

33333 ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

**Федеральное государственное образовательное бюджетное
учреждение высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»**

А.Н. Губин

СЕТИ ХРАНЕНИЯ ДАННЫХ

ПРАКТИКУМ

СПбГУТ)))

**САНКТ-ПЕТЕРБУРГ
2018**

УДК 004.43333

Рецензент

кандидат технических наук, доцент кафедры робототехники и автоматизации производственных систем Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» *А. В. Шевченко*

*Утверждено редакционно-издательским советом СПбГУТ
в качестве практикума по дисциплине "Сети хранения данных"*

Губин, А.Н.

Сети хранения данных: практикум / А.Н. Губин – СПб.: Изд-во СПбГУТ, 2018. - 76 с.

Рассматриваются основные типы задач, решение которых предусмотрено при изучении дисциплины сети хранения данных.

Учебное издание предназначено для бакалавров по направлению подготовки 09.03.02 – Информационные системы и технологии.

УДК 004.43

© Губин А.Н., 2018

© Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. Основные сведения необходимые для решения задач	5
1.1. Принципы организации хранения данных на жестком диске	5
1.2. Файловые системы	9
1.3. Оценка потери памяти при записи файлов	15
1.4. Производительность дисковых систем	16
1.5. Структура дисковых систем	20
2. Примеры решения задач и задания для самостоятельной работы	23
2.1. Задача 1	23
2.2. Задача 2	26
2.3. Задача 3	30
2.4. Задача 4	37
2.5. Задача 5	39
3. Методические указания к выполнению лабораторных работ	40
3.1. Лабораторная работа 1	40
3.2. Лабораторная работа 2	43
3.3. Лабораторная работа 3	46
3.4. Лабораторная работа 4	48
3.5. Примеры использования команд управления элементами СХД	50
ЗАКЛЮЧЕНИЕ	54
СПИСОК ЛИТЕРАТУРЫ	55

ВВЕДЕНИЕ

Хранение информации является важнейшей составляющей современных информационных технологий. Рост количества производящих контент устройств обуславливает существенный (экспоненциальный) рост объемов как структурированных (данные организованные в определенные структуры и строго форматированные по содержанию), так и неструктурированных данных (веб-страницы, изображения, аудио/видео потоки, тексты и др.). Возможность обработки данных предполагает их хранение и обеспечение доступности для приложений, использующих эти данные. Причем, в последнее время все большую актуальность приобретает возможность хранения и обработки неструктурированных данных.

Данные представляют собой некоторую совокупность фактов представленных в формализованном виде.

Традиционные базы данных, как правило, ориентированы на выполнение жестко определенных операций по обработке данных. Возможности обработки нерегламентированных запросов в таких базах данных существенно ограничены.

В то же время, необходимость выполнения по отношению к системе хранения данных таких операций как

- формирование запросов произвольной формы;
- интегрирование данных из различных генерирующих контент систем;
- обработка больших объемов данных,

вызвало появление новых технологий организации баз данных, а именно технологии хранилищ данных.

В основе концепции хранилища данных лежат две основные идеи:

- интеграция разьединенных детализированных данных в едином хранилище;
- разделение наборов данных и приложений, используемых для оперативной обработки и применяемых для решения задач анализа.

Определение понятия "хранилище данных"(Data Warehouse) первым дал Уильям Г. Инмон в 2002 году. Он определил хранилище данных как "предметно-ориентированную, интегрированную, содержащую исторические данные, не разрушаемую совокупность данных, предназначенную для поддержки принятия управленческих решений".

Система хранения данных (СХД) - это программно-аппаратное решение по организации надёжного хранения данных с использованием принципов построения хранилищ данных и предоставления к ним гарантированного доступа.

В данном учебном издании рассмотрены основные типы задач, решение которых предусмотрено при изучении дисциплины сети хранения данных.

При работе над практикумом автор неоднократно осуществляли заимствование из приведенных ниже источников определений, примеров и методов решений задач, с соответствующими ссылками в тексте.

1. Основные сведения необходимые для решения задач

1.1. Принципы организации хранения данных на жестком диске

Дисковое устройство в настоящее время является наиболее распространенным типом устройств, реализующих функции хранения, записи и считывания данных, используемых в современных вычислительных системах.

Дисковое устройство использует для записи/чтения данных быстро вращающийся плоский диск, покрытый магнитным материалом. Данные записываются на диск и считываются с диска через головку чтения/записи. Несколько собранных вместе магнитных дисков с блоком головок чтения/записи и устройством управления (контроллером) образуют устройство, которое получило название жесткого диска (рис.1).



Рис. 1. Общее устройство жесткого диска

На магнитных дисках (МД) информация записывается на концентрических дорожках, разделенных на секторы. Дорожка образуется как намагниченная узкая область поверхности МД (рис. 2).

Каждая дорожка делится на секторы. Сектор – дуга дорожки с фиксированным угловым размером.

Начальная разметка диска на дорожки и секторы производится на заводе изготовителе с помощью программы форматирования на низком уровне.

Совокупность дорожек, одновременно находящихся под блоком головок считывания, называется цилиндром.

Информационная емкость одного сектора равна 512 байт (на любой дорожке).

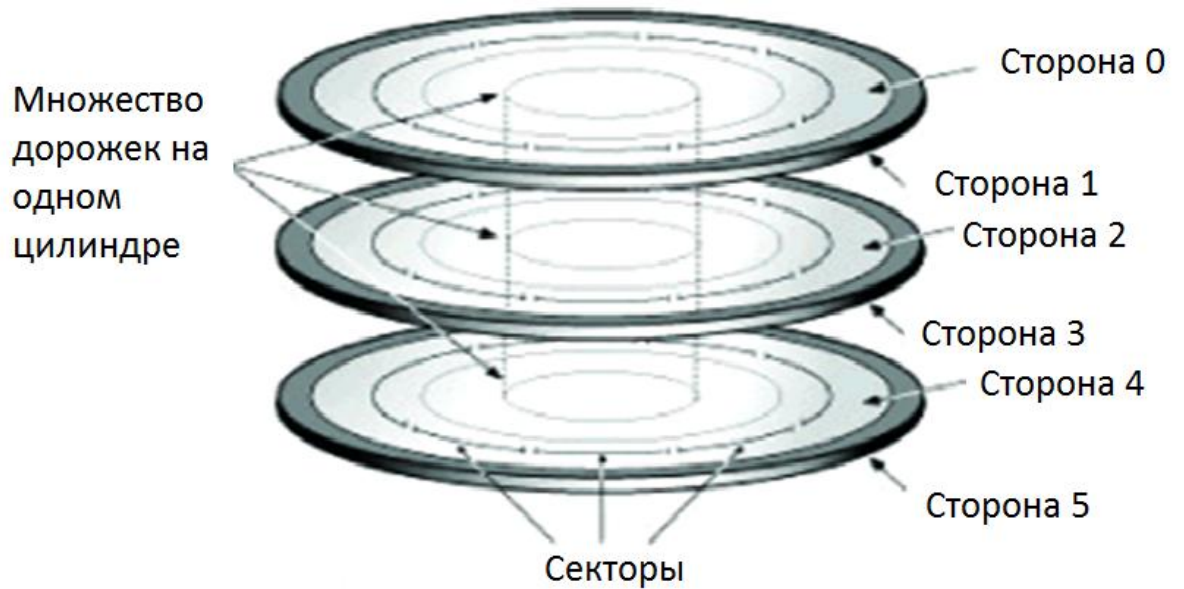


Рис. 2. Общая структура организации хранения информации на жестком диске

Вначале количество секторов на дорожках диска было одинаковым и следовательно плотность записи на них разная: чем ближе к центру, тем выше плотность записи (рис.3).



Рис. 3. Общая структура магнитного диска без зон

Пространство магнитного диска в этом случае используется неэффективно. Для повышения эффективности использования дискового пространства используют зонную запись секторов, которая предусматривает расположение секторов по зонам и выравнивание плотности записи информации по магнитным дорожкам (рис.4).

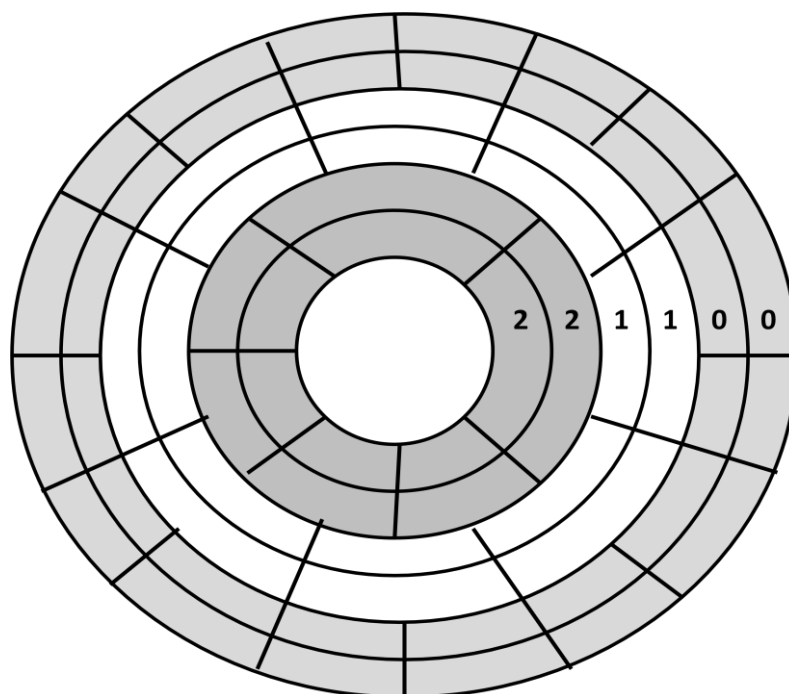


Рис. 4. Общая структура магнитного диска с зонами

Зоны нумеруются, начиная с крайней на диске, которая получает нулевой номер. Каждой зоне присваивается соответствующее количество секторов на магнитной дорожке. Все дорожки одной зоны имеют одинаковое количество секторов.

Каждый сектор состоит из двух частей. В первой части (по направлению вращения диска) записывается адрес сектора и дополнительная служебная информация (16 байт). Информационная часть сектора содержит фрагменты файлов и всегда содержит 512 байт (рис.5).

Адрес сектора – это три числа: < номер головки> <номер дорожки> <номер сектора> (ГДС).

Современные магнитные диски используют адресацию логических блоков (Logical block addressing -LBA).

Механизм LBA предусматривает адресацию и доступ к блоку данных на магнитном диске, когда контроллеру нет необходимости учитывать геометрию самого жесткого диска (количество цилиндров, сторон, секторов на цилиндре). При этом каждый блок, адресуемый на жёстком диске, имеет свой номер, целое число, начиная с нуля (то есть первый блок LBA=0, второй LBA=1, ...).



Рис. 5. Общая структура сектора на магнитном диске

Однако для осуществления физического доступа к любому блоку информации необходимо преобразовать номер блока в адрес сектора (ГДС). Преобразование осуществляется в соответствии со следующими выражениями:

$$Г = [(LBA - (C - 1)) / C_{кол.}] \bmod Г_{кол.},$$

$$Д = (LBA - (C - 1) - Г C_{кол.}) / C_{кол.} Г_{кол.},$$

$$C = (LBA \bmod C_{кол.}) + 1,$$

где $Г$ - номер головки, $Д$ - номер дорожки, $С$ - номер сектора, $C_{кол.}$ - количество секторов на дорожке, $Г_{кол.}$ - количество головок, mod - операция взятия остатка от деления.

Обратное преобразование адреса сектора в номер блока производится следующим образом

$$LBA = (Д Г_{кол.} + Г) C_{кол.} + C - 1.$$

1.2. Файловые системы

После форматирования МД на низком уровне диск становится доступным для записи и считывания информации. Такой доступ с использованием ГДС называется адресным.

Для обеспечения доступа пользователя к информации, записанной на МД в виде файлов, используется файловый доступ к МД. При файловом доступе пользователю достаточно знать только имя файла (его физическое расположение на МД пользователю неизвестно).

