогда же производится запись идентификационной информации. Этот формат от типа ОС не зависит.

Физический формат включает параметры:

- размер сектора в байтах;
- количество секторов на дорожке;
- количество дорожек на поверхности;
- количество поверхностей.

Кластер – несколько секторов, образующих единую единицу дискового пространства. Кластеры введены для сокращения объема адресной информации и сокращения времени поиска. Их размеры определяются объемом накопителя и колеблются от 2 до 32К.

Способы размещения файла на диске

1. Непрерывными участками дисковой памяти. Такой способ обеспечивает высокую скорость, минимум адресной информации (хранить необходимо только адрес начала и объем файла). Однако имеет высокую фрагментацию и совершенно непредсказуемую реакцию на увеличение размера файла. Тем не менее, такой способ находит применение в системах реального времени.

2. Связанным списком кластеров. В этом случае предполагается, что в начале каждого кластера помимо самих данных имеется еще указатель на следующий кластер. Адрес файла задается при этом только номером первого кластера списка. Списочная структура предполагает объединение произвольных кластеров, поэтому фрагментация исключается, становится допустимым увеличение размеров файла. Однако перемещение по файлу предполагает чтение самих кластеров, а это увеличивает время поиска нужного фрагмента. Кроме того, объем данных, хранящихся на каждом кластере не кратен двум, что усложняет расчеты объема памяти.

3. Связанным списком индексов. Файлу также выделяется область в виде связанного списка кластеров, при этом с каждым кластером связывают его номер (индекс). Номер первого кластера запоминается в записи каталога, где хранятся характеристики этого файла. Индексы хранят в другой таблице, размер которой соответствует количеству кластеров на диске. Каждый элемент второй таблицы является ссылкой на следующий кластер файла. Получается список (последовательность) индексов кластеров, хранящих данный файл.

Различают следующие по смыслу коды в таблице кластеров:

- 0 - свободный кластер (в нем можно сохранять данные);
- FFFFh – последний кластер списка;

- FFF7h – поврежденный кластер (bad – плохой - в нем система ничего не размещает. Такие кластеры выявляются при форматировании);
- зарезервирован для специальных целей.
- № следующего кластера (в состав этих кодов включаются все коды кроме ранее названных и имеющих специальное назначение).

Такая таблица представляет собой карту распределения дискового пространства на уровне кластеров. Ее называют ***FAT-таблица*** – File Allocation Table (таблица размещения файлов). Такой способ наиболее популярен, потому что предполагает минимум адресной информации, не влечет за собой фрагментацию, не имеет проблем при изменении размера файла, достаточно эффективен при поиске (читается только таблица индексов, а не все сами кластеры), сам кластер заполняется целиком и его объем кратен двум.

4. **Перечислением номеров кластеров** файла. Способ предполагает, что размещение файла задается перечислением номеров занятых им кластеров. Способ имеет максимальную скорость доступа, но длина самой записи зависит от размера файла, что вызывает нерегулярность управляющей информации.

Логическая организация диска

Прежде, чем диск будет отформатирован под конкретную ФС, он может быть разбит на разделы, которые могут использоваться либо одной, либо различными ОС.

Раздел - это непрерывная часть физического диска, которую ОС представляет пользователю как логическое устройство. Работа с ним осуществляется как с отдельным диском по символьным именам А, В, С, D Каждый раздел может быть отформатирован в соответствии со своей файловой системой.

Разметка диска под конкретную ОС выполняется в результате высокоуровневого логического форматирования. При этом определяется размер ***кластера***, которым оперирует ОС. Место на диске файлам выделяется кластерами. Все разделы одного диска имеют одинаковый размер сектора, но различные логические диски могут иметь различный размер кластера.

ОС может поддерживать различные статусы разделов, отмечая разделы, который могут быть использованы для загрузки модулей ОС, и разделы, в которых можно только хранить файлы. Разделы диска могут быть двух типов - primary и extended. На диске обязательно должен быть хотя бы один primary-раздел (загружаемый). Если таких разделов несколько, только один из них может быть активным. Загрузчику, расположенному в активном разделе, передается управление при загрузке ОС. Остальные primary-разделы считаются скрытыми. На жестком диске может быть единственный extended - раздел, который может быть разделен на большое число подразделов - логических дисков.

Логический формат включает данные о:

- нумерации сторон, дорожек, секторов;

- размещении специальных программ;
- размещение специальных таблиц для работы с диском;
- размере кластера.

А) структура логического диска

Диск начинается с **загрузочной записи** (рис.1.3), которая всегда расположена по самому первому адресу (самый 1-й сектор): 0-0-1. В настоящее время размер этой записи может быть увеличен до нескольких секторов. Но всегда обращение к ней при загрузке осуществляется по физическому адресу.



