

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
к практическим занятиям студентов
09.02.07 Информационные системы и программирование
Общепрофессиональный цикл**

Разработали: преподаватели:
Батыргареева С.Ф.,
Корепанова Н.Л.,
Кротова О.В.
Сушкова Ю.А.

Севастополь
2024

Практическая работа № 1

Работа с интерпретатором командной строки

Цель работы: изучение команд среды командной строки windows и приобретение практических навыков их использования.

1. Основные теоретические сведения

Командная строка Windows является стандартным средством диагностики, настройки и управления компонентами операционной системы и прикладным программным обеспечением.

Командная строка Windows может быть запущена как и любое другое стандартное приложение - через Главное меню Windows, с использованием диалога Выполнить (комбинации клавиш Win+R, Win+X), а также с использованием открытия Проводником (по двойному щелчку) ярлыка или исполняемого файла C:\WINDOWS\System32\cmd.exe.

В результате запуска откроется окно командной строки с приглашением к вводу команд. Приложение cmd.exe часто называют командным процессором или интерпретатором команд, а его основное окно – консолью Windows.

Результаты выполнения команд отображаются в окне консоли Windows и нередко зависят от наличия достаточных прав у текущего пользователя. Для выполнения команд в контексте учетной записи Администратора в операционных системах Windows Vista - Windows 10 необходимо использовать режим Запуск от имени Администратора ПКМ.

Вся информация в компьютере хранится в файлах. Файл – это логически связанная совокупность данных (программ, текстов, изображений и т.д.) определенной длины, имеющая имя. Каждый файл имеет обозначение, которое состоит из двух частей: имени и расширения. Хотя расширение файла является необязательным, его использование удобно для классификации файлов по типу, например:

.com, .exe – программы, которые могут быть выполнены;

.bat – пакетные командные файлы;

.bak - резервные копии;

.txt - текстовые файлы.

В среде есть зарезервированные имена устройств, которые нельзя использовать в качестве имени файла:

PRN - принтер; NUL - «пустое» устройство;

LPT1 - LPT3 - устройства, присоединяемые к параллельным портам;

COM1 - COM4 - устройства, связанные с последовательными асинхронными портами;

CON - при вводе информации - клавиатура, при выводе - экран;

Каталог – это специальное место на диске, в котором хранятся имена файлов, сведения о размерах, времени последнего редактирования, атрибуты и т.д. Один и тот же файл на диске может быть зарегистрирован только в одном каталоге. Текущим называется каталог, с которым в настоящий момент производится работа. По умолчанию команды DOS ищут нужные файлы в текущем каталоге. Если используется файл не из текущего каталога, необходимо указать путь к файлу, например:

c:\program files\far\far.exe - файл far.exe в подкаталоге far каталога program files.

Для указания группы файлов из одного каталога можно употреблять символы «*» (любое число любых символов) и «?» (один произвольный символ), например:

***.txt** -все файлы с расширением .txt;

d*.e* -все файлы с именем, начинающимся на d, и расширением, начинающимся на букву e;

a??.* -файлы любого расширения с длиной имени в три символа и начинающихся на букву a.

Подробную информацию по любой команде можно получить одним из следующих способов:

help/имя_команды/

имя_команды /?.

Команды состоят из имени и параметров, разделенных пробелами. Далее при записи формата команд будет принято, что параметры, заключенные в квадратные скобки, не являются обязательными.

При вводе команд можно пользоваться следующими клавишами для редактирования вводимой команды:

F3 – вызов в командную строку предыдущей команды;

Esc– очистка всей командной строки.

Выполнение любой команды можно прекратить, нажав комбинацию клавиш **Ctrl+C** или **Ctrl+Break**.

Если команда выдает слишком много информации на экран, можно воспользоваться комбинацией **Ctrl+S** для приостановки выдачи. Повторное нажатие **Ctrl+S** возобновит выдачу.

Для очистки экрана монитора используется команда **cls**.

2. Команды для работы с каталогами

Смена текущего дисковогода. Для этого необходимо набрать требуемое имя дисководов и двоеточие, например, **c:.**

Смена текущего каталога. Формат команды:

CD [дисковод:] [путь]

дисковод - имя накопителя, где находится искомый каталог; после имени необходимо поместить двоеточие; если имя не указано, то считается, что сменяется каталог на текущем диске;

путь - указывает путь к новому каталогу; путь должен указывать на существующий каталог;

без параметров - выводит имя текущего каталога на текущем накопителе.

Пример:

cd - переход в корневой каталог на текущем диске.

cd\util\nc - переход в подкаталог nc каталога util на текущем диске;

cd c:\windows

Просмотр каталога. Формат команды:

DIR [дисковод:][путь\][имя_файла] [/P] [/W] [/A]

(dir не понимает отдельных имен)

В имени файла можно употреблять символы * и ?. Если имя файла не задано, то выводится оглавление каталога, иначе выводятся сведения о данном файле. Если в команде не указаны дисковод или путь, то подразумевается текущий дисковод и текущий каталог.