Преобразование адресного доступа в файловый доступ обеспечивает файловая система (рис.б).

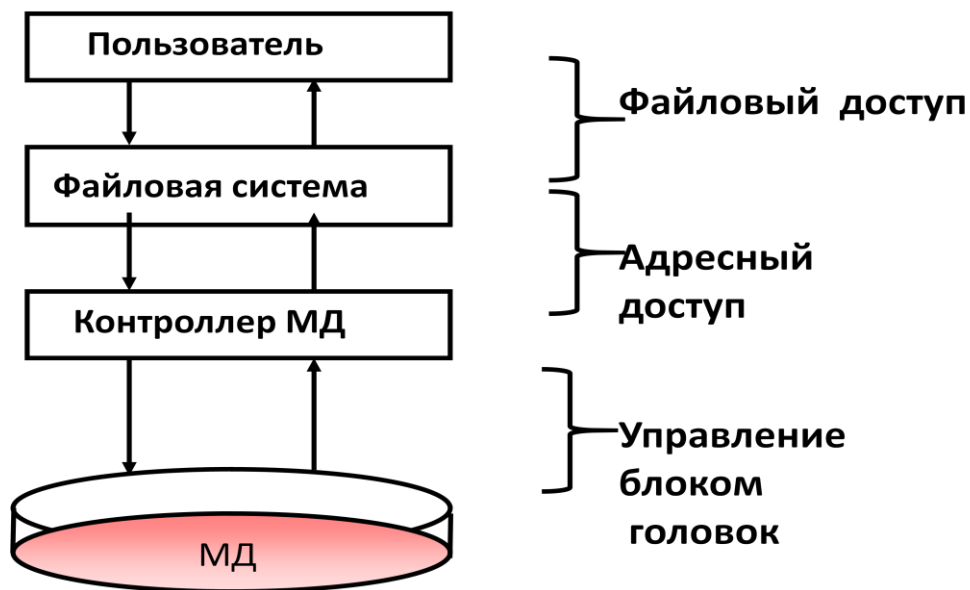


Рис. 6. Общая структура доступа к данным

Файловая система появляется на МД в результате выполнения операций высокоуровневого форматирования магнитного диска.

Базовым элементом файловой системы является файл.

Файл – это именованная область памяти, в которую могут записываться и из которой могут считываться данные.

При выполнении форматирования диска появляется новый параметр, который требуется задать или выбрать – это размер *кластера*.

Кластером называется группа расположенных подряд секторов (рис.7).

Объем кластера определяется количеством входящих в него секторов (определяется при форматировании) и может быть:

1(0,5К); 2(1К); 4(2К); 8(4К); 16(8К); 32(16К); 64(32К).

Все кластеры нумеруются в порядке натурального ряда чисел (на диске миллионы кластеров). Каждому номеру кластера соответствует адрес его начального сектора (Г-Д-С).

В реальных файловых системах набор ссылок на кластеры, которые составляют файл, обычно отрывается от элементов хранения информации и помещается в отдельную область диска в виде управляющей таблицы.

Такая таблица называется таблицей размещения файлов (File Allocation Table – FAT).

FAT состоит из такого количества ячеек – сколько кластеров размещено на диске.

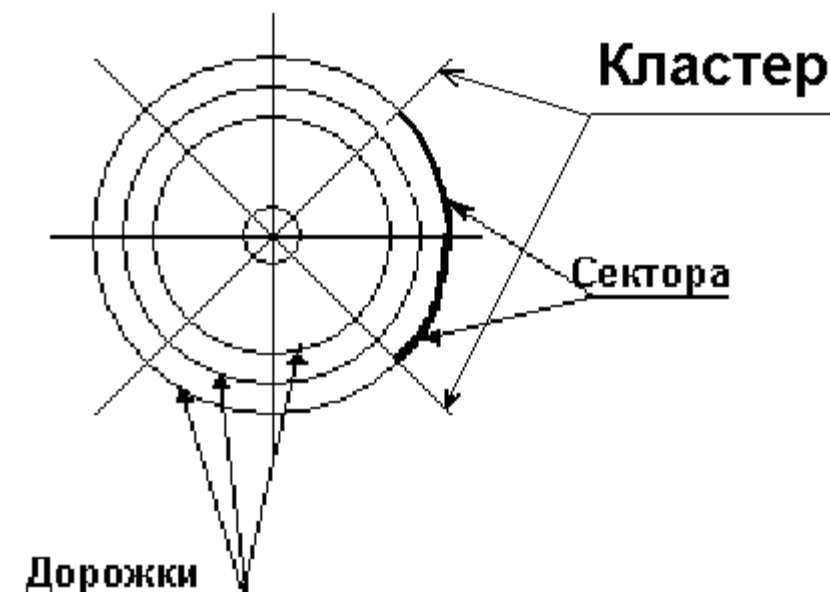


Рис. 7. Общая структура кластера на магнитном диске

На магнитном диске кроме таблицы FAT помещают таблицу, содержащую список имен файлов и точки входа этих файлов. Такая таблица называется директорией (каталогом, папкой).

Основное назначение каталога – указание начальных кластеров файлов (точек входа), их связь с именами файлов и определение атрибутов файлов.

При решении задачи рассматриваются варианты, когда для хранения данных используется файловая система FAT-32 и файловая система NTFS.

Файловая система FAT-32

FAT32 (File Allocation Table — «таблица размещения файлов») — это файловая система, разработанная компанией Microsoft.

FAT32 использует 32-разрядную адресацию (нумерацию) кластеров.

Старшие 4 бита каждого 32-битного значения адреса зарезервированы и не участвуют в формировании номера кластера, поэтому максимальное число кластеров в 32-разрядной FAT равно $2^{32-4} = 268435456$. За вычетом зарезервированных значений (10) максимальное количество кластеров, которыми может управлять FAT32, составит 10 268 435 445.

Независимо от своей природы и содержания все файлы представляют собой цепочки связанных кластеров (рис. 8). Расположение кластеров файла на диске пользователю неизвестно.

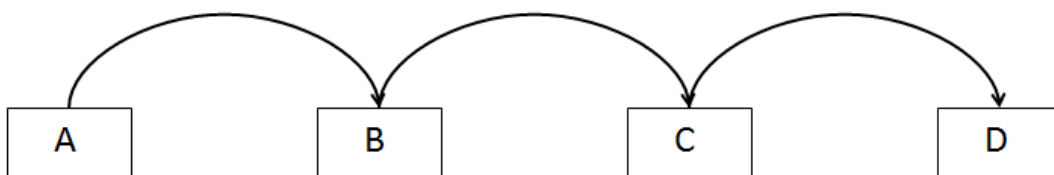


Рис. 8. Цепочка связанных кластеров

Для хранения таких данных обычно используют списковые структуры, при использовании, которых элементы данных могут располагаться в произвольных местах памяти, однако при этом каждый элемент, кроме собственно информации должен содержать и адрес связи со следующим элементом файла (рис. 9).

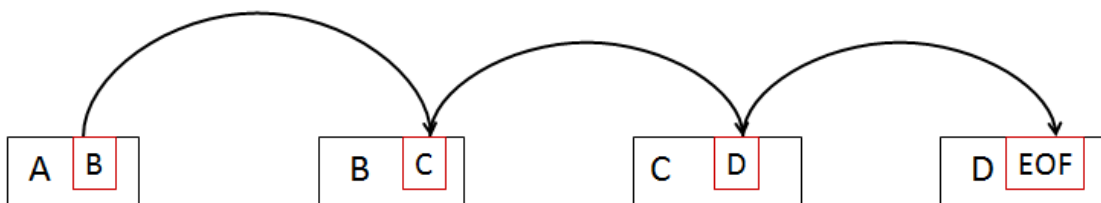


Рис. 9. Списковая структура элементов

Чтение списков происходит медленнее, чем чтение данных при использовании массивов. Увеличивается путь перемещения головок. Но операции удаления и добавления кластеров в состав файла при этом производится гораздо быстрее.

В реальных устройствах набор ссылок на кластеры, которые составляют файл, обычно отрывается от элементов хранения информации и помещается в отдельную (системную) область диска в виде управляющей таблицы.

Такая таблица называется таблицей размещения файлов (File Allocation Table – FAT).

FAT состоит из такого количества ячеек – сколько кластеров размещено на диске.

Номер каждой клетки таблицы (начиная с 1) соответствует номеру кластера, а содержимое клетки может быть следующим:

- 0 – кластер свободен и может быть включен в состав любого файла;
- Число - номер следующего кластера в цепочке, составляющей файл;
- EOF – конец файла (ссылки на следующий кластер нет);

BAD – плохой кластер, не подлежит распределению и использованию для записи информации.

Для файла $St=A \rightarrow B \rightarrow C \rightarrow D$ таблица расположения файла (FAT) может иметь следующий вид (рис. 10).

№ Клетки	199	200	201	202	203	204	205
Содержимое клетки	0	201	204	BAD	200	EOF	0

Рис. 10. FAT – таблица для файла $St=203 \rightarrow 200 \rightarrow 201 \rightarrow 204$

Чтобы система могла построить цепочку кластеров для считывания или записи файла, необходимо знать номер начального кластера этого файла (точку входа для файла).

Для хранения точек входа в файловой части диска помещают системную таблицу, содержащую список имен файлов и точки входа для этих файлов.

Такая таблица называется директорией (каталогом, папкой).

Основное назначение каталога – указание начальных кластеров файлов (точек входа) и их связь с именами файлов, кроме того в каталоге хранятся значения атрибутов файлов.

Табличную модель файловой системы FAT32 можно представить в виде, представленном на рис. 11.



Рис. 11. Табличная модель файловой системы FAT32

Файловая система NTFS

В файловой системе NTFS базовой единицей распределения дискового пространства является экстенда (отрезок).

Экстенда – непрерывная область кластеров. В качестве адреса экстенды используется логический номер первого кластера экстенды (*LCN*), кроме того, в качестве характеристики экстенды (ее объема) указывают количество кластеров в экстенде – число *k*.

Для определения порядка считывания экстент при формировании файла, используют виртуальный номер кластера – *VCN*.

Таким образом, полное описание экстенды представляется тремя числами (*VCN*, *LCN*, *k*)

Виртуальный номер кластера (*VCN*) - это порядковый номер кластера (экстенды) внутри файла. Эта информация определяет порядок считывания экстенды при формировании файла.

Логический номер кластера (*Logical Cluster Number, LCN*) - представляет собой порядковый номер кластера на диске, который используется для поиска начала экстенды на диске.

Количество кластеров в экстенде (*k*) – определяет, сколько кластеров необходимо считать для формирования экстенды.

Описание файла в NTFS задается простым перечислением экстент, входящих в состав файла. Так файл *St*, определенный в предыдущем разделе, в

терминах файловой системы NTFS может быть записан в виде следующей последовательности экстент:

$$St = \{1, 203, 1\}, \{2, 200, 2\}, \{3, 204, 1\}.$$

1.3. Оценка потери памяти при записи файлов

Потеря дисковой памяти при записи файлов обуславливается появлением незанятых участков памяти. Такие участки появляются вследствие деления памяти на единицы фиксированного размера (кластеры), в то время как объём файла не обязательно кратен этому делению. Таким образом, в среднем половина последнего кластера файла остается неиспользованной (рис.12). Это явление получило название внутренней фрагментации дисковой памяти.