Рис.1.3. Структура логического диска

ров (Windows 95\98 и выше).

Далее следует корневой каталог, размер которого ограничен, но положение заранее известно. Он существует с таким статусом для логического диска в единственном экземпляре. Каталог также размещается после загрузки ОС в ОЗУ. В отличие от остальных каталогов, которые создаются с помощью специальных команд MS DOS, корневой каталог создается при форматировании, но может редактироваться системными командами.

И только после системной области размещаются данные, которые существуют на накопителе в виде файлов. Количество каталогов и файлов области данных не ограничено.

Других на этот момент просто нет.

После секторов, отведенных под запись начальной загрузки, располагаются две копии FAT- таблиц. Дублирование таблиц произведено в целях повышения надежности хранения информации. Таблицы начинаются с сектора 2, если не оговорено другое количество зарезервированных секторов. Размер FAT и число копий определено в параметрической информации (табл.1). FAT переписывается в ОЗУ чтобы увеличить быстродействие, но это уменьшает свободное адресное пространство.

Размер FAT зависит от размера логического диска. В зависимости от количества кластеров на диске (и, соответственно, числа разрядов, отводимых для задания номеров кластеров) различают таблицы с 12, 16 и 32 разрядными ячейками. Соответственно, таблицы называются FAT-12, FAT-16 и FAT-32. При 32 разрядах получается более 4 млрд кластеров

**Б) состав загрузочной записи
(записи начальной загрузки)**

В составе загрузочной записи выделяются четыре характерные области (рис.1.4):

- команда безусловного перехода на начало загрузчика;

- параметрическая информация (disk parameter block)– для идентификации физического и логического формата логического диска. До чтения этой информации о диске при загрузке ничего неизвестно: размещение системной информации, размеры таблиц, тип файловой системы и т.д. Состав параметрической информации представлен в табл.1 поля по адресам от 03h до 3Eh;

- **программа начальной загрузки**, которая запускается BIOS после успешной перезаписи загрузочной записи в ОП. Ее размер не превышает 512 байт и ее целью является загрузка следующей программы, стартового сектора ОС с накопителя.

- сигнатура загрузочной записи. При наличии в этой области кода 0AA55h система считает диск системным и продолжает выполнение загрузки ОС.

Безусловный переход JMP 3Eh
Параметрическая информация
Программа начальной загрузки
Сигнатура 0AA55h

Рис.1.4. Состав загрузочной записи

Таблица 1

Структура загрузочной записи Boot Record для FAT16

Смещение поля, байт	Длина поля, байт	Содержимое поля
00H(0)	3	Безусловный переход JMP 3EH
03H(3)	8	Системный идентификатор (например, DOS 3/1)
0BH(11)	2	Размер сектора, байт (например, 512=200h)
0DH(13)	1	Число секторов в кластере
0EH(14)	2	Число зарезервированных секторов в начале диска (если загрузочная запись занимает более 1 сектора)
10H(16)	1	Число копий FAT
11H(17)	2	Максимальное число элементов основного каталога (под элементом понимается одна полная запись длиной 32 байта)
13H(19)	2	Число секторов на логическом диске, если его размер не превышает 32 Мбайт; иначе

		0000H
15H(21)	1	Идентификатор формата
16H(22)	2	Размер FAT, секторов
18H(24)	2	Число секторов на дорожке
1AH(26)	2	Число рабочих поверхностей
1CH(28)	4	Число скрытых секторов
20H(32)	4	Число секторов на логическом диске, если его размер превышает 32 Мбайт
24H(36)	1	Тип логического диска (00H-гибкий, 80H-жесткий)
25H(37)	1	Резерв
26H(38)	1	Маркер с кодом 29H
27H(39)	4	Серийный номер тома
2BH(43)	11	Метка тома
36H(54)	8	Имя файловой системы
3EH(62)		Загрузчик (System bootstrap)
1FEH(510)	2	Сигнатура (слово AA55H)

В) структура каталога FAT-16 (FAT-12)

Каталог, в том числе и корневой каталог, представляет собой таблицу с записями (элементами) о файлах и других каталогах, которые являются для него дочерними (подкаталогами). Каждому подкаталогу или файлу отводится одна запись размером 32 байта. После загрузки корневой каталог находится в ОП постоянно, что и определяет формирование пути с началом именно от него. Из всех остальных каталогов в памяти находится только активный (текущий) каталог. Структура каталога представлена в табл.2.