/P- выводит список содержимого каталога постранично; при окончании вывода очередной страницы система ждет нажатия любой клавиши для продолжения вывода;

/W - задает вывод только краткой информации (имен);

/A - выводит все подкаталоги и файлы (включая скрытые и системные);

без параметров - выводит содержимое текущего каталога.

Для каждого файла из каталога сообщается его имя, расширение, размер файла в байтах, дата, время создания или последнего обновления. Подкаталоги обозначаются <DIR>, в последних строках сообщается размер свободного пространства на диске.

Пример:

dir - вывести оглавление текущего каталога;

dir *.exe- вывести сведения о файлах с расширением .exe из текущего каталога.

Создание каталога. Формат команды:

MD [диск:][путь\] имя_каталога

Пример:

md info- создание подкаталога games в текущем каталоге;

md c:\testmd- создание каталога testmd в корневом каталоге диска c:.

Уничтожение каталога. Формат команды:

RD [диск:][путь\]

Удаляемый каталог должен быть пустым, поэтому перед вызовом команды RD необходимо удалить все его файлы и подкаталоги.

Пример:

rd info- удаление подкаталога info в текущем каталоге;

rd c:\testmd- удаление подкаталога testmd в корневом каталоге диска c.

Копирование каталога. Формат команды:

copy [source] destination

3. Команды для работы с файлами

Копирование файлов. Формат команды:

COPY имя_файла [+имя_файла] имя_копии [/V] [/Y | /-Y]

+ - операция объединения файлов;

/V - проверка файлов после копирования;

/Y - перезапись совпадающих по имени файлов без предупреждения;

/-Y- запрос разрешения на перезапись совпадающих по имени файлов.

Если в параметре *имя_копии* отсутствует имя файла, то имена файлов при копировании не меняются. Если в параметре *имя_копии* задано имя файла, то он указывает новое имя копируемого файла. В качестве имени файла можно использовать шаблон с символами «*» и «?». Использование данных символов в имени файла параметра *имя_копии* означает, что соответствующие символы в именах копируемых файлов при копировании не меняются.

Пример:

cd c:\student\

copy con infocon.txt - ввод содержимого консоли в файл infocon.txt.

copy con infobak.txt - ввод содержимого консоли в файл infobak.txt.

Для разделения строк вводимого с консоли файла, необходимо нажимать клавишу Enter, а для окончания ввода – F6 (или Ctrl+Z) и Enter.

(Текст infocon.txt:

Консоль это:

клавиатура при вводе информации,

экран при выводе информации

Текст infobak.txt:

Файл с расширением .bak– это копия файла, сделанная перед его изменением.)

copy infocon.txt infocon.mfc – в текущем каталоге создается копия файла infocon.txt с новым именем infocon.mfc.

md infocon

copy infocon. *infocon- копирование файлов infocon.txt и infocon.mfc в папку infocon.

Переименование файлов. Формат команды:

REN [диск:][путь/] имя_файла новое_имя

В качестве имени файла можно использовать символы «*» и «?». Если диск и путь не указаны, то подразумеваются текущие диск и каталог.

Пример:

ren *.txt *.mfc- переименование всех файлов с расширением .txt в текущем каталоге в файлы с расширением .mfc.

Удаление файлов. Формат команды:

DEL [диск:][путь/] имя_файла [/P]

/P - запрашивает подтверждение перед каждым удалением. В имени файла можно употреблять символы «*» и «?».

Пример:

del *.bak- удаление всех резервных копий из текущего каталога.

Вывод содержимого файла на экран. Формат команды:

TYPE имя_файла

Если команда выдает слишком много информации на экран, можно воспользоваться комбинацией **Ctrl+S** для приостановки выдачи. Повторное нажатие **Ctrl+S** возобновит выдачу.

Пример:

type c:\student\infocon\infocon.txt- вывод на экран файла infocon.txt.

4. Команды общесистемного назначения

Изменение вида приглашения. Формат команды:

PROMPT[текст]

В тексте, указываемом в команде prompt, можно использовать следующие сочетания символов:

\$p – текущий диск и каталог;	\$h – удаление предыдущего символа;
\$n – текущий диск;	\$g – символ «>»;
\$d – текущая дата;	\$l – символ «<»;
\$t – текущее время;	\$\$ – символ «\$»;
\$v – текущая версия операционной системы;	\$b – символ « »
\$s – пробел;	\$_ - переход на новую строку

Пример:

prompt\$p\$g – устанавливает приглашение DOS, содержащее информацию о текущем диске и каталоге и символ «>»;

prompt\$t\$h\$h\$h\$h\$h\$h\$h\$h\$p\$g – устанавливает приглашение DOS в виде текущего времени без секунд и миллисекунд, текущего диска и каталога и символа «>».