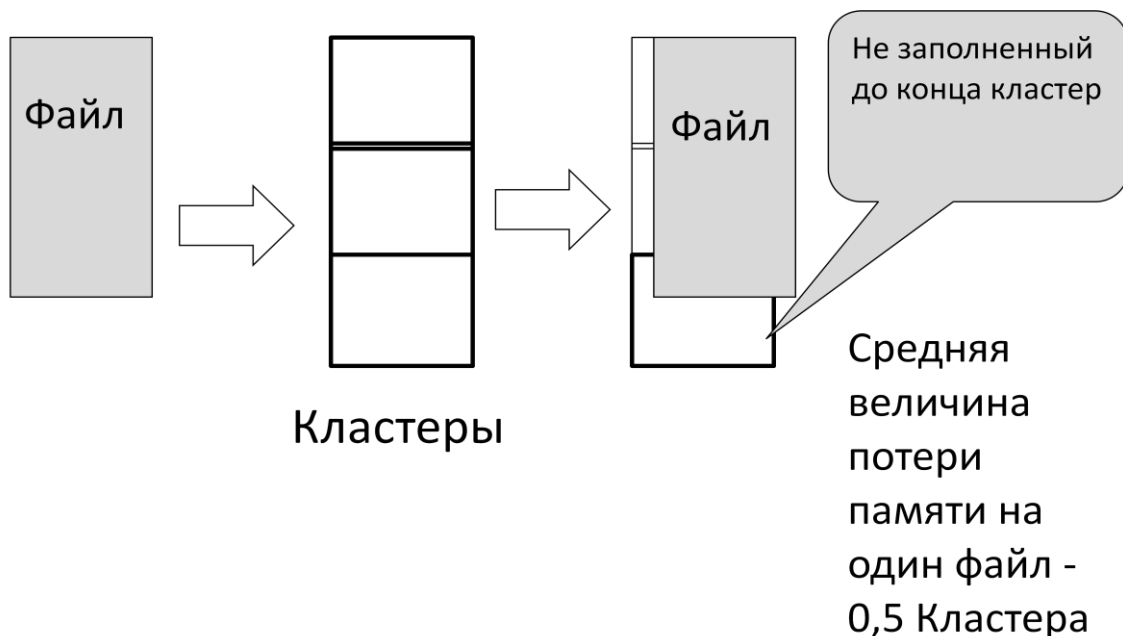


Рис. 12. Внутренняя дефрагментация дисковой памяти

Если размер файла меньше кластера, этот файл все равно занимает весь кластер. Оставшаяся часть кластера не может быть передана для использования в другом файле.

Для каждого файла потери дисковой памяти можно вычислить следующим образом. Определим количество кластеров, которые занимает файл объемом V_f , для чего к целой части результата деления объема файла на объем кластера прибавим единицу. Затем от произведения количества кластеров в

файле на объем кластера вычтем объем файла. Полученный результат будет показывать объем неиспользованной дисковой памяти.

$$N_{\text{кл}} = \left\lceil \frac{V_{\text{ф}}}{V_{\text{кл}}} \right\rceil + 1,$$

$$V_{\text{потерь}} = N_{\text{кл}} \cdot V_{\text{кл}} - V_{\text{ф}}.$$

1.4. Производительность дисковых систем

Для анализа производительности дисковой системы целесообразно представить ее в виде черного ящика на входе, которого формируется очередь из запросов на выполнение операций по вводу/выводу информации, а на выходе – поток обработанных запросов. Источником запросов на ввод/вывод информации является приложение, решающее основную задачу обработки данных (рис. 13).

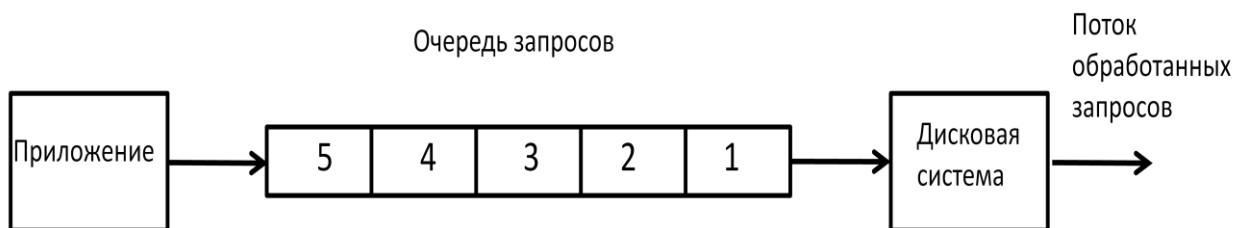


Рис. 13. Обработка запросов дисковой системой

Определим основные показатели производительности дисковой системы.

Запросы на выполнение операций ввода/вывода поступают на вход дисковой системы со скоростью работы приложения. Эта скорость называется частотой поступления.

Общее количество запросов в очереди и на обработке в дисковой системе - N определяется частотой поступления - a и средним временем отклика системы - R (общее время от прибытия запроса в очередь и до отправки обработанного запроса) в соответствии со следующим выражением

$$N = a \cdot R$$

При этом уровень загрузки дисковой системы - U определится как

$$U = a \cdot R_s,$$

где R_s - среднее время обслуживания запроса на ввод/вывод информации.

Уровень загрузки системы может рассматриваться как коэффициент использования дисковой системы, его значение варьируется от 0 до 1.

Если рассчитать среднее время между запросами как

$$R_a = \frac{1}{a},$$

Тогда уровень загрузки определится как

$$U = \frac{R_s}{R_a},$$

Среднюю скорость отклика системы можно определить как разницу между скоростью обслуживания запросов и скоростью поступления этих запросов в очередь.

Тогда среднее время отклика можно определить как величину обратную средней скорости отклика системы.

$$R = 1 / \left(\frac{1}{R_s} - \frac{1}{R_a} \right)$$

или

$$R = 1 / \left(\frac{1}{R_s} - a \right)$$

и далее

$$R = R_s / (1 - aR_s)$$

и

$$R = R_s / (1 - U)$$

Если загрузка дисковой системы стремится к 1 (дисковая система работает на максимуме загрузки), то время отклика стремится к бесконечности.

Количество запросов в очереди (средний размер очереди) можно определить как

$$N_Q = N - U$$

После преобразований N_Q определится как

$$N_Q = U^2(1 - U)$$

Достаточно важным параметром дисковой системы является значение времени, которое затрачивается запросом в очереди на обслуживание.

Это время определяется как разность среднего времени отклика системы и времени затраченного на обработку запроса. То есть,

$$R_Q = R_S / (1 - U) - R_S$$

или после преобразований

$$R_Q = UR$$

Рассмотрим работу дисковой системы, когда на ее вход поступают запросы на обслуживание со скоростью $a = 100$ запросов в секунду, а время обслуживания запроса составляет $R_S = 8$ миллисекунд.

Определим такие параметры производительности дисковой системы как коэффициент загрузки, суммарное время отклика на запрос, средний размер очереди, общее время проведенное запросом в очереди.

Время между запросами можно определить как

$$R_a = \frac{1}{a} = 10 \text{ мс.}$$

Коэффициент использования дисковой системы вычисляется следующим образом

$$U = \frac{R_S}{R_a} = \frac{8}{10} = 0,8$$

или 80% .

Определим время отклика дисковой системы

$$R = \frac{R_S}{1 - U} = \frac{8}{1 - 0,8} = 40 \text{ мс.}$$

Средний размер очереди составит

$$N_Q = U^2(1 - U) = 0,8^2(1 - 0,8) = 3,2.$$

Время, проведенное запросом в очереди

$$R_Q = UR = 0,8 \cdot 40 = 32 \text{ мс.}$$

Чтобы оценить динамику зависимости показателей функционирования дисковой системы от ее параметров увеличим мощность дисковой системы вдвое, то есть, предположим, что

$$R_S = 4 \text{ мс.}$$

Тогда

$$U = \frac{R_S}{R_a} = \frac{4}{10} = 0,4,$$

$$R = \frac{R_S}{1 - U} = \frac{4}{1 - 0,4} = 6,67 \text{ мс,}$$

$$N_Q = U^2(1 - U) = 0,4^2(1 - 0,4) = 0,26.$$

$$R_Q = UR = 0,4 \cdot 6,67 = 2,67 \text{ мс.}$$

Очевидно, что при увеличении мощности дисковой системы (уменьшение времени обслуживания системы вдвое) существенно уменьшается время отклика системы (почти в шесть раз).

Исследования подобных систем показывают, что связь между уровнем загрузки системы и временем отклика имеет нелинейный характер и может быть представлена следующим графиком (рис. 14).

При загрузке дисковой системы выше, чем $U = 0,7$ время отклика дисковой системы начинает увеличиваться в геометрической прогрессии относительно загрузки системы.

То есть параметры дисковой системы при реализации СХД следует выбирать так, чтобы их загрузка (коэффициент использования) не превышал 0,7.

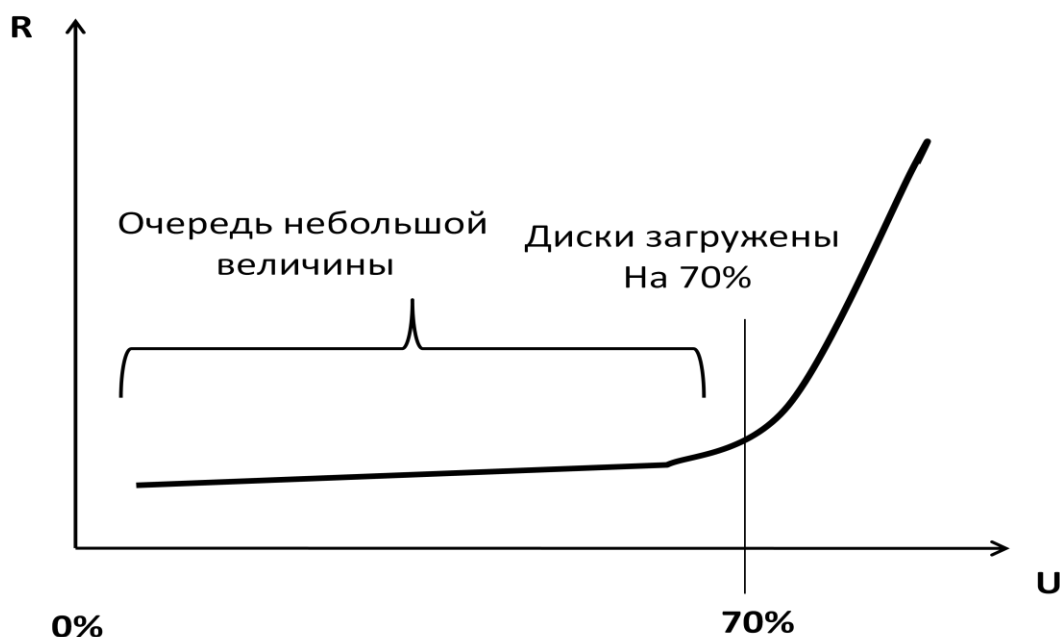


Рис. 14. Зависимость времени отклика от загрузки дисковой системы

1.5. Структура дисковых систем

Структура используемых дисковой системы определяется результатами анализа потребностей приложения.

Объем запоминающего устройства (емкость магнитного диска) достаточно легко рассчитывается исходя из объема подлежащей хранению информации, количества файловых систем, компонентов баз данных и других компонентов ИУС.

Требуемые значения временных характеристик дисковой системы также определяется на основании потребностями приложения.

Важнейшей характеристикой работы дискового устройства является значение IOPS (Input/Output Operations Per Second) – количество операций ввода/вывода в секунду.

Время обслуживания запроса приложения на запись или на чтение информации (R_a) можно определить как

$$R_a = E + L + X, \text{ где}$$

E – среднее время поиска блока информации,

L – среднее время задержки (время ожидания прохода блока информации под головкой)

X – среднее время передачи данных по внутренним шинам дисковой системы.

Рассмотрим работу дисковой системы, которая обладает следующими характеристиками.

$E = 5$ мс.

Скорость вращения диска равна 15000 об/мин.

Средняя скорость передачи данных по внутренним шинам диска – 40 Мб/с.

Объем передаваемых блоков информации составляет 32 Кб.