Имя файла занимает 8 байт. Если имя содержит менее 8 символов, то поле дополняется справа пустыми позициями (пробелами – код 20h). Операционная система преобразует буквы любого регистра в прописные, поэтому алфавит при вводе имени значения не имеет. Первый байт поля используется для обозначения трех специальных случаев:

1. 00h – элемент каталога никогда не был использован. При последовательном заполнении каталога это означает, что дальше записей просто нет и поиск там производить не надо;
2. 0E5h – заносится при стирании файла (каталога) без изменения остальных элементов записи. Это дает возможность восстанавливать данные, но до выполнения записи в данный каталог. Этот код обозначает свободную строку каталога и при записи нового файла она может быть занята с перезаписью всех старых данных. При этом прежние сведения будут утрачены полностью. Стирание, таким образом, не есть уничтожение самих данных до перезаписи;
3. 2Eh – символ "." - элемент для описания самого каталога. В случае его активности по этим адресам осуществляется сохранение новой записи. Если и второй элемент содержит 2Eh – ".." – элемент описы-

вает родительский каталог для подъема по дереву логической модели.

Таблица 2

Структура каталога FAT-16 (FAT-12)

32 байта

Имя файла	Расширение	Атрибуты Файла	Резерв
8	3	1	4

Резерв	Время	Дата	№ 1-го кластера	Размер файла, байт
6	2	2	2	4

Атрибуты:

1 - каталог	1 - только для чтения
1 - архивный	1 - скрытый
0 - не используется	1 - системный
0 - не используется	1 - метка тома

Атрибуты файла – 1 байт: каждый бит байта задает конкретный атрибут, если в нем сохранена единица. В табл.2 указано назначение каждого разряда байта атрибутов в соответствии с его положением. Например, при значении атрибута 01h – файл допустимо только читать, но не производить запись в него. При атрибуте 10h – файл является каталогом, и его система читает и выводит на экран в формате каталога. Это атрибут является для файла типа каталога неким аналогом расширения для обычных файлов. Таким образом, ОС поступает с файлом в зависимости от атрибутов особым образом. Файл может иметь несколько атрибутов одновременно.

Номер первого кластера – 2 байта – служит указателем на первый кластер файла в области данных и одновременно указывает начало цепочки элементов FAT этого файла. В каждом элементе FAT для этого файла записывается номер следующего кластера. Цепочка заканчивается кластером, для которого в FAT размещен код 0FFFFh. Соответствующая цепочке номеров индексов последовательность кластеров на накопителе является последовательностью кластеров, в которых хранятся данные этого файла.

Размер файла (в байтах) - 4 байта. Обычно ОС распознает конец файла по информации из каталога. В свойствах файла, соответственно, указывается две цифры размера файла в байтах: одна – фактическая, вторая – размер отведенной для файла внешней памяти, определяемой по числу задействованных кластеров, что всегда больше (очень редко) равно фактической.

Файловые операции

Работа с файлом предусматривает две группы операций:

- 1) непосредственная обработка данных: создание, удаление, копирование и т.д.;
- 2) специальные операции для всех файлов (системные операции):
 - поиск по имени файла характеристик файла в ФС;
 - копирование характеристик файла в ОП;
 - проверка прав пользователя на выполнение запрошенной операции (чтение, запись, удаление, просмотр атрибутов);
 - освобождение памяти от временно хранимых характеристик файла.

Варианты выполнения операций над файлами:

1) без запоминания состояния операций: при каждой работе с данными каждый раз выполняются как специальные, так и уникальные операции. Такой способ более устойчив к сбоям, т.к. каждая цепочка операций не зависит от предыдущей. Поэтому способ иногда применяется в распределенных сетевых файловых системах (Network File System);

2) Специальные операции выполняются только в начале и конце всех работ с файлом. Такой способ поддерживают большинство файловых систем как более экономичный и быстрый. Для такого способа вводят специальные системные вызовы: операции OPEN (открыть) и CLOSE (закрыть). Основной задачей Open вызова является преобразование символического имени файла в его числовой адрес, копирование характеристик с диска в буфер ОП и проверка прав пользователя на выполнение операции. Наоборот Close освобождает буфер и делает невозможным выполнение операций с файлом. Такие приемы применялись при программировании в среде Turbo Pascal.

Процедура обращения к файлу в общем случае состоит из следующих операций:

- создание файла с заданным именем в указанном каталоге или открытие файла, если он был создан ранее;
- запись в файл или чтение из файла всего содержимого либо его части;
- закрытие файла.

Уровни обращения к магнитным дискам

1). Нижний. Предполагает непосредственное обращение к программам управления диском через прерывание 13h BIOS. При этом выполняются операции типа чтение/запись сектора (функции 02/03), позиционирование головки по дорожкам, форматирование дорожки (функция 05) и т.д. Файловая система при этом не используется, обращение осуществляется по физическим адресам (параметрам): цилиндр-поверхность-сектор;

2). Верхний уровень. Реализуется функциями DOS (прерывание 21h) для обслуживания файловой системы. Работа при этом осуществляется на уровне файловой системы и параметрами являются логический диск, каталог,

имя файла, путь. Типовыми операциями являются: создание каталога (функция 39h), удаление каталога (3Ah), смена каталога (3Bh), создание файла (3Ch), открытие файла (3Dh), закрытие файла (3Eh) и т.д.