5. Пакетные командные файлы

В процессе работы с компьютером часто требуется повторно выполнять некоторую последовательность команд. Данные команды можно записать в специальный пакетный командный файл с расширением .bat и в дальнейшем для выполнения требуемой последовательности будет достаточно вызвать данный командный файл. При запуске файла расширение (.bat) можно не указывать.

Параметры

Командные файлы могут использовать до 9 параметров, указываемых в строке вызова файла, например,

getmark Ivanov (командный файл *getmark*, параметр *–Ivanov*)

В тексте файла параметры обозначаются символами %1-%9. Если в командной строке при вызове командного файла задано меньше девяти параметров, то лишние символы замещаются пустыми строками. Параметр %0 ссылается на имя командного файла.

Вывод команд и сообщений на экран

По умолчанию команды пакетного файла выводятся на экран перед выполнением. Если в файл вставить команду **echo off**, то выполняемые за ней команды не будут выводиться на экран. Для того, чтобы избежать вывода на экран отдельной командной строки, перед текстом команды ставится символ @. Например, чтобы избежать вывода на экран текста *echo off*, следует записать его в следующем виде: **@echo off**. Команда **echo** также используется для того, чтобы вывести на экран какое-то сообщение, например,

Echo Введите группу, в которой учится Иванов

6. Создание загрузочной флешки в командной строке

DISKPART - запускаем программу, командный интерпретатор текстового режима, который позволяет управлять объектами (дисками, разделами или томами) с помощью сценариев или прямым вводом команд из командной строки.

list disk - отображаем список дисковых накопителей, подключенных к персональному компьютеру.

select disk № - выбираем диск под номером "№", у которого указано, что он является съемным накопителем-флешкой.

clean - очищает все данные со съемного носителя - флешки.

create partition primary - создаем первичный раздел.

select partition 1 - выбираем созданный раздел.

active - делаем раздел активным.

format fs=NTFS - форматируем флешку в файловой системе NTFS.

assign letter=T - если необходимо, то таким образом можно присвоить букву для флешки.

Exit - выход из программы DISKPART.

ЗАГРУЗОЧНАЯ ФЛЕШКА СОЗДАНА!

2. Основная часть.

Задание

1. Открыть редактор командной строки используя сочетание клавиш "Win+R" для вызова окна "Выполнить" и ввести в пустое поле значение "cmd".
1. В своей папке D:/P121 создать каталог с именем COMANDS.
2. В этом каталоге создать каталоги: с именем KATALOGS, с именем DISCS, с именем FILES.
3. В каталоге DISCS создать файлы: с именем disk.txt, в который записать: свою фамилию, имя, отчество, группу, определение <Диск - это:>; с именем com_disk.txt, в котором описать команды для работы с дисками.
4. В каталоге KATALOGS создать файл org.txt, в котором написать определение каталога, и каталога: с именем SOZDAN_K, с именем UDALEN_K, с именем PROSM_K.

5. В каталоге SOZDAN_K создать файл с именем sozd.txt, в котором описать формат команды md и её назначение.
6. В каталоге UDALEN_K создать файл с именем udal.txt, в котором описать формат команды удаления каталога и условия удаления.
7. В папке PROSM_K создать файл с именем soder.txt, в котором описать все известные ключи команды DIR и их назначение.
8. В каталоге FILES создать файлы: с именем com_file.txt, в котором описать команду удаления файлов; с именем opr.txt, в котором написать определение файла <Файл - это:>; с именем kop.txt, в котором написать формат команды копирования файлов.
9. Объединить файлы kop.txt, opr.txt, com_file.txt папки FILES в один файл с именем obed.txt
10. Содержимое каталога FILES перенаправить в файл cod_file.txt
11. Переименовать файл sozd.txt в файл с именем proba.txt.
12. Скопировать из каталога FILES все файлы с расширением txt в каталог DISCI.
13. Установить текущим каталогом COMANDS и построить в нем дерево каталогов.
14. Удалить каталог PROSM_K.
15. Сделать рабочим каталогом COMANDS и построить в него дерево каталогов.
16. Удалить в каталоге FILES все файлы, имя которых начинается на букву С.
17. Добавить в файл obed.txt из каталога DISCS текущее время.
18. В строке приглашения среды вывести системную дату.
19. Изменить приглашение системы (выбор самостоятельно)
20. Написать пакетный файл с именем start для запуска любой программы с диска С:
(предварительно очистив экран, выдав на экран сообщение о назначении запускающего файла, изменить цвет фона и символов) и сохранить в папке COMANDS. В пакетном файле должна быть указана фамилия автора.
21. Показать результат преподавателю вместе с электронным отчетом, после получения оценки удалить все созданное на диске.
22. Очистить экран от служебных записей.