Заданная скорость вращения диска позволяет определить среднее время ожидания нужного блока данных как время, необходимое для половины оборота диска

$$L = \frac{0,5 \cdot 60 \cdot 1000}{15 \cdot 000} = 2 \text{ мс}$$

Среднее время передачи блока данных можно определить как

$$X = \frac{32 \text{ Кб}}{40 \cdot 000 \text{ Кб/с}} = 0,8 \text{ мс}$$

Тогда максимальное количество операций ввода/вывода в одну секунду при принятых характеристиках составит

$$IOPS = \frac{1}{R_a} = \frac{1}{E + L + S} = \frac{1}{(5 + 2 + 0,8) \text{ мс}} = 128 \text{ 1/с}$$

Изменяя объем блоков, используемых при выполнении операций ввода/вывода, можно изменять значение IOPS для рассматриваемого диска.

В табл. 1 представлены результаты расчетов IOPS для различных значений объема обрабатываемых блоков

Таблица 1. Результаты расчета IOPS

Объем блока (Кб)	Время обслуживания диска (R_a , мс)	IOPS (1/с)
4	7,1	140
8	7,2	139
16	7,4	135
32	7,8	128
64	8,6	116

Общее количество дисков (N), необходимых для приложения, рассчитывается следующим образом

$$N = \text{Max}(C, I), \text{ где}$$

C – количество дисков необходимых для выполнения требований по объему запоминающего устройства,

I - количество дисков необходимых для достижения требуемого значения IOPS.

Пример. Требования по объему запоминающего устройства для приложения составляет 1,46 Тб.

Максимальная работоспособность приложения составляет 9000 операций IOPS.

Поставщик оборудования предлагает диски со следующими характеристиками:

- емкость диска 146 Гб;
- скорость вращения диска 15000 об/мин;
- при загрузке 70% диск обеспечивает 180 операций IOPS.

Необходимо определить количество приобретаемых дисков.

Чтобы удовлетворить требования по объему запоминающего устройства потребуется

$$C = 1,46 \text{ Тб} / 146 \text{ Гб} = 10$$

дисков.

Для выполнения 9000 операций IOPS потребуется

$$I = (9000)/(180) = 50$$

дисков.

Таким образом, количество дисков, которое удовлетворяет всем запросам приложения составляет

$$N = \text{Max}(10, 50) = 50.$$

Рассмотрим еще один пример определения параметров функционирования дисковой системы в составе ИСУ.

Пример. Запросы в системе обрабатываются со скоростью 80 операций IOPS. Среднее время обслуживания запросов R_S (обычно задается поставщиком)

$$R_S = 6 \text{ мс}.$$

Необходимо определить:

- загруженность контроллера ввода/вывода дисковой системы (U);
- суммарное время отклика на запрос (R);
- средний размер очереди (N_Q);
- суммарное время, проведенное запросом в очереди (R_Q).

Время, затраченное диском на обработку запроса определяется как

$$R_a = \frac{1}{IOPS} = \frac{1}{80 \left(\frac{1}{c}\right)} = 12,5 \text{ мс},$$

Загруженность контроллера составит

$$U = \frac{R_S}{R_a} = \frac{6}{12,5} = 0,4 = 40\%.$$

Суммарное время отклика на запрос определится как

$$R = \frac{R_S}{(1 - U)} = \frac{6}{0,52} = 11,53 \text{ мс}.$$

Средний размер очереди рассчитывается как

$$N_Q = \frac{U^2}{(1 - U)} = \frac{0,23}{0,52} = 0,44.$$

Суммарное время, проведенное запросом в очереди составит

$$R_Q = UR = 0,48 \cdot 11,53 = 6 \text{ мс}.$$

2. Примеры решения задач и задания для самостоятельной работы

2.1. Задача 1

1. Построить FAT – таблицу для заданных файлов, с учетом заданных точек доступа, объема файлов и объема кластеров.
2. Определить потерю памяти для каждого файла.

Исходные данные представлены в табл. 2.

Таблица 2. Исходные данные для задачи 1

Вариант	Имя файла	Объем файла (байт)	Точка доступа	Объем кластера (Кб)
0	A	4500	2	4
	B	7800	1	
	C	8100	4	
	D	15250	7	
	E	23200	9	
1	A	14500	9	1
	B	17800	7	
	C	8100	1	
	D	5250	3	
	E	21200	5	
2	A	10500	3	2
	B	12800	1	
	C	8100	8	
	D	15250	10	
	E	8200	12	
3	A	4500	9	1
	B	9800	7	
	C	8100	5	
	D	4250	3	
	E	7200	1	
4	A	14500	2	4
	B	17800	7	
	C	8100	10	
	D	5250	3	
	E	21200	5	
5	A	13500	2	2
	B	14800	7	
	C	8100	1	
	D	5000	3	
	E	11200	4	
6	A	24500	1	8
	B	27800	3	
	C	28100	5	
	D	15250	8	
	E	21200	7	
7	A	4500	5	1
	B	7800	7	
	C	8100	1	
	D	5250	3	
	E	1200	4	

8	A	4500	7	0,5
	B	1800	5	
	C	8100	1	
	D	250	3	
	E	1250	4	
10	A	4500	9	2
	B	17800	7	
	C	7100	10	
	D	5250	3	
	E	2200	5	

Пример решения задачи 1

Рассмотрим решение задачи с нулевым вариантом исходных данных. Определим объемы файлов в кластерах, как целую часть следующего выражения

$$V_{fcl} = V_f / V_{cl} + 1$$

Результаты расчетов представлены в табл. 3.

Таблица 3. Объем хранящихся на диске файлов в кластерах

Имя файла	Объём в кластерах (V _{fcl})
A	2
B	2
C	3
D	4
E	6

Количество столбцов в FAT таблице определится общим количеством всех кластеров, необходимых для записи на диск всех файлов. В нашем случае эта величина составит 17 столбцов.

Учитывая материалы, представленные на рис.10, составим цепочки кластеров для каждого файла.

$$\begin{aligned}
 A &= 2 \rightarrow 3 \\
 B &= 1 \rightarrow 5 \\
 C &= 4 \rightarrow 6 \rightarrow 8 \\
 D &= 7 \rightarrow 10 \rightarrow 11 \rightarrow 12 \\
 E &= 9 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17
 \end{aligned}$$

Общий вид варианта заполнения FAT таблицы с отмеченными точками входа для каждого файла и соответствующими ссылками представлен на рис. 15.

	↓ В	↓ А		↓ С			↓ D		↓ E	
№ кластера	1	2	3	4	5	6	7	8	9	10
№ ссылки	5	3	eof	6	eof	8	10	eof	13	11
№ кластера	11	12	13	14	15	16	17	18	19	20
№ ссылки	12	eof	14	15	16	17	eof	0	0	0

Рис. 15. FAT таблица для нулевого варианта задания

Для расчета потерь памяти в файле А от произведения количества кластеров в этом файле на объем кластера вычтем объем файла А.

$$V_{\text{пот}} = 2\ 4048 - 4500 = 3596$$

Расчет потерь памяти при записи на диск остальных файлов производится аналогично. Результаты расчетов представлены в табл. 4.

Таблица 4. Потери памяти при записи на диск файлов А, В, С, D и E

Имя файла	Потери дисковой памяти (байт)
А	3596
В	296
С	4044
D	1142
E	1088

2.2. Задача 2

1. Описать заданные файлы в формате файловой системы NTFS (с использованием экстент);

2. Дефрагментировать диск с целью оптимизации описания файлов (допускается изменять точки доступа) и представить оптимизированные формы записи файлов

Исходные данные представлены в табл. 5.

Таблица 5. Исходные данные для решения задачи 2

Вариант	Имя файла	Объем файла (байт)	Точка доступа	Объем кластера (Кб)
0	A	4500	2	4
	B	7800	1	
	C	8100	4	
	D	15250	5	
	E	23200	10	
1	A	14500	9	1
	B	17800	5	
	C	8100	1	
	D	5250	8	
	E	21200	3	
2	A	10500	3	2
	B	12800	10	
	C	8100	8	
	D	15250	1	
	E	8200	12	
3	A	4500	3	1
	B	9800	7	
	C	8100	5	
	D	4250	10	
	E	7200	1	
4	A	14500	2	4
	B	17800	3	
	C	8100	10	
	D	5250	8	
	E	21200	5	
5	A	13500	5	2
	B	14800	7	
	C	8100	1	
	D	5000	8	
	E	11200	4	
6	A	24500	1	8
	B	27800	3	
	C	28100	5	
	D	15250	2	
	E	21200	7	
7	A	4500	5	1

	B	7800	7	
	C	8100	10	
	D	5250	3	
	E	1200	1	
8	A	4500	7	0,5
	B	1800	10	
	C	8100	1	
	D	250	3	
	E	1250	9	
9	A	4500	9	2
	B	17800	11	
	C	7100	10	
	D	5250	3	
	E	2200	1	

Пример решения задачи 2

Рассмотрим решение задачи с первым вариантом исходных данных.

Определим объемы файлов в кластерах, как целую часть следующего выражения

$$V_{fcl} = V_f / V_{cl} + 1$$

Результаты расчетов представлены в табл. 6.

Таблица 6. Объем хранящихся на диске файлов в кластерах

Имя файла	Объём в кластерах (V_{fcl})
A	15
B	18
C	9
D	6
E	22

Количество столбцов в FAT таблице определится общим количеством всех кластеров, необходимых для записи на диск всех файлов. В нашем случае эта величина составит 70 столбцов.

Общий вид варианта заполнения FAT таблицы с отмеченными точками входа для каждого файла и соответствующими ссылками представлен на рис. 16.

	↓ C			↓ E		↓ B		↓ D	↓ A	
№ кластера	1	2	3	4	5	6	7	8	9	10
№ ссылки	2	39	4	51	6	7	24	46	10	11
№ кластера	11	12	13	14	15	16	17	18	19	20
№ ссылки	12	13	14	15	16	17	18	19	20	21
№ кластера	21	22	23	24	25	26	27	28	29	30
№ ссылки	22	23	eof	25	26	27	28	29	30	31
№ кластера	31	32	33	34	35	36	37	38	39	40
№ ссылки	32	33	34	35	36	37	38	eof	40	41
№ кластера	41	42	43	44	45	46	47	48	49	50
№ ссылки	42	43	44	45	eof	47	48	49	50	eof
№ кластера	51	52	53	54	55	56	57	58	59	60
№ ссылки	52	53	54	55	56	57	58	59	60	61
№ кластера	61	62	63	64	65	66	67	68	69	70
№ ссылки	62	63	64	65	66	67	68	69	70	eof

Рис. 16. FAT таблица для первого варианта задания задачи 2

FAT таблица, приведенная на рис.16, определяет следующие цепочки кластеров для каждого файла.

A=9 →10 →11 →12 →13 →14 →15 →16 →17 →18 →19 →20 →21 →22 →23.

B=5 →6 →7 →24 →25 →26 →27 →28 →29 →30 →31 →32 →33 →34 →35 → 36 →37 →38.

C=1 →2 →39 →40 →41 →42 →43 →44 →45.

D=8 →46 →47 →48 →49 →50.

E=3 →4 →51 →52 →53 →54 →55 →56 →57 →58 →59 →60 →61 →62 →63 →64 →65 →66 →67 →68 →69 →70.

Согласно материалам, представленным в разделе 1.2, указанные файлы в терминах файловой системы NTFS могут быть записаны в виде следующей последовательности экстент:

$A = \{1, 9, 15\};$
 $B = \{1, 5, 3\}, \{2, 24, 15\};$
 $C = \{1, 1, 2\}, \{2, 39, 7\};$
 $D = \{1, 8, 1\}, \{2, 46, 5\};$
 $E = \{1, 3, 2\}, \{2, 51, 20\}.$

После проведения дефрагментации, изменяются точки доступа файлов таким образом, чтобы каждый файл мог быть записан в виде одной экстенты. В результате преобразований получим следующую форму записи файлов.

$A = \{1, 1, 15\};$
 $B = \{1, 16, 18\};$
 $C = \{1, 34, 9\};$
 $D = \{1, 43, 6\};$
 $E = \{1, 48, 22\}.$

2.3. Задача 3

1. Рассчитать общее время считывания файлов с заданного учебного диска с учетом следующих допущений:

- временем поиска имени файла в каталоге и временем построения цепочки кластеров файла пренебрежем;
- время передачи данных по шине не учитывается;
- время перехода головки на заданную дорожку определяется как время перехода на одну дорожку, умноженное на число пересекаемых дорожек.