Типы файлов:

1). Обычные – содержание которых определяется создающим их приложением: doc, xls, txt, pdf и др. ОС должна распознавать зарегистрированные типы файлов и открывать их средствами соответствующих приложений. Сюда же относятся исполняемые файлы, которые запускаются на исполнение (расширение типа exe или com) или обрабатываются командным процессором (bat);

2). Каталоги – специальный тип файла, содержащий справочную информацию о наборе файлов (подкаталогов), сгруппированных пользователем в этом каталоге, и дополнительную служебную информацию о своем размещении и родительской вершине. Структура такого файла соответствует табл.2. По сути именно каталоги устанавливают соответствие между именами файлов и их характеристиками, используемыми ФС для управления файлами. Каталоги ОС распознает как свои собственные файлы и поступает с ними по назначению, в соответствии с логикой работы файловой системы. Например, копирование каталога приводит к копированию самой таблицы каталога и всех файлов и подкаталогов, включенных в этот каталог. Это соответствует логическому определению каталога как контейнера для размещения файлов и подкаталогов. Во всех остальных отношениях каталоги рассматриваются файловой системой как обычные файлы, хранящиеся на кластерах в области данных диска.

Таким образом, Каталог = файл = перечень всех подкаталогов и файлов этого каталога;

3). Специальные – фиктивные файлы, ассоциированные с УВВ для унификации операций ввода-вывода. Это позволяет пользователю выполнять операции ввода и вывода посредством обычных команд записи в файл или чтения из файла. Команды обрабатываются файловой системой и на некотором этапе выполнения запроса преобразуются ОС в команды управления соответствующим внешним устройством. К специальным относятся и некоторые другие особые файлы, например, файлы подкачки.

Выполнение операций с файлами базируется на использовании **SFT** – **System File Table** – таблица открытых файлов. Это специальная системная таблица, формируемая в ОП среди системных областей данных. Объем этой таблицы определяет максимальное число одновременно открытых файлов. Элемент SFT (блок описания файла) образуется для открываемого файла и в него заносятся данные: имя, длина, атрибуты, дата и время создания, номер первого кластера, физический адрес на диске записи каталога об этом файле и др. Часть данных копируется из каталога, другая часть - создается самой ОС. Среди таких данных необходимо выделить:

- **дескрипторы** (файловые индексы, файловые описатели) – некие номера открытых файлов, которые закрепляются за сформированным блоком описания файла. Количество допустимых номеров ограничено, поэтому после закрытия файла цифра высвобождается и будет присвоена другому открываемому файлу. Обращение к открытому файлу осуществляется по дескриптору как по имени. Дескрипторы имеют только открытые файлы. По мере работы с файлом система модифицирует информацию в блоке SFT, поддерживая ее всегда в актуальном состоянии (всегда отражает текущее состояние файла);

- **указатель файла** – номер байта относительно начала файла, с которого начинается очередная операция записи или чтения. Указатель позволяет чтение или запись выполнять с любого места файла (организовать прямой доступ к файлу).

Выполнение файловых операций

Создание файла (нового)

1) ОС ищет 1-й свободный кластер на диске, просматривая FAT-таблицу. Обычно она это делает для увеличения быстродействия с текущего номера кластера, на котором закончилась предыдущая операция. Признаком свободного кластера является 00 в FAT-таблице.

2) ОС заполняет полностью новую запись в таблице активного (указанного) каталога, занося туда номер первого найденного свободного кластера и размер файла. Если файл создается пустой, то в FAT-таблице в ячейку с номером найденного кластера заносится код конца файла – 0FFFFh.

Открытие файла

Операция выполняется перед началом любых действий с файлом.

- 1) Проверка прав пользователя;
- 2) Файлу назначается очередной свободный блок таблицы SFT (закрепляется дескриптор);
- 3) Формируются параметры блока копированием записи об этом файле в таблице каталогов;
- 4) Указатель устанавливается на нуль.

Чтение и работа с файлом

Операция выполняется только с открытыми файлами, имеющими дескриптор. Модифицируется в процессе работы таблица открытых файлов SFT, в частности, объем файла и указатель. Сам файл до сохранения накапливается в буфере соответствующего приложения.

Сохранение (запись) файла на носителе

При выполнении операции сохранения параметры блока SFT переносятся в таблицу каталога, если файл уже существовал. При этом выполняется поиск недостающих свободных кластеров и образование цепочки индексов в FAT-таблице. Формируется длина файла, дата изменения и прочие необходимые параметры. Если файл сохраняется впервые, то первоначально выполняются операции аналогичные созданию файла.