Отчет должен содержать:

- титульный лист;
- цель работы;
- ход работы - протокол диалога пользователя с ЭВМ при выполнении задания.
- выводы по работе

3. Заключительная часть.

Проверьте свои знания, ответив на контрольные вопросы:

1. Понятие операционной системы, ее назначение.
2. Как воспользоваться справкой по командам среды?
3. Какие команды операционной системы называются внутренними, а какие внешними?
4. Какой дисковод называется текущим?
5. Как обозначаются различные устройства в командной строке?
6. Как перейти с одного диска на другой?

Практическая работа № 2

Работа с операционной оболочкой Double Commander

Цель работы: *получить основные навыки работы с программой Double Commander: освоить различные способы выполнения операций с файлами и папками.*

1. Основные теоретические сведения



Происхождение

Double Commander – продукт отечественного программиста Александра Коблова (Alexx2000@mail.ru), основанный на Sexi Commander

Использование

Есть несколько способов получить Double Commander:

- Страница [релизов](#) на GitHub.
- Страница [релизов](#) на SourceForge.
- Double Commander доступен в репозиториях многих дистрибутивов Linux и в коллекции портов FreeBSD.
- У Double Commander есть собственный [репозиторий](#) для нескольких дистрибутивов Linux (официальные репозитории дистрибутивов Linux не всегда содержат текущую (последнюю) версию программы).
- И вы можете [собрать](#) Double Commander самостоятельно.

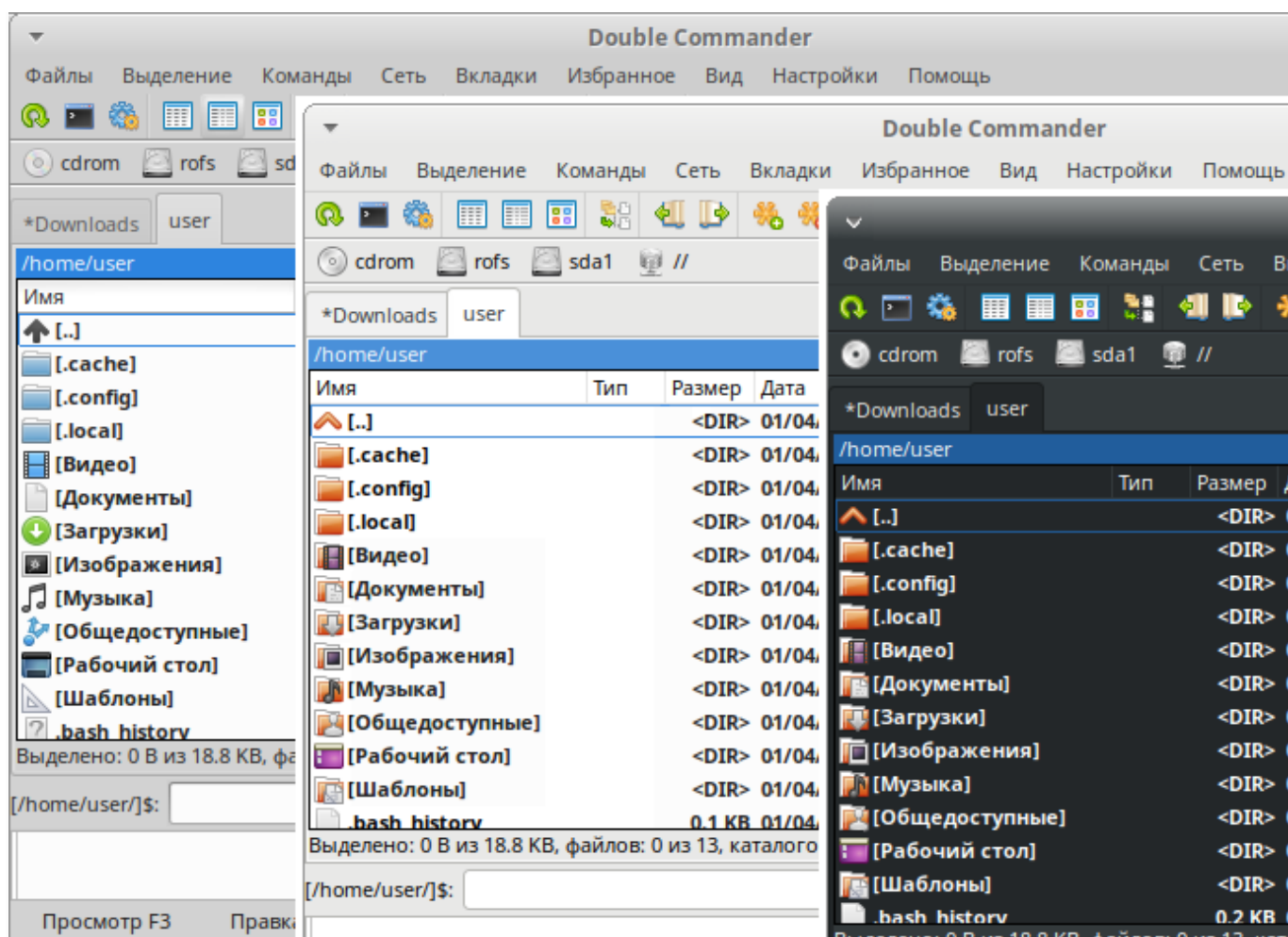
Сначала вы должны определиться с операционной системой (Windows, Linux, macOS^(*)), архитектурой процессора, а затем можете выбрать версию для загрузки.