2. Дефрагментировать диск и рассчитать общее время считывания всех файлов после дефрагментации

Варианты исходных данных для решения задачи представлены в табл. 7.

Таблица 7. Исходные данные для решения задачи 3

Последняя цифра номера зач. книжки	Число дорожек $N_{дор.}$	Число секторов на дорожке $N_{сек.}$	Число секторов в кластере $N_{кл.}$	Время перемещения головки на 1 дорожку T_1 , мс	Скорость вращения диска $N_{об.}$, об/мин
0	5	16	2	4	3600
1	4	24	4	3	4800
2	6	14	2	5	4200
3	6	16	4	6	3200
4	5	20	4	4	4600
5	5	24	4	2	3800

6	8	12	2	3	3600
7	7	14	2	4	4200
8	4	24	4	5	4800
9	5	18	2	6	3800

Для всех вариантов исходных данных при решении задачи следует учесть, что число поверхностей на учебном диске равно 1, объем сектора на диске равен 0,5 Кб, на диске записаны файлы А,В,С,Д и Е, параметры которых заданы в табл. 8.

Таблица 8. Параметры записанных на диск файлов

Имя файла	Объем в байтах	Точка входа
А	9500	12
В	10500	5
С	7000	6
Д	6500	1
Е	4500	10

Пример решения задачи 3

1. Расчет общего времени считывания файлов с заданного учебного диска.

Рассмотрим учебный магнитный диск с параметрами, указанными в табл. 9.

Таблица 9. Параметры магнитного диска.

Наименование параметра	Обозначение параметра	Значение параметра
Число дорожек (цилиндров)	<i>Ntrk</i>	4
Число секторов в дорожке	<i>Nsect</i>	20
Число секторов в кластере	<i>Nscl</i>	4
Объем сектора в байтах	<i>Vsect</i>	512
Время перемещения головки считывания/записи на одну дорожку в мс	<i>Tl</i>	4
Количество оборотов в мин	<i>Nob</i>	3600

На этом диске в корневом каталоге хранятся 5 файлов, объем которых и начальный кластер каждого файла (точка входа) заданы в табл.10.

Таблица 10. Параметры хранящихся на диске файлов

Имя файла	Объём в байтах	Начальный кластер файла
A	1500	12
B	4800	5
C	450	6
D	7000	1
E	6500	10

Составим таблицу распределения файлов (FAT) для заданных выше условий. Для этого определим объемы файлов в кластерах. Объем кластера для заданных условий определяется как

$$V_{cl} = V_{sect} \cdot N_{scl} = 512 \cdot 4 = 2048 \text{ байт}$$

или 2 килобайта. Соответственно объемы файлов в кластерах определится как целая часть следующего выражения

$$V_{fcl} = V_f / V_{cl} + 1$$

Результаты расчетов представлены в табл. 11.

Таблица 11. Объем хранящихся на диске файлов в кластерах

Имя файла	Объём в кластерах (V_{fcl})
A	1
B	3
C	1
D	4
E	4

Общее количество позиций в таблице расположения файлов (FAT) определяется емкостью диска в кластерах. Для решаемой задачи это значение можно рассчитать как

$$N_{dcl} = N_{trk} \cdot N_{sect} / N_{scl} = 4 \cdot 20 / 4 = 20$$

Таблица расположения файлов для заданного диска, с учетом объема каждого из файлов и их точек входа будет иметь следующий вид (табл.12).

Таблица 12. Таблица расположения файлов на диске

№ кластера	1	2	3	4	5	6	7	8	9	10
№ ссылки	2	3	4	eof	7	eof	8	eof	0	11
№ кластера	11	12	13	14	15	16	17	18	19	20
№ ссылки	13	eof	14	eof	0	0	0	0	0	0

В таблице 12 во второй строке (№ ссылки) указан или номер следующего кластера входящего в состав файла, или признак конца файла - eof (последний входящий в состав файла кластер), или признак свободного кластера - 0 (кластер не входит в состав файла).

Из таблицы 12 можно получить информацию о цепочках кластеров составляющих файлы:

$$A = 12,$$

$$B = 5 \rightarrow 7 \rightarrow 8,$$

$$C = 6,$$

$$D = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4,$$

$$E = 10 \rightarrow 11 \rightarrow 13 \rightarrow 14.$$

Следующим этапом решения поставленной задачи является построение карты диска, которая отражает расположение кластеров по дорожкам диска. Количество дорожек задано и равно 4, общий объем диска в кластерах равен 20, следовательно, на каждой дорожке диска расположено 5 кластеров.

Карта диска будет иметь следующий вид (табл.13).

Таблица 13. Карта диска

№ дорожки	Кластеры				
1	1 D1	2 D2	3 D3	4 D4	5 B1
2	6 C	7 B2	8 B3	9 0	10 E1
3	11 E2	12 A	13 E3	14 E4	15 0
4	16 0	17 0	18 0	19 0	20 0

Каждая клетка карты диска содержит номер кластера диска (число в левой части клетки) и обозначение входящего в состав файла кластера с указанием его номера (B3- третий кластер файла B).

Определим время, за которое диск делает один оборот

$$T_{ob} = 60 \cdot 1000 / N_{ob} = 60000 / 3600 = 16,7 \text{ мс},$$

где 60 – количество секунд в минуте, 1000 количество миллисекунд в секунде.

За это время под головкой считывания/записи диска проходят все кластеры одной дорожки, для нашего случая - это пять кластеров, следовательно, время считывания одного кластера составит

$$T_{cl} = T_{ob} / 5 = 16,7 / 5 = 3,34 \text{ мс}$$

Для считывания первого кластера файла головка должна быть установлена на дорожку, где расположен этот кластер. Номер дорожки определяется по карте диска.

Время перемещения головки на одну дорожку задано, значит, легко можно определить время позиционирования головки на дорожку как:

$$T_{poz} = T_1 \cdot N = 3 \cdot N \text{ мс},$$

где $T_1 = 4 \text{ мс}$ - время перемещения головки на одну дорожку (задано), N - число переходов с дорожки на дорожку при перемещении головки считывания/записи.

Когда головка установится на нужной дорожке, необходимо потратить какое-то время на ожидание появления под головкой нужного кластер.

Поскольку этот процесс случайный, то, предполагая равномерный закон распределения вероятностей значений времени ожидания, можно принять за время ожидания величину полуоборота диска, то есть:

$$T_{oj} = T_{ob} / 2 = 8,35 \text{ мс}.$$

В общем случае, время считывания любого кластера определится как сумма

$$T_{ocl} = T_{poz} + T_{oj} + T_{cl}$$

А общее время считывания файла будет представлять собой сумму значений считывания кластеров, которые входят в состав файла.

Определим время считывания файла А. Файл занимает один кластер, который расположен на третьей дорожке, следовательно

$$T_a = T_{poz} + T_{oj} + T_{cl} = 3 \cdot 4 + 8,35 + 3,34 = 23,69 \text{ мс}.$$

При расчете времени считывания файла В, необходимо учесть, что файл состоит из трех кластеров, которые расположены на первой и второй дорожках диска.

Время считывания первого кластера файла В определим следующим образом

$$T_{b1} = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

Для считывания второго кластера файла В необходимо переместить головку считывания с первой на вторую дорожку, учесть время ожидания прохождения нужного кластера под головкой считывания и считать кластер.

$$T_{b2} = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

Для считывания третьего кластера файла В перемещения на головки считывания на другую дорожку не требуется, не требуется и ожидания кластера, так как кластер b3 располагается на той же дорожке сразу за кластером b2, следовательно

$$T_{b3} = T_{cl} = 3,34 \text{ мс.}$$

Общее время считывания файла В определится как сумма времен считывания всех кластеров, входящих в состав этого файла.

$$T_b = T_{b1} + T_{b2} + T_{b3} = 15,69 + 15,69 + 3,34 = 34,72 \text{ мс.}$$

Аналогично рассчитывается время считывания данных для остальных файлов.

$$T_c = T_{poz} + T_{oj} + T_{cl} = 2 \cdot 4 + 8,35 + 3,34 = 19,69 \text{ мс.}$$

$$T_{d1} = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

$$T_{d2} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс.}$$

$$T_{d3} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс.}$$

$$T_{d4} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс.}$$

$$T_d = T_{d1} + T_{d2} + T_{d3} + T_{d4} = 15,69 + 3,34 + 3,34 + 3,34 = 25,71 \text{ мс.}$$

$$T_{e1} = T_{poz} + T_{oj} + T_{cl} = 2 \cdot 4 + 8,35 + 3,34 = 19,69 \text{ мс.}$$

$$T_{e2} = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

$$T_{e3} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 3,34 + 3,34 = 6,68 \text{ мс.}$$

$$T_{e4} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс.}$$

$$T_e = T_{e1} + T_{e2} + T_{e3} + T_{e4} = 19,69 + 15,69 + 6,68 + 3,34 + 3,34 = 48,74 \text{ мс.}$$

$$T_{sum} = T_a + T_b + T_c + T_d + T_e = 23,69 + 34,72 + 19,69 + 25,71 + 48,74 = 152,55 \text{ мс.}$$

Анализ результатов расчетов показывает, что основные затраты времени при считывании информации с магнитных дисков определяются временем

позиционирования головок на требуемую дорожку диска и временем ожидания прохода нужного кластера под головкой считывания.

2. Дефрагментация диска и расчет времени считывания файлов.

Дефрагментация диска предполагает выполнение операций упорядочения расположения кластеров для каждого из записанных на диске файлов с целью повышения эффективности использования дисковой памяти и уменьшения времени считывания файлов.

Поскольку на алгоритм проведения дефрагментации диска не накладывается никаких ограничений, то порядок выполнения операций дефрагментации определяется студентом самостоятельно.

В настоящем примере рассматривается один из вариантов дефрагментации, который предполагает размещение небольших файлов на внешних дорожках файла.

Результаты такой дефрагментации представлены в таблице 14.

Таблица 14. Карта диска после дефрагментации

№ дорожки	Кластеры				
1	1 A	2 C	3 B1	4 B2	5 B3
2	6 D1	7 D2	8 D3	9 D4	10 E1
3	11 E2	12 E3	13 E4	14 0	15 0
4	16 0	17 0	18 0	19 0	20 0

Рассчитаем общее время считывания файлов с диска после выполнения операции дефрагментации.

$$T_a = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

$$T_c = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

$$T_{b1} = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

$$T_{b2} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс}$$

$$T_{b3} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс}$$

$$T_b = T_{b1} + T_{b2} + T_{b3} = 15,69 + 3,34 + 3,34 = 22,37 \text{ мс.}$$

$$T_{d1} = T_{poz} + T_{oj} + T_{cl} = 2 \cdot 4 + 8,35 + 3,34 = 19,69 \text{ мс.}$$

$$T_{d2} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс}$$

$$T_{d3} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс}$$

$$T_{d4} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс}$$

$$T_d = T_{d1} + T_{d2} + T_{d3} + T_{d4} = 19,69 + 3,34 + 3,34 + 3,34 = 29,71 \text{ мс.}$$

$$T_{e1} = T_{poz} + T_{oj} + T_{cl} = 2 \cdot 4 + 8,35 + 3,34 = 19,69 \text{ мс.}$$

$$T_{e2} = T_{poz} + T_{oj} + T_{cl} = 1 \cdot 4 + 8,35 + 3,34 = 15,69 \text{ мс.}$$

$$T_{e3} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс}$$

$$T_{e4} = T_{poz} + T_{oj} + T_{cl} = 0 \cdot 4 + 0 + 3,34 = 3,34 \text{ мс}$$

$$T_e = T_{e1} + T_{e2} + T_{e3} + T_{e4} = 19,69 + 15,69 + 3,34 + 3,34 = 42,06 \text{ мс.}$$

$$T_{sum1} = T_a + T_b + T_c + T_d + T_e = 15,69 + 22,37 + 15,69 + 29,71 + 42,06 = 125,52 \text{ мс.}$$

После проведения операций дефрагментации диска общее время считывания файлов уменьшилось в 1,22 раза.