Закрытие файла

Операция выполняется после окончания работы с файлом. При этом:

- 1) Перезаписывается для закрываемого файла в таблице каталогов дата, время;
- 2) Закрываются все файлы, буферы, которые были открыты или созданы этим файлом;
- 3) Освобождается блок в таблице SFT (утрачивается дескриптор). Дальнейшие операции с файлом становятся невозможными.

Удаление файла

При удалении файла в первый байт 32-байтной записи каталога (см. табл.2) заносится специальный признак "запись свободна" – код 0E5h. Это изменение соответствует изменению первого символа имени файла. Во все индексные указатели цепочки этого файла в FAT-таблице заносится код 0000h – "кластер свободен". Больше ничего не выполняется. Стирание не есть еще уничтожение самих данных !!! В кластерах носителя, занятых ранее этим файлом, данные остались неизменными. Однако при открытии каталога, из которого файл удалялся, строка с первым кодом 0E5h считается пустой и на экране не показывается, хотя все данные о файле в таблице также остаются неизменными. Если в этот каталог размещается новый файл (подкаталог), то такие строки (с кодом 0E5h) являются первыми кандидатами для заполнения при последовательном просмотре строк сверху вниз, и данные о стертом файле в этом случае затираются (замещаются данными сохраняемого файла или подкаталога).

Восстановления стертого файла

Технология восстановления стертого файла может быть следующая:

1. Ничего не записывать на диск, т.к. это может привести к использованию кластеров с индексами 0000h, а это ведет к затиранию интересующих нас данных.
2. Исправить первый байт имени файла (код 0E5h на любой рисуемый символ, если имя восстановить трудно). Этого достаточно чтобы эта запись в

таблице каталогов начала отображалась на экране с помощью любого файлового менеджера;

3. В FAT-таблице читать последовательно кластеры с индексами 0000h, следующие за первым, номер которого сохранился в таблице каталогов. Количество кластеров должно соответствовать объему восстанавливаемого файла;

4. Каждый читаемый кластер должен быть оценен на принадлежность файлу. Проще это сделать для текстовых файлов, т.к. там можно распознать разумный текст. Если кластер принадлежит файлу, то этот номер должен быть включен в список индексов в FAT-таблице путем ее исправления. Делается присоединение путем внесения номера присоединяемого кластера по адресу ячейки FAT, соответствующей номеру первого кластера. В ячейке с номером присоединяемого кластера устанавливается код 0FFFFh. После этих операций цепочка кластеров состоит из двух кластеров, и она является завершенной. Такой файл уже можно читать в среде текстового редактора (но размер его может еще не соответствовать действительности). Операции следует продолжить до полного восстановления всех данных файла.

Вопрос 1.1. Исследование структуры системных ресурсов с помощью утилит

Запустите утилиту *System Information*. Изучите структуру системных ресурсов. К отчету представьте информацию о следующих ресурсах вашего компьютера:

- система;
- дисплей;
- память;
- диск;
- сеть.

Изучите возможности утилиты *Norton System Doctor*.

Norton System Doctor помогает оградить компьютер от проблем и обеспечить максимальную эффективность его работы. По умолчанию Norton System Doctor работает в фоновом режиме, непрерывно контролируя параметры работы ПК. Если требуется вмешательство пользователя, он немедленно выдает сигнал, но может устранять целый ряд проблем автоматически, не прерывая текущей работы.

Norton System Doctor использует датчики для контроля чуть ли не каждого параметра системы, в т.ч. дисков, памяти, ЦП и сети; выдает сигналы в тех случаях, когда значения контролируемых параметров становятся критическими; дает возможность настроить свою конфигурацию. System Doctor заставляет обратить внимание пользователя на критическое состояние системы до того, как возникнет серьезная проблема.

Датчики Norton System Doctor делятся на следующие основные типы:

- Датчики диска
- Датчики памяти
- Датчики кэша диска
- Другие датчики

В зависимости от типа выводимой информации в Norton System Doctor используются следующие три вида датчиков:

Графические датчики - обеспечивают вывод данных в реальном времени. Большинство датчиков Norton System Doctor относятся к этому типу. Графические датчики можно представлять в виде аналогового циферблата, аналоговой шкалы или гистограммы. Циферблат или шкала показывают только текущие значения, а гистограмма содержит целый ряд замеров, накопленных за некоторое время.

Датчики-светофоры - датчики Целостности диска, Данных образа диска и Готовности Rescue с установленной периодичностью контролируют систему и выдают результаты в виде зеленого, желтого или красного “сигналов светофора”. При появлении красного света на любом из трех этих датчиков Norton System Doctor может открывать соответствующую программу Norton Utilities для автоматического устранения проблем.