Для Windows доступны инсталляторы (.exe и .msi, Inno Setup и MSI соответственно) и архив ZIP с портативной версией.

Для Linux существуют версии с использованием библиотек GTK и Qt. Также доступны портативные версии (скомпилированные двоичные файлы, упакованные XZ, для запуска используйте `doublecmd.sh`).

При первом запуске Double Commander автоматически определит системный язык и установит его для интерфейса. DC переведён на многие языки. Язык можно изменить в [Настройки > Параметры... > Язык](#).

Double Commander определяет и использует текущую системную тему. Выглядит это так (Ubuntu, три темы и два набора значков):



На изображении видны левая файловая панель, командная строка и ниже окно протокола (оно показывает все действия файлового менеджера). Double Commander также поддерживает тёмный режим в Windows (Windows 10 1809 и новее). *(Бета-версия)*

Double Commander использует API плагинов Double Commander, так что под Windows вы можете использовать плагины для Double Commander (многие из них совместимы с DC). Описание типов плагинов смотрите в разделе настроек Плагины. DC поставляется с несколькими предустановленными плагинами, другие плагины могут быть установлены вручную.

2. Основная часть.

Задание

Работа с файлами и папками в панелях

1. Найдите на рабочем столе ярлык и запустите программу *Double Commander*.
2. Найдите левую и правую панели, просмотрите содержимое корневого каталога. Для этого нажмите кнопку с символом «\» в окне выбора дисков.
3. Запустите программу Блокнот, в блокноте наберите текст: «Я сейчас изучаю возможности программы *Double Commander*». Сохраните файл с именем Double в папке 2 курс.
4. Для просмотра содержимого этого текстового файла в левой панели окна Double Commander перейдите в папку 2 курс, выделите файл Double.txt. Среди функциональных клавиш найдите кнопку «Просмотр» или нажмите F3. Закройте окно просмотра.
5. В правой панели откройте папку 2 курс. Для создания каталога найдите кнопку «Каталог» или нажмите F7. В появившемся окне укажите имя ЭКСПЕРИМЕНТ и нажмите ок. Откройте эту папку двойным щелчком левой кнопкой мыши.

6. Для того чтобы переместить файл Double.txt. из папки 2 курс в папку ЭКСПЕРИМЕНТ, перейдите в левую панель, щёлкнув левой кнопкой мыши или нажав клавишу Tab. Выделив файл Double.txt, нажмите кнопку «Перемещение» или F6.
7. Щелчком правой кнопкой мыши по файлу Double.txt вызовите контекстное меню. Выполните команды – создать ярлык и добавить в архив Double.rar.
8. Закройте папку ЭКСПЕРИМЕНТ, для этого выполните двойной щелчок по значку .

Меню Файл и Вид

1. Откройте пункт меню «Файл» и выберите команду «Свойства файла/каталога». Просмотрите размер вашей папки, тип, дату создания, атрибуты.
2. Заархивируйте папку ЭКСПЕРИМЕНТ, поместив архив в папку P22X. Для этого:
 - в одной из панелей выберите и откройте папку P22X (эта папка - получатель);
 - в другой панели выделите щелчком левой кнопкой мыши папку ЭКСПЕРИМЕНТ (источник);
 - в пункте меню Файл выберите упаковать;
 - в появившемся окне необходимо выбрать архиватор (например, zip);
 - нажать ok.
1. Разархивируйте полученный архив в папку 3 курс, предварительно открыв эту папку на одной из панелей (получатель) и примените команду распаковать для выделенного архива (источника).
2. Примечание. Можно создать многотомный архив (в случае, если архив более 1.44 Mb). Тогда устанавливают напротив этого пункта флажок.
3. Поиск файла (например, Double.txt) выполняют с помощью команды Поиск файлов в пункте меню КОМАНДЫ. В окне поиска необходимо ввести Double.txt. В окне результата поиска можно не только сразу же перейти к искомому файлу (или папке), можно вынести найденное на панель, а также просмотреть найденный файл.
4. Вызовите окно Сведения о системе. (Выберите пункт КОМАНДЫ и сведения о системе).
5. Откройте пункт меню ВИД. Примените различные способы отображения файлов и папок в панелях Double Commander. Организуйте отбор файлов по шаблону Text Files с помощью команды ФИЛЬТР, убедитесь в том, что в панелях видны только текстовые файлы.