2.4. Задача 4

Рассчитать следующие параметры производительности дисковой системы:

1. Коэффициент загрузки - U ;
2. Средний размер очереди запросов - N_Q ;
3. Среднее время, проведенное запросом в очереди R_Q .

Определить новые параметры дисковой системы (R_S), позволяющие уменьшить время нахождения запроса в очереди на указанную в исходных данных величину.

Варианты исходных данных для решения задачи представлены в табл. 15.

Таблица 15. Варианты исходных данных для задачи 4

Вариант	Скорость поступления запросов a 1/с	Время обслуживания запросов R_S мс	Уменьшение среднего времени нахождения запроса в очереди ΔR_Q
0	150	4,71	25%
1	200	4,00	10%
2	80	6,25	50%
3	225	1,78	15%
4	175	3,43	30%
5	120	5,00	20%
6	100	5,75	40%
7	90	3,00	35%
8	130	4,50	25%
9	190	4,25	17%

Пример решения задачи 4.

Рассмотрим решение задачи со вторым вариантом исходных данных.

Коэффициент загрузки системы определим с учетом согласования размерности величин следующим образом.

$$U = a \cdot R_s = 80 \cdot 0,001 \cdot 6,25 = 0,5.$$

$$R_Q = U^2(1-U) = 0,25 \cdot 0,5 = 0,125.$$

$$R_Q = \frac{R_s}{1-U} - R_s = \frac{6,25}{1-0,5} - 6,25 = 6,25 \text{ мс.} \quad (*)$$

Уменьшим значение R_Q на 50% то есть, новое значение среднего времени, проведенное запросом в очереди составит

$$R_{Q1} = R_Q - 0,5 \cdot R_Q = 6,25 - 3,125 = 3,125 \text{ мс.}$$

Определим значение характеристики R_{S1} для новой дисковой системы, которое обеспечит снижение времени нахождения запросов в очереди до величины 3,125 мс.

Преобразуем выражение (*) к следующему виду

$$R_{Q1} = \frac{R_{S1}}{1-U} - R_{S1} = \frac{R_{S1} - (R_{S1} - R_{S1}U)}{1-U} = \frac{R_{S1}U}{1-U} = \frac{R_{S1}^2 a}{1 - aR_{S1}}.$$

Откуда следует, что

$$aR_{S1}^2 + aR_{Q1}R_{S1} - R_{Q1} = 0 ,$$

и далее, решив квадратное уравнение относительно R_{S1} , определим

$$R_{S1} = 4,88 \text{ мс.}$$

Проверка правильности решения заключается в вычислении значения времени ожидания в очереди R_{Q1} при использовании новой дисковой системы со значением параметра $R_{Q1} = 4,88 \text{ мс.}$

$$U_1 = a \cdot R_{S1} = 80 \cdot 0,001 \cdot 4,88 = 0,39$$

$$R_{Q1} = \frac{R_{S1}}{1-U} - R_{S1} = \frac{4,88}{1-0,39} - 4,88 = 3,125 \text{ мс,}$$

что полностью соответствует условиям задачи.

2.5. Задача 5

Определить какой должна быть скорость вращения диска (N_d) в оборотах в минуту, чтобы обеспечивалось заданное значение IOPS. Исходные данные для расчетов приведены в таблице.

Варианты исходных данных для решения задачи представлены в табл.16.

Таблица 16. Варианты исходных данных для задачи 5

Вариант	Среднее время поиска блока информации на диске (E) мс.	Средняя скорость передачи блока информации по шинам (v) Мб/с.	Объем используемых для передачи данных блоков (V) Кб	Требуемое значение IOPS 1/с
0	5	40	32	128
1	4	60	16	116
2	6	20	8	135
3	4	20	4	140
4	4	60	16	115
5	5	10	8	120
6	3	15	32	140
7	5	40	64	110
8	4	20	16	90
9	5	18	32	125

Пример решения задачи 5.

Рассмотрим решение задачи с третьим вариантом исходных данных.

Определим время обслуживания одного запроса

$$R_a = \frac{1}{IOPS} = \frac{1000}{140} = 7,14 \text{ мс.}$$

Среднее время передачи блока составит

$$X = \frac{V}{v} = \frac{4 \cdot 1000}{20000} = 0,2 \text{ мс.}$$

Среднее время ожидания нужного блока при считывании или записи информации определим как разность

$$L = R_a - (E + X) = 7,14 - (4 + 0,2) = 2,94 \text{ мс.}$$

Если учесть, что среднее время ожидания нужного блока составляет время необходимое для половины оборота диска, то скорость вращения диска составит

$$N_D = \frac{0,5 \cdot 60 \cdot 1000}{L} = \frac{30000}{2,94} = 10205 \text{ об / мин .}$$

3. Методические указания к выполнению лабораторных работ

3.1. Лабораторная работа 1

Управление файлами и каталогами из командной оболочки ос

Цель работы

Исследование и получение практических навыков управления файлами и каталогами из командной оболочки операционной системы (ОС).

Основные теоретические положения

ОС наряду с графическим интерфейсом снабжается командной оболочкой. Командная оболочка – это отдельная программная составляющая ОС, которая обеспечивает прямую связь между пользователем и операционной системой с текстовым интерфейсом командной строки. Командная оболочка обеспечивает выполнение команд управления ОС, результаты выполнения отображаются на экране в виде сходном с результатами работы интерпретатора Command.com MS DOS. Командная оболочка современных ОС (Windows XP, Vista, 7, 10) использует для перевода введенной команды в формат понятный компьютеру и выполнения команды интерпретатор команд cmd.com.

Запуск командной оболочки осуществляется следующим образом:

Пуск-Все программы- Стандартные- Командная строка

Порядок выполнения работы.

3.1.1. Работа с каталогами

1. Запустить командную строку.

2. Пользуясь внутренними командами `cmd.com` для работы с каталогами (`C:` - переход на диск `C`, `dir (путь)(имя_файла) (/p) (/w)`, `cd путь`, `md путь`, `rd путь`), создать заданное дерево каталогов (рис.17).

Сведения о поддерживаемых `cmd.exe` командах можно получить, используя запрос `help` из командной строки ОС, сведения о порядке использования любой из команд можно получить из справочной системы ОС, используя запрос следующего вида `md /?` (в данном случае – все о команде `md`). Перед построением заданного дерева каталогов необходимо перейти на диск `D:`, создать каталог `UD`, перейти в этот каталог, создать директорий `UD_FAMILIA` (где `FAMILIA` две буквы из фамилии студента), перейти в этот каталог и уже в нем создать дерево каталогов согласно заданию.

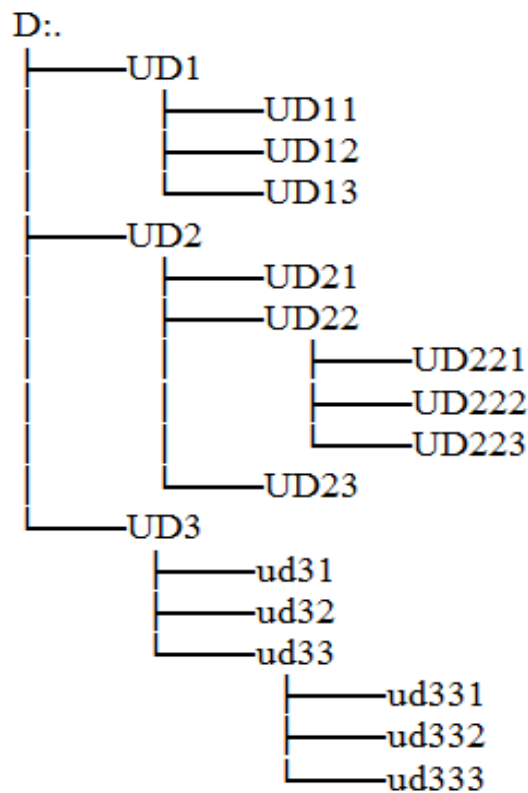


Рис. 17. Заданное дерево каталогов

3. Вывести на экран содержание каталога `UD_FAMILIA`, включая все подкаталоги.

4. Удалить подкаталоги `ud331`, `ud332`, `ud333`

5. Вывести на экран содержание каталога `UD_FAMILIA`, включая все подкаталоги.

6. Все действия оператора и результаты работы введенных команд документировать с помощью операции пометить (в окне командной строки) и операции вставить (в окне текстового редактора).

3.1.2. Работа с файлами

Пользуясь внутренними командами (*copy*, *echo*) *cmd.com*, создать файлы, перечисленные в задании:

f1.txt, f2.txt, f3.txt,
f4.pas, f5.pas, f6.pas,
f7.cpp, f8.cpp, f9.cpp
f10.bat, f11.bat, f12.bat,
f13.exe, f14.exe, f15.exe,
f16.gif, f17.gif, f18.gif,
f19.com, f20.com, f21.com,
f22.tmp, f23.tmp, f24.tmp.

Все файлы расположить в директории UD_FAMILIA.

Содержимое файлов определить как текст: *FILE 1* для f1.txt, *FILE 11* для f11.bat и т.д.

1. Вывести на экран содержание директории UD_FAMILIA, включая все подкаталоги. Вывести на экран все временные файлы из директории UD_FAMILIA. Показать дату и время создания файлов, а также их размер. Привести вид экрана.

2. Вывести на экран все программные файлы из директории UD_FAMILIA, показать только имена файлов. Привести вид экрана.

3. К файлам f1.txt, f2.txt присоединить по два потока: к файлу f1.txt – поток 1 с содержанием: potok1, поток 2 с содержанием: potok2; к файлу f2.txt – поток 3 с содержанием: potok3, поток 4 с содержанием: potok4. Привести вид экрана.

4. Распечатать на экране содержимое неименованных потоков для f1.txt и f2.txt. Привести вид экрана

5. Распечатать на экране содержимое именованных потоков для f1.txt и f2.txt. Привести вид экрана

6. Скопировать f1.txt и f2.txt на мобильный носитель (дискету, карту флэш-памяти).

7. Распечатать на экране содержимое именованных потоков для f1.txt и f2.txt, находящихся на мобильном носителе. Привести вид экрана

8. Из каталога D:\ UD_FAMILIA скопировать:

в каталог D:\ UD_FAMILIA\ UD2\UD22\UD222 все текстовые файлы
в каталог D:\ UD_FAMILIA\ UD2\UD22\UD223 файлы рисунков
в каталог D:\ UD_FAMILIA\ UD2\UD22\UD221 все командные файлы.

9.Из каталога D:\ UD_FAMILIA скопировать одной командой *FOR* все программные файлы (.cpp и .pas) в каталог D:\ UD_FAMILIA\ UD3\UD32

10.Из каталога UD_FAMILIA переместить :

в каталог D:\ UD_FAMILIA\ UD1\UD12\ исходные модули на языке C++, в каталог D:\ UD_FAMILIA\ UD1\UD13\ исполняемы машинные программы

11.Вывести на экран содержание всего дерева каталогов, начиная с директории UD_FAMILIA, показать только имена файлов. Привести вид экрана

12.Удалить временные файлы из каталога UD_FAMILIA

13.Вывести на экран содержание всего дерева каталогов, начиная с директории UD_FAMILIA, показать только имена файлов. Привести вид экрана.

14.Все действия оператора и результаты работы введенных команд документировать с помощью операции **пометить** (в окне командной строки) и **операции вставить** (в окне текстового редактора).

3.1.3. Резервное копирование материалов

1.Перейти в корневой каталог лабораторной работы (UD_FAMILIA)

2.Создать директорий *backuplab1*

3.С помощью команды **xcopy** выполнить резервное копирование всех результатов лабораторной работы (включая пустые каталоги) в директорий *backuplab1*.

4.Вывести на экран содержание всего каталога *backuplab1*. Привести вид экрана.

5.Все действия оператора и результаты работы введенных команд документировать с помощью операции **пометить** (в окне командной строки) и **операции вставить** (в окне текстового редактора).