Датчики-календари - датчики Времени работы Windows и текущих даты и времени выдают календарь.

Датчики, контролирующие состояние дисков:

- Фрагментация диска
- Данные образа диска
- Целостность диска
- Место на диске
- Тест поверхности диска

Датчик **Фрагментация** диска показывает, насколько фрагментирован локальный диск. Он может автоматически запускать Speed Disk для дефрагментации места на диске или же выдавать сигнал в тех случаях, когда уровень фрагментации достигает определенного процента.

Датчик **Целостность диска** также сканирует диски. Этот датчик контролирует состояние локального диска. При обнаружении проблем он может автоматически запускать Norton Disk Doctor или извещать об этом пользователя. Аналогично Norton Disk Doctor, этот датчик обнаруживает ошибки в таблице размещения файлов, файлы с перекрестными ссылками и потерянные кластеры. В отличие от Norton Disk Doctor, этот датчик не проверяет загрузочный сектор, поверхность диска, таблицу разделов и сжатые диски.

Датчик **Место на диске** контролирует степень использования локального или сетевого диска, к которому есть доступ.

Датчик **Тест поверхности** диска сканирует поверхность локального диска. При обнаружении неполадок на поверхности он может автоматически запустить Norton Disk Doctor или сообщить о них пользователю.

Датчики, контролирующих использование памяти:

- Физическая память
- Виртуальная память

- Загрузка памяти
- Ресурсы GDI
- Ресурсы User
- Селекторы (16-разрядные)
- Загрузка кэша
- Нити
- Память DOS
- Размер файла подкачки
- Загрузка файла подкачки
- Открытые файлы

Датчики памяти обеспечивают контроль за использованием памяти в реальном времени. При открытии, работе и закрытии приложений можно проследить за тем, как изменяется объем свободной памяти. Таким образом, можно отследить, какие из приложений и в каких режимах потребляют больше всего памяти.

Селекторы (16-разрядные). Этот датчик фиксирует использование селекторов, необходимых 16-разрядным приложениям и процессам для управления памятью. Этот датчик настраивается на выдачу количества либо свободных, либо занятых селекторов.

Память DOS. Этот датчик контролирует наличие основной памяти. Следует иметь в виду, что память DOS — это не та же самая память, используемая в Windows 95 для открытия окон DOS. Память DOS занимают некоторые старые 16-разрядные программы. Этот датчик фиксирует объем памяти, доступный для таких распределений памяти.

Ресурсы GDI. Этот датчик измеряет использование ресурсов интерфейса графических устройств (GDI). Датчик настраивается на вывод доли либо занятых, либо свободных ресурсов GDI.

Загрузка памяти. Этот датчик контролирует комбинацию физической памяти (ОЗУ) и виртуальной памяти обеспечиваемой за счет файла подкачки Windows. Датчик настраивается на показ либо свободной, либо занятой памяти. Результат отображается в процентном отношении к общему объему физической плюс виртуальной памяти. Windows, прежде всего, использует большую часть свободной физической памяти. Если для приложений требуется еще память, то Windows создает на жестком диске файл подкачки. Чем больше места на диске, тем большим может быть файл подкачки.

Открытые файлы. Этот датчик контролирует количество открытых файлов на локальном жестком или сетевом диске, к которому есть доступ. В ходе работы приложения и процессы Windows открывают файлы. В Windows 95 больше нет ограничений на количество одновременно открытых файлов. Тем не менее, большое количество открытых файлов может снизить быстродействие системы. Этот датчик способствует анализу производительности данной машины и конкретных приложений (в особенности приложений с базами данных, поскольку они открывают помногу файлов). Большое количество одновременно открытых файлов может, например, объяснить низкую производительность кэша диска. Если вам необходимо, чтобы этот датчик

выдавал сигнал, то необходимо при установке порога срабатывания учитывать характер выполняемой работы.

Физическая память. Этот датчик контролирует использование ОЗУ. Приложения лучше работают с физической (ОЗУ), нежели с виртуальной памятью. Этот датчик поможет вам оптимизировать производительность ПК и принять решение об усилении аппаратных средств.

Нити. Этот датчик контролирует текущее количество работающих нитей. Эта величина выдается либо в численном виде, либо в процентном отношении от максимального замеренного количества нитей. Новые 32-разрядные приложения и 32-разрядный сегмент самой операционной системы могут разбивать свое выполнение на отдельные нити. Это свойство обеспечивает возможность максимального использования ресурсов ЦП, поскольку приложения, разделенные на множество нитей, работают быстрее и лучше реагируют на действия пользователя. Например, разбитое на нити приложение может предоставить пользователю возможность отменить длительную операцию (такую, как открытие объемного документа) или переключиться на другую задачу, пока проходит эта операция. Для операций высшего (таких, как реакция на клавиатуру или мышь) или низшего приоритета при работе в фоновом режиме (например, разбивка документа на страницы или подготовка его к печати) приложение может создавать отдельные нити. Тем не менее, большое количество одновременно работающих нитей приводит к повышенному потреблению памяти и замедлению работы ЦП.