Меню Сеть

1. Создайте копию каталога ЭКСПЕРИМЕНТ и назовите его СЕТЕВОЙ ЭКСПЕРИМЕНТ.
2. Откройте меню Сеть, выберите команду Открыть общий доступ к каталогу и настройте общий доступ к каталогу СЕТЕВОЙ ЭКСПЕРИМЕНТ.
3. Создайте сетевой диск с именем Z из каталога СЕТЕВОЙ ЭКСПЕРИМЕНТ.

Оформить электронный отчет по практическому занятию.

Отчет должен содержать:

- титульный лист;
- цель работы;
- ход работы - протокол диалога пользователя с ЭВМ при выполнении задания.
- выводы по работе

3. Заключительная часть.

Проверьте свои знания, ответив на контрольные вопросы:

1. Опишите структуру окна Double Commander.
2. Укажите сходство Double Commander и программы Проводник.
3. Какие способы выделения в панелях Double Commander вы знаете?
4. Как осуществить операции копирования, перемещения и удаления?
5. Как узнать свойства файла?
6. Продемонстрируйте, как заархивировать папку.
7. Как разбить и собрать файл? Когда используются эти команды?
8. Как найти папку и поместить её на панель?

9. Как отобразить файлы и папки в виде дерева на одной из панелей Double Commander?
10. Как выполнить сортировку по дате создания?

Практическое занятие №3

Исследование работы планировщиков процессов в мультипрограммных операционных системах

ЦЕЛЬ РАБОТЫ:

- 1) знакомство с программой имитирующей модель управления работой планировщика процессов в операционной системе;
- 2) сравнение методов планирования по различным критериям.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:

Операционные системы различаются

- особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами),
- особенностями использованных методов проектирования,
- типами аппаратных платформ,
- критериями эффективности,
- особенностями реализации сетевых решений
- и многими другими свойствами.

Мультипрограммирование, или многозадачность, — это способ организации вычислительного процесса, при котором на одном процессоре выполняются сразу несколько программ (создается видимость одновременного выполнения нескольких программ).

Мультипрограммирование применяется для повышения эффективности вычислительной системы.

Эффективность же может пониматься как

- общая пропускная способность вычислительной системы;
- удобство работы пользователей, например, возможность интерактивной работы для нескольких пользователей или возможность работы одного пользователя с несколькими приложениями на одной машине;
- реактивность системы – то есть способность системы выдерживать заранее заданные (возможно, очень короткие) интервалы времени между запуском программы и получением результата.

В зависимости от выбранного критерия эффективности операционные системы делятся на:

- системы пакетной обработки (например, ОС ЕС),
- системы разделения времени (UNIX, VMS),
- системы реального времени (QNX, RT/11).

Системы пакетной обработки служат для решения задач в основном вычислительного характера, не требующих быстрого получения результатов.

Главной целью и критерием эффективности таких систем является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени.

Схема функционирования систем пакетной обработки данных:

- в начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам;

- из этого пакета заданий формируется мультипрограммная смесь, то есть множество одновременно выполняемых задач. Для одновременного выполнения выбираются задачи, предъявляющие отличающиеся требования к ресурсам, так, чтобы обеспечивалась сбалансированная загрузка всех устройств вычислительной машины; (в мультипрограммной смеси желательно одновременное присутствие вычислительных задач и задач с интенсивным вводом-выводом).

Таким образом, выбор нового задания из пакета заданий зависит от внутренней ситуации, складывающейся в системе, то есть выбирается "выгодное" задание.

В таких ОС невозможно гарантировать выполнение того или иного задания в течение определенного периода времени.

В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач.

Взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что он приносит задание, отдает его диспетчеру-оператору, а в конце дня после выполнения всего пакета заданий получает результат. Очевидно, что такой порядок снижает эффективность работы пользователя.

Основной недостаток систем пакетной обработки - изоляция пользователя-программиста от процесса выполнения его задач.

Системы разделения времени (интерактивные) призваны исправить недостаток систем пакетной обработки данных.

Каждому пользователю системы разделения времени предоставляется терминал, с которого он может вести диалог со своей программой.

Каждой задаче выделяется только квант процессорного времени,

и ни одна задача не занимает процессор надолго, и время ответа оказывается приемлемым.

Если квант выбран достаточно небольшим, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них единолично использует машину.

Системы разделения времени обладают меньшей пропускной способностью, чем системы пакетной обработки, так как

- на выполнение принимается каждая запущенная пользователем задача, а не та, которая "выгодна" системе,
- увеличивается время работы, так как выполняется более частое переключение процессора с задачи на задачу.

Таким образом, критерием эффективности систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.

Системы реального времени применяются для управления различными техническими объектами, такими, например, как станок, спутник, научная экспериментальная установка или технологическими процессами, такими, как гальваническая линия, доменный процесс и т.п.

Существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа, управляющая объектом, в противном случае может произойти авария: спутник выйдет из зоны видимости, экспериментальные данные, поступающие с датчиков, будут потеряны, толщина гальванического покрытия не будет соответствовать норме.

Таким образом, критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия).

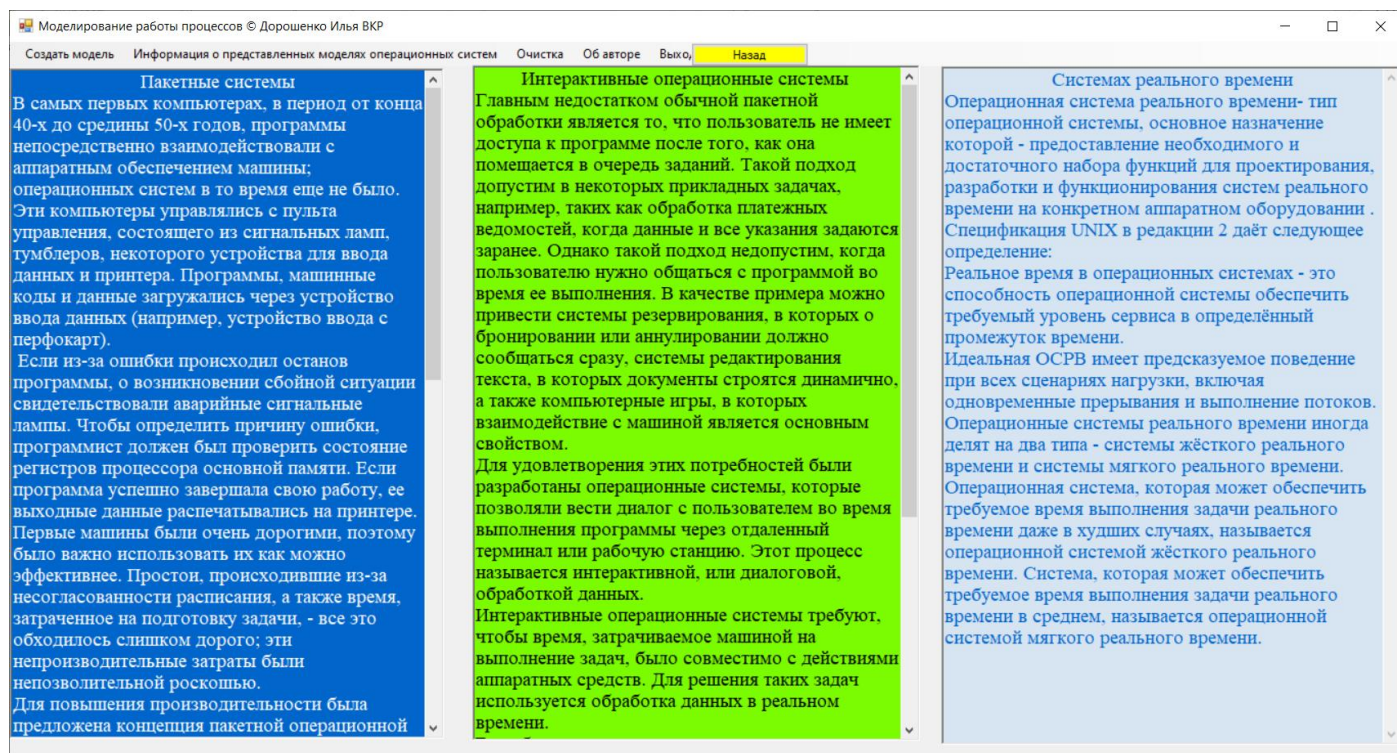
Это время называется временем реакции системы, а соответствующее свойство системы - реактивностью.

Для этих систем мультипрограммная смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

Замечание. Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме пакетной обработки, а часть - в режиме реального времени или в режиме разделения времени. В таких случаях режим пакетной обработки часто называют фоновым режимом.

ХОД РАБОТЫ:

1. Скачать с диска [приложение planprocess](#) для исследования алгоритмов планирования процессов;
2. Выбрать из таблицы вариантов свой вариант по списку алгоритм по каждому виду систем;
3. Ввести для каждого алгоритма данные о пяти процессах в системе;
4. Сравнить эффективность работы алгоритмов и выявить процесс, который выполнится самым первым (почему?), последним.



СОСТАВ ОТЧЕТА:

Титульный лист

Цель работы

Ход работы: сведения о типе системы по критериям эффективности, сведения об алгоритме планирования процессов, скриншоты введенных данных о процессах, скриншоты диаграммы/таблицы выполнения процессов

Выводы.