3.1.4. Содержание отчета

Отчет должен содержать:

1. Задание на лабораторную работу.

2. Распечатки команд и результатов их выполнения по каждому пункту задания.

3.2. Лабораторная работа 2

Командные файлы. Часть 1

Цель работы

Исследование и получение практических навыков программирования командных файлов.

Основные теоретические положения

Командный (пакетный, бат-) файл представляет собой текстовый файл с набором инструкций командному процессору **cmd. exe**, позволяющий автоматически выполнить записанные в текстовом файле набор команд ОС.

В качестве инструкций могут выступать как имена исполняемых файлов, так и специальные инструкции командного процессора.

Command.com MS DOS. Командная оболочка современных ОС (Windows XP, Vista) использует для перевода введенной команды в формат понятный компьютеру и выполнения команды интерпретатор команд **cmd. com**.

Запуск командной оболочки осуществляется следующим образом:

Пуск-Все программы- Стандартные- Командная строка

Порядок выполнения работы.

3.2.1. Работа с каталогами

1. Запустить командную строку.

2. Перед выполнением задания необходимо перейти на диск D:, создать каталог UD, перейти в этот каталог, создать директорий UD_FAMILIA (где FAMILIA две буквы из фамилии студента), перейти в этот каталог и уже в нем выполнять задание лабораторной работы.

3. Пользуясь текстовым редактором (Блокнот) создать командный файл kom1.bat, выполнение которого создаст заданное дерево каталогов и выведет на экран содержание каталога UD_FAMILIA, включая все подкаталоги

4. Запустить командный файл на исполнение и проверить его работу

5. Все действия оператора и результаты работы введенных команд документировать с помощью операции **пометить** (в окне командной строки) и **операции вставить** (в окне текстового редактора).

Сведения о командах можно получить из справочной системы, используя запросы следующего вида : **cd / ?**

3.2.2. Работа с файлами

1. Пользуясь текстовым редактором (Блокнот, Word) создать командный файл kom2.bat, выполнение которого создаст файлы, перечисленные в задании:

f1.txt, f2.txt, f3.txt,
f4.pas, f5.pas, f6.pas,
f7.cpp, f8.cpp, f9.cpp
f10.bat, f11.bat, 12.bat,
f13.exe, f14.exe, f15.exe,
f16.gif, f17.gif, f18.gif,
f19.com, f20.com, f21.com,
f22.tmp, f23.tmp, f24.tmp.

Все файлы расположить в директории UD_FAMILIA.

Содержимое файлов определить как текст: **FILE 1** для f1.txt, **FILE 11** для f11.bat и т.д. (Для создания файлов и определения их содержания использовать комбинацию команд **echo** и **>**)

2. В командном файле предусмотреть выполнение следующих операций:

Вывод на экран содержания каталога UD_FAMILIA, включая все подкаталоги. Вывести на экран все временные файлы из директории UD_FAMILIA. Показать дату и время создания файлов, а также их размер.

Вывести на экран все программные файлы из директории UD_FAMILIA, показать только имена файлов.

Вывести на экран все программные файлы из директории UD_FAMILIA, показать только имена файлов.

Из каталога D:\ UD_FAMILIA скопировать:

в каталог D:\ UD_FAMILIA\ UD2\UD22\UD222 все текстовые файлы

в каталог D:\ UD_FAMILIA\ UD2\UD22\UD223 файлы рисунков

в каталог D:\ UD_FAMILIA\ UD2\UD22\UD221 все командные файлы.

Из каталога D:\ UD_FAMILIA скопировать одной командой FOR все программные файлы (.cpp и .pas) в каталог D:\ UD_FAMILIA\ UD3\UD32

Из каталога UD_FAMILIA переместить :

в каталог D:\ UD_FAMILIA\ UD1\UD12\ исходные модули на языке C++

в каталог D:\ UD_FAMILIA\ UD1\UD13\ исполняемые машинные программы

Удалить временные файлы из каталога UD_FAMILIA

Вывести на экран содержание всего дерева каталогов, начиная с директории UD_FAMILIA , показать только имена файлов.

3. Запустить командный файл на исполнение и проверить его работу
4. Все действия оператора и результаты работы введенных команд документировать с помощью операции **пометить** (в окне командной строки) и **операции вставить** (в окне текстового редактора).

3.2.3. Резервное копирование материалов

1. Пользуясь текстовым редактором (Блокнот, Word) создать командный файл kom3.bat, выполнение которого реализует резервное копирование в каталог backuplab2 всех файлов и подкаталогов учебного каталога UD_FAMILIA
2. Вывести на экран содержание всего каталога backuplab1.
3. Все действия оператора и результаты работы введенных команд документировать с помощью операции **пометить** (в окне командной строки) и **операции вставить** (в окне текстового редактора).

3.2.4. Содержание отчета

Отчет должен содержать:

1. Задание на лабораторную работу.
2. Распечатки командных файлов и результатов их выполнения по каждому пункту задания.
3. Описание используемых в лабораторной работе команд.

3.3. Лабораторная работа 3

Командные файлы часть 2

Цель работы

Исследование процесса и получение практических навыков программирования командных файлов.

Основные теоретические положения

Командный (пакетный, бат-) файл представляет собой текстовый файл с набором инструкций командному процессору **cmd.exe**, позволяющий автоматически выполнить записанный в текстовом файле набор команд ОС.

В качестве инструкций могут выступать как имена исполняемых файлов, так и специальные инструкции командного процессора **Command.com MS DOS**.

Командная оболочка современных операционных систем (Windows XP, Vista, 7, 10) использует для перевода введенной команды в формат понятный компьютеру и выполнения команды интерпретатор команд **cmd. com**.

Запуск командной оболочки осуществляется следующим образом:

Пуск-Все программы- Стандартные- Командная строка

Порядок выполнения работы.

3.3.1. Присвоение значений внутренним переменным. Передача параметров в бат-файлы.

1. Запустить командную строку.
2. Пользуясь внутренними командами **cmd.com** для работы с каталогами (**C:** - переход на диск C, **dir** (путь)(*имя_файла*) (**/p**) (**/w**), **cd** путь, **md** путь, **rd** путь) , перейти на диск D:, создать каталог U:\ UD_FAMILIA
3. Пользуясь текстовым редактором (Блокнот, Word) создать командный файл kom31.bat, выполнение которого реализует следующую последовательность операций:
 - a. Взять 1-й параметр бат-файла, присвоить внутренней переменной p1 численное значение 1-го параметра
 - b. Взять 2-й параметр бат-файла, присвоить внутренней переменной p2 численное значение 2-го параметра
 - c. Присвоить внутренней переменной p3 значение суммы p1+p2
 - d. Вывести на экран значения переменных p1, p2, p3 и сообщение об имени отработавшего бат-файла
 - e. Направить результаты работы бат-файла в файл протокола работы командного файла.
 - f. Имя файла протокола сформировать как конкатенацию: (имя бат-файла)_log1.txt
4. Запустить командный файл на исполнение и проверить его работу
5. Используя команду SET, проконтролировать появление переменных P1, P2 и P3 в списке переменных окружения ОС.
6. Исследовать работу бат-файла при неопределенных параметрах.
7. Используя команду SET, исследовать поведение переменных P1, P2 и P3 в списке переменных окружения ОС.
8. Все действия оператора и результаты работы введенных команд документировать с помощью операции **пометить** (в окне командной строки) и **операции вставить** (в окне текстового редактора).
9. Содержимое файла протокола работы бат-файла привести в отчете
10. Сведения о командах можно получить из справочной системы ОС, используя запросы следующего вида: **set / ?**

3.3.2. Использование if и goto для исключения возможности работы командного файла с неопределенными параметрами

1. Запустить командную строку.
2. Пользуясь внутренними командами **cmd.com** создать командный файл kom32.bat как копию kom31.bat.

Пользуясь текстовым редактором (Блокнот, Word) отредактировать командный файл kom32.bat таким образом, чтобы его запуск с неопределенными параметрами не приводил к изменению значений внутренних переменных p1, p2 и p3. В этом случае запуск командного файла должен вывести на экран (и в файл протокола исполнения) сообщение **parameters absent**

3. Используя команду SET, проконтролировать поведение переменных P1, P2 и P3 в списке переменных окружения ОС.

3.3.3. Содержание отчета

Отчет должен содержать:

1. Задание на лабораторную работу.
2. Распечатки командных файлов и результатов их выполнения по каждому пункту задания.
3. Описание используемых в лабораторной работе команд.
4. Распечатки файлов протоколов исполнения командных файлов.

3.4. Лабораторная работа 4

Работа с дисковым запоминающим устройством из командной оболочки ОС

Цель работы

Исследование и получение практических навыков работы с дисковым запоминающим устройством из командной оболочки операционной системы.

Основные теоретические положения

ОС наряду с графическим интерфейсом снабжается командной оболочкой. Командная оболочка – это отдельная программная составляющая ОС, которая обеспечивает прямую связь между пользователем и операционной системой с текстовым интерфейсом командной строки. Командная оболочка

обеспечивает выполнение команд управления ОС, результаты выполнения отображаются на экране в виде сходном с результатами работы интерпретатора **Command.com MS DOS**. Командная оболочка современных ОС (Windows XP, Vista, 7, 10) использует для перевода введенной команды в формат понятный компьютеру и выполнения команды интерпретатор команд **cmd.com**.

Запуск командной оболочки осуществляется следующим образом:
Пуск-Все программы- Стандартные- Командная строка

Порядок выполнения работы.

3.4.1. Работа с дисковым запоминающим устройством

1. Запустить командную строку.
2. Пользуясь внутренними командами **cmd.com** для работы с каталогами (**C:** - переход на диск C, **dir** (путь)(*имя_файла*) (**/p**) (**/w**), **cd** путь, **md** путь, **rd** путь) , создать заданное дерево каталогов

Сведения о поддерживаемых **cmd.exe** командах можно получить, используя запрос **help** из командной строки ОС, сведения о порядке использования любой из команд можно получить из справочной системы ОС, используя запрос следующего вида **md /?** (в данном случае – все о команде **md**).

Перед выполнением задания на лабораторную работу необходимо перейти на диск **D:**, создать каталог **UD**, перейти в этот каталог, создать директорий **UD_FAMILIA** (где **FAMILIA** две буквы из фамилии студента), перейти в этот каталог и уже в нем продолжать работу.

3. Пользуясь командой **subst** создать виртуальный диск **k:** . Перейти на диск **k:** .
4. Используя команду **FOR** , с соответствующим ключом и команду **chkntfs** , определить, имеются ли на дисках **c:**, **d:**, **k:** фрагменты недействительных (потерянных) данных.
5. Определить значение установленного таймаута для команды **chkntfs**
6. Изменить значение таймаута отмены запуска **chkntfs** . Задать значение таймаута 30 сек.
7. Используя команду **chkdsk**, определить текущий размер файла-журнала (мета-файла) в NTFS.
8. Используя команду **chkdsk**, проверить диск **d:**, отчет о результатах проверки привести в отчете по лабораторной работе. Отчет о результатах проверки диска направить в файл **CHKDSK_LOG.TXT**
9. Все действия оператора и результаты работы введенных команд документировать с помощью операции **пометить** (в окне командной строки) и **операции вставить** (в окне текстового редактора).

3.4.2. Работа с переменными окружения ОС

1. Пользуясь командой **set** вывести на экран все переменные окружения ОС.
2. Распечатать на экране значение переменной **Path**.
3. Запомнить значение переменной **Path** в файле **fpath.txt**
4. Запомнить содержание переменной **Path** во вспомогательной переменной **P1**. Проконтролировать наличие **P1** в списке переменных окружения ОС.
5. Запустить из командной строки текстовый редактор Блокнот.
6. Удалить из списка переменных окружения ОС переменную **Path**.
7. Запустить из командной строки текстовый редактор Блокнот. Оценить состав операций при запуске текстового редактора в первом и во втором случаях.
8. Восстановить значение переменной **Path** в списке переменных окружения ОС. Запустить из командной строки текстовый редактор Блокнот.
9. Добавить новый путь (**C:\TMP**) в переменную **Path**. Распечатать новое значение **Path** на экране. Сохранить новое значение в файле, добавив это значение в конец файла **fpath.txt**.