Ресурсы User. Этот датчик контролирует использование ресурсов User. Датчик настраивается на выдачу либо свободных, либо занятых ресурсов User. Для работы приложений Windows необходимы свободные ресурсы User, ресурсы GDI и (для приложений Windows 3.x) селекторы. Если эти потребности не удовлетворяются, то становится невозможным дополнительно открыть другие приложения, а если работающие приложения исчерпают эти ресурсы, то выдаются сообщения о нехватке памяти, и приходится закрывать одно или несколько приложений.

Виртуальная память. Этот датчик контролирует долю свободного места на диске, в данный момент не используемую под временное хранение данных (в т.ч. место, зарезервированное для файла подкачки Windows и в данный момент не используемое).

Размер файла подкачки. Это датчик контролирует свободное место на диске, доступное для файла подкачки Windows. Windows динамически увеличивает и уменьшает это место на диске в зависимости от потребностей системы.

Датчики, обеспечивающие возможность контроля за работой кэша диска:

- Обращения в кэш
- Загрузка кэша
- Пропускная способность кэша

Эти датчики способствуют совершенствованию настройки ПК, анализу работы кэша и принятию решений по расширению возможностей техниче-

ских средств. Например, если объем кэша мал, процент обращения к кэшу низок, или он работает медленно, можно рассмотреть возможность расширения ОЗУ. В принципе, увеличение объема кэша увеличивает процент обращения к нему и его скорость. С другой стороны, использование кэша снижает объем ОЗУ, доступный для других приложений и процессов.

Датчик *Загрузка ЦП* измеряет время, затрачиваемое ЦП на выполнение команд, по отношению к свободному времени. Этот датчик может быть настроен на выдачу процента времени, в течение которого ЦП либо работает, либо простаивает. Если процент свободного времени ЦП мал или его нет совсем, это может означать, что приложения или другие процессы неэффективно расходуют ресурсы, или же открыто слишком много приложений. Этот датчик способствует контролю за работой системы, диагностированию неполадок и анализу работы конкретных приложений и процессов.

Датчик *Пропускная способность кэша* измеряет быстродействие обработки запросов на считывание данных с диска. При этом учитываются как быстродействие системной памяти так и скорость обращения к диску. Датчик может быть настроен на выдачу результатов в мегабайтах, килобайтах или байтах в секунду. Чем выше показания датчика, тем эффективнее работает кэш диска. Низкая пропускная способность кэша может быть вызвана следующими обстоятельствами: большое количество процессов одновременно запрашивающих данные, высокая степень фрагментации данных на диске, малый объем кэша, недостаточный объем ОЗУ в системе, низкое быстродействие дисководов.

Датчик *Загрузка кэша* контролирует использование кэша диска. Его можно настроить на показ свободной или занятой в данный момент доли кэша. Результат может быть выражен в мегабайтах, килобайтах, байтах или в процентном отношении от максимального измеренного объема.

Датчик *Обращения в кэш* контролирует, сколько данных, необходимых программным средствам, получено из кэша диска а не непосредственно с диска. "Обращение в кэш" означает, что кэш содержит необходимые данные, а "неудачное обращение в кэш" — что данные пришлось отыскивать на диске. Низкий процент обращений в кэш может быть обусловлен одним из следующих обстоятельств: большое количество процессов одновременно запрашивающих данные, высокая степень фрагментации диска, малый размер кэша.

К отчету представить: краткую характеристику описанных датчиков в соответствии с оценками вашего компьютера на рабочем месте.

Вопрос 1.2. Исследование структуры магнитного диска

1. Изучить (по указанию преподавателя) структуру магнитного диска с помощью электронного учебника. Выполните следующие пункты задания, ответив на все выделенные жирным текстом вопросы.

! ВНИМАНИЕ !
РАБОТАТЬ ТОЛЬКО С ВЫДАННОЙ ДИСКЕТОЙ, на диске А !

2. Вставьте дискету и включите компьютер. Загрузка должна закончиться отображением панелей оболочки VC (аналогичны Norton Comander). Прочитайте задание для выполнения данного этапа исследований (NU\ZadLR6\txt). Используйте команду F3 (View – просмотр).

3. В каталоге NU создайте **свой каталог**. Для этого откройте каталог NU и воспользуйтесь командой F7 (Make Directory – создать директорию). Алфавит необходимо использовать латинский и длина имени должна быть не более 8 символов.