3.4.3. Содержание отчета

Отчет должен содержать:

1. Задание на лабораторную работу.
2. Распечатки команд и результатов их выполнения по каждому пункту задания.
3. Описание используемых в лабораторной работе команд.

3.5. Примеры использования команд управления элементами СХД

Чтобы получить информацию о командах управления элементами СХД наберите в командной строке компьютера команду **HELP**. Результат работы этой команды представлен на рис. 18.

C:\Users\GANI>HELP	
Для получения сведений об определенной команде наберите HELP <имя команды>	
ASSOC	Вывод либо изменение сопоставлений по расширениям имен файлов.
ATTRIB	Отображение и изменение атрибутов файлов.
BREAK	Включение и выключение режима обработки комбинации клавиш CTRL+C.
BCDEDIT	Задаёт свойства в базе данных загрузки для управления начальной загрузкой.
CACLS	Отображение и редактирование списков управления доступом (ACL) к файлам.
CALL	Вызов одного пакетного файла из другого.
CD	Вывод имени либо смена текущей папки.
CHCP	Вывод либо установка активной кодовой страницы.
CHDIR	Вывод имени либо смена текущей папки.
CHKDSK	Проверка диска и вывод статистики.
CHKNTFS	Отображение или изменение выполнения проверки диска во время загрузки.
CLS	Очистка экрана.
CMD	Запуск еще одного интерпретатора командных строк Windows.
COLOR	Установка цветов переднего плана и фона, используемых по умолчанию.
COMP	Сравнение содержимого двух файлов или двух наборов файлов.
COMPACT	Отображение и изменение сжатия файлов в разделах NTFS.
CONVERT	Преобразование дисковых томов FAT в NTFS. Нельзя выполнить преобразование текущего активного диска.
COPY	Копирование одного или нескольких файлов в другое место.
DATE	Вывод либо установка текущей даты.
DEL	Удаление одного или нескольких файлов.
DIR	Вывод списка файлов и подпапок из указанной папки.
DISKCOMP	Сравнение содержимого двух гибких дисков.
DISKCOPY	Копирование содержимого одного гибкого диска на другой.
DISKPART	Отображение и настройка свойств раздела диска.
DOSKEY	Редактирование и повторный вызов командных строк; создание макросов.
DRIVERQUERY	Отображение текущего состояния и свойств драйвера устройства.
ECHO	Вывод сообщений и переключение режима отображения команд на экране.
ENDLOCAL	Конец локальных изменений среды для пакетного файла.
ERASE	Удаление одного или нескольких файлов.
EXIT	Завершение работы программы CMD.EXE (интерпретатора командных строк).
FC	Сравнение двух файлов или двух наборов файлов и вывод различий между ними.
FIND	Поиск текстовой строки в одном или нескольких файлах.
FINDSTR	Поиск строк в файлах.
FOR	Запуск указанной команды для каждого из файлов в наборе.
FORMAT	Форматирование диска для работы с Windows.
FSUTIL	Отображение и настройка свойств файловой системы.
FTYPE	Вывод либо изменение типов файлов, используемых при сопоставлении по расширениям имен файлов.
GOTO	Передача управления в отмеченную строку пакетного файла.
GPRESULT	Отображение информации о групповой политике для компьютера или пользователя.
GRAFTABL	Позволяет Windows отображать расширенный набор символов в графическом режиме.
HELP	Выводит справочную информацию о командах Windows.
ICACLS	Отображение, изменение, архивация или восстановление списков ACL для файлов и каталогов.
IF	Оператор условного выполнения команд в пакетном файле.
LABEL	Создание, изменение и удаление меток тома для дисков.

Рис. 18. Результаты работы команды HELP

3.4.1. Команды работы с директориями и файлами

Порядок применения и результаты работы команд управления директориями, файлами и другими элементами СХД приведен на рисунках 19-23.

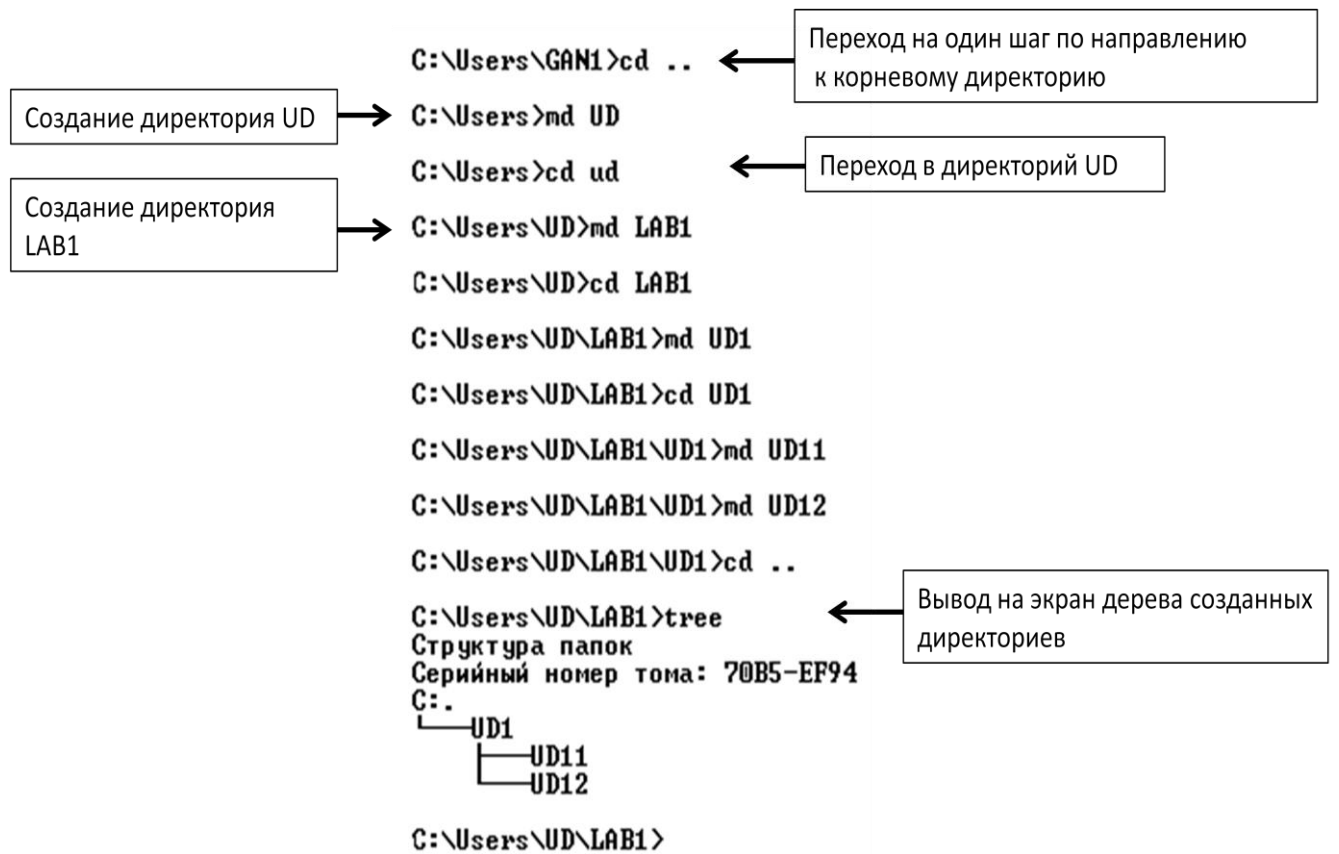


Рис. 19. Порядок использования и результаты работы команд управления директориями

```

C:\Users\UD\LAB1>echo FILE1 > f1.txt
C:\Users\UD\LAB1>DIR
Том в устройстве C не имеет метки.
Серийный номер тома: 70B5-EF94

Содержимое папки C:\Users\UD\LAB1

07.10.2017  11:50    <DIR>          .
07.10.2017  11:50    <DIR>          ..
07.10.2017  11:52                8 f1.txt
07.10.2017  11:19    <DIR>          UD1
                1 файлов          8 байт
                3 папок  418 979 782 656 байт свободно

C:\Users\UD\LAB1>type f1.txt
FILE1

C:\Users\UD\LAB1>

```

← Создание файла командой **echo**

← Вывод на экран содержимого директория LAB1

← Вывод на экран содержимого файла f1.txt

Рис. 20. Порядок использования и результаты работы команд управления файлами

```

C:\Users\Alex>cd ..
C:\Users>cd ..
C:\>md ud_IVANOU
C:\>cd ud_IVANOU
C:\ud_IVANOU>subst k: C:\ud_IVANOU
C:\ud_IVANOU>cd ..
C:\>k:
K:\>

```

← Создание виртуального диска K:

← Переход на диск K:

Рис. 21. Порядок использования и результаты работы команды создания виртуального диска

```
K:\>for %F in (C: K:) do (chkntfs %F)
```

← Совместное использование
Команд for и chkntfs

```
K:\>(chkntfs C: )  
Тип файловой системы: NTFS.  
Ошибки на C: не обнаружены.
```

```
K:\>(chkntfs K: )  
Невозможно опросить состояние диска K:
```

```
K:\>
```

Рис. 22. Порядок использования и результаты работы команд for и chkntfs

```
K:\>set path > path.txt
```

← Использование команды set для сохранения
переменной path в файле path.txt

```
K:\>dir  
Том в устройстве K имеет метку BOOTCAMP  
Серийный номер тома: AC92-9D5C
```

Содержимое папки K:\

```
24.10.2017 14:20 <DIR>      .  
24.10.2017 14:20 <DIR>      ..  
24.10.2017 14:23             169 path.txt  
                1 файл      169 байт  
                2 папок  40 891 424 768 байт свободно
```

```
K:\>type path.txt  
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32  
\WindowsPowerShell\v1.0\  
PATHEXT=.COM;.EXE;.BAT;.CMD;.UBS;.UBE;.JS;.JSE;.WSF;.WSH;.MSC
```

```
K:\>
```

Рис. 23. Порядок использования и результаты работы команды set

ЗАКЛЮЧЕНИЕ

Сегодня системы хранения данных работают во многих организациях по всему миру. В основном это организации, операционная деятельность кото-

рых предъявляет чрезвычайно высокие требования к доступности и производительности приложений, работающих с данными. Таковыми являются финансовые организации, для которых важны как производительность (поскольку задержки в функционировании основных систем могут стоить миллионы долларов), так и надежность (поскольку потеря важной финансовой информации может повлечь за собой еще более тяжелые последствия). Другим важнейшим потребителем систем хранения данных являются телекоммуникационные компании, где необходимо реализовать биллинг огромного количества информации в пределах заданного времени и, следуя нормативным актам регулирующих органов, надежно сохранять эту информацию длительный период времени.

Следует подчеркнуть, что в данном пособии рассмотрены вопросы решения лишь основных задач по определению параметров систем хранения данных.

В дальнейшем при изучении современных систем хранения данных следует обратить внимание на широкое использование в серверах и СХД корпоративного класса устройств долговременного хранения данных на базе флэш-памяти.

СПИСОК ЛИТЕРАТУРЫ

1. От хранения данных к управлению информацией / EMC Education Services ; ред. К. Белецкая [и др.] ; пер. с англ. В. Воротинцев, О. Гринвуд. - СПб. : Питер , 2010. - 522 с.
2. Джордж Джад. Основы проектирования SAN. Из-во "Edvance Edition", М. -2008, с. 589

Губин Александр Николаевич
СЕТИ ХРАНЕНИЯ ДАННЫХ
ПРАКТИКУМ

Редактор

План 2018 г., п.

Подписано к печати
Объем усл.-печ. л. Тираж экз. Заказ

Издательство СПбГУТ. 191186 СПб., наб. р. Мойки, 61

Отпечатано в