4. В созданном каталоге создайте **свой текстовый файл**. Для этого используйте команду Shift+F4. Текст должен помещаться на один экран. В тексте сформулируйте определения следующих понятий: дорожка, сектор, кластер, цилиндр, FAT-таблица, корневой каталог. Переход на русский алфавит – две клавиши Shift одновременно.

5. Запустите Нортоновскую утилиту Diskedit.exe из каталога NU. Снимите запрет на изменения данных, воспользовавшись подсказкой программы при запуске. Ответы на появляющиеся вопросы необходимо получать из соответствующей справки программы. Сформулируйте **назначение программы**.

6. Проанализируйте состав имеющихся каталогов на дискете и структуру записи в них. Выявите и исследуйте, затем законспектируйте **действия для выбора и просмотра**

- диска,
- каталога,
- файла,
- кластера,
- сектора.

Выбор объекта осуществляется в меню ОБЪЕКТ.

Найдите **отличия в записях корневого (и других) каталога о файлах и каталогах**. Запишите (или дополните свой лекционный материал) состав информации в каталоге. Сформулируйте назначение каждого поля **записи в каталоге**. Зафиксируйте в рабочей тетради записи каталога, соответствующие файлу настоящего задания и Вами созданного файла.

7. Исследуйте **структуру загрузочной записи**. Вывод ее на экран осуществляется через меню ОБЪЕКТ. Запишите схему операций для отображения загрузочной записи, структуру и параметры ее информационной части с учетом смещения каждого поля. Выясните смысл каждого из приведенных параметров. Найдите **адрес** расположения загрузочной записи на дискете.

8. Исследуйте содержимое 1-й копии **FAT-таблицы**. Ее отображение также осуществляется через меню ОБЪЕКТ. Занесите в отчет:

- **последовательность** операций для отображения FAT-таблицы;
- **список индексов** (кластеров), задействованных для размещения файла с текстом настоящего задания и созданного Вами файла;

- **список кластеров** для хранения каталога NU;
- посчитайте (или определите) **код**, соответствующий последнему кластеру файла. Занесите его в отчет;
- найдите **адрес** расположения двух копий FAT.

9. Прочитайте **содержимое первого кластера** Вами созданного файла. Убедитесь, что в нем содержится именно тот текст, который Вы набирали. Перенесите в рабочую тетрадь **первую строку данных** с экрана, отображающих начало вашего файла. Сформулируйте **назначение** каждого из полей экрана. По экрану определите **размер** кластера. Соответствует ли он параметрам в информационной части загрузочной записи?

Зафиксируйте **последовательность операций** для просмотра кластера, сектора.

Прочитайте в HEX-формате **содержимое физического сектора, в котором размещена загрузочная запись**. Найдите на экране данные, характеризующие размер сектора в байтах, число секторов в кластере, число копий FAT на диске, число секторов на дорожке. Сравните полученные данные с результатами просмотра информационной части загрузочной записи.

10. Закройте нортонскую утилиту и в оболочке VC удалите файл с настоящим заданием. Снова запустите Diskedit. Откройте каталог, в котором была запись об удаленном файле. Найдите те **изменения в записи**, которые произошли после удаления файла.

Проверьте **состояние цепочки кластеров**, которые были выделены для хранения удаленного файла. Проверьте **содержимое кластера**, в котором (в которых) был удаленный файл. Найдите **код**, по которому система определяет, что файл удален.

Запишите **этапы методики восстановления** удаленных с магнитного носителя данных и реализуйте ее. Не забывайте сохранять те изменения, которые Вы выполните с помощью утилиты.

Прочитайте содержимое получившегося после восстановления файла (средствами Diskedit или VC). Проверьте средствами VC восстановление ранее удаленного файла.

Контрольные вопросы (ПК-11):

1. Что входит в физический и логический формат магнитного диска?
2. Запишите физический адрес размещения загрузочной записи.
3. Какова структура загрузочной записи диска.
4. Каково содержание записи в таблице каталога.
5. Поясните принципы построения FAT – таблицы.
6. Поясните назначение (.) и (..) в отдельных записях каталогов.
7. Что показывает номер кластера в записи файла и каталога в корневом каталоге (дочерних каталогах)?
8. Как изменяется запись о файле в каталоге при его удалении? Какие еще изменения происходят ?

9. Какая цифра записывается в FAT для обозначения последнего кластера в списке индексов?
10. Сформулируйте последовательность действий при восстановлении утраченной информации на магнитном диске.
11. Поясните, что собой представляет HEX-формат вывода на экран.
12. Каков диапазон номеров кластеров, используемый для представления логической структуры дискеты 3,5"? Какая файловая система создается при ее форматировании?
13. Найдите коды окончания строки.
14. Что и где записывается на диск при сохранении, создании, удалении, закрытии файла?
15. Сколько кластеров на диске A:\?