Отчет предъявить преподавателю для оценивания

Таблица вариантов

№	Алгоритм систем пакетной обработки	Алгоритм систем разделения времени	Алгоритм систем реального времени
1.	Неприоритетные алгоритмы FIFO	Циклическое	статические
2.	Неприоритетные алгоритмы SJF	Приоритетное	динамические
3.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Циклическое	динамические
4.	Неприоритетные алгоритмы FIFO	Приоритетное	статические
5.	Неприоритетные алгоритмы SJF	Циклическое	статические

6.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Приоритетное	динамические
7.	Неприоритетные алгоритмы SJF	Циклическое	динамические
8.	Неприоритетные алгоритмы FIFO	Приоритетное	динамические
9.	Неприоритетные алгоритмы SJF	Циклическое	статические
10.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Приоритетное	статические
11.	Неприоритетные алгоритмы FIFO	Циклическое	динамические
12.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Приоритетное	динамические
13.	Неприоритетные алгоритмы FIFO	Циклическое	динамические
14.	Неприоритетные алгоритмы SJF	Приоритетное	статические
15.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Циклическое	статические
16.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Приоритетное	динамические
17.	Неприоритетные алгоритмы SJF	Циклическое	статические
18.	Неприоритетные алгоритмы FIFO	Приоритетное	статические
19.	Неприоритетные алгоритмы SJF	Циклическое	динамические
20.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Приоритетное	статические
21.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Циклическое	статические
22.	Неприоритетные алгоритмы SJF	Приоритетное	статические
23.	Неприоритетные алгоритмы FIFO	Циклическое	динамические
24.	Приоритетные алгоритмы – по наименьшему алгоритму планирования	Приоритетное	динамические
25.	Неприоритетные алгоритмы FIFO	Циклическое	динамические

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Дайте определение «процесс», «планировщик процесса».
2. В каких состояниях может находиться процесс?
3. Какую информацию о процессе хранит операционная система?
4. Какие очереди процессов формирует система? Каков принцип обслуживания очереди?
5. Дайте определение мультипрограммирование.
6. По каким критериям классифицируются операционные системы?
7. Какие приоритеты может получать процесс в системе?
8. Какие приоритетные алгоритмы планирования вам известны? Неприоритетные?

Практическое занятие №4

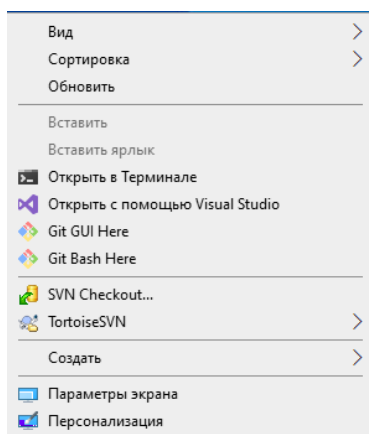
Настройка рабочего стола. Настройка системы с помощью Панели управления

ЦЕЛЬ РАБОТЫ:

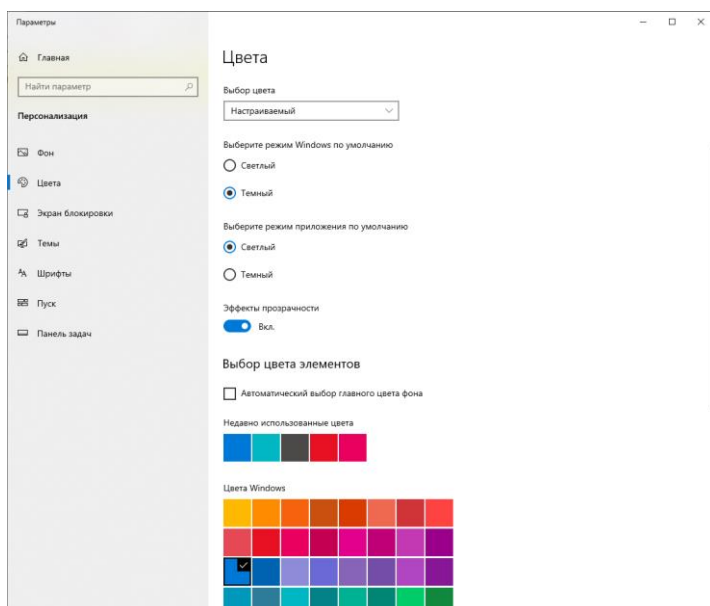
- 3) знакомство с настройками операционной системы;
- 4) получение практических навыков настройки интерфейса системы.

ХОД РАБОТЫ:

1. Ознакомьтесь с содержанием раздела «[Основы работы в Windows](#)» [Руководство по использованию OS Windows 8.1 :: Школа Windows \(windows-school.ru\)](#)
2. Вызовите контекстное меню на свободном месте Рабочего стола. Выберите команду Персонализация. Внесите в отчет информацию о настройках фона рабочего стола



3. В настройках цвета установите темный режим по умолчанию для windows и светлый для приложений. Включите эффекты прозрачности и смените цвет элементов на любой фиолетовый или зеленый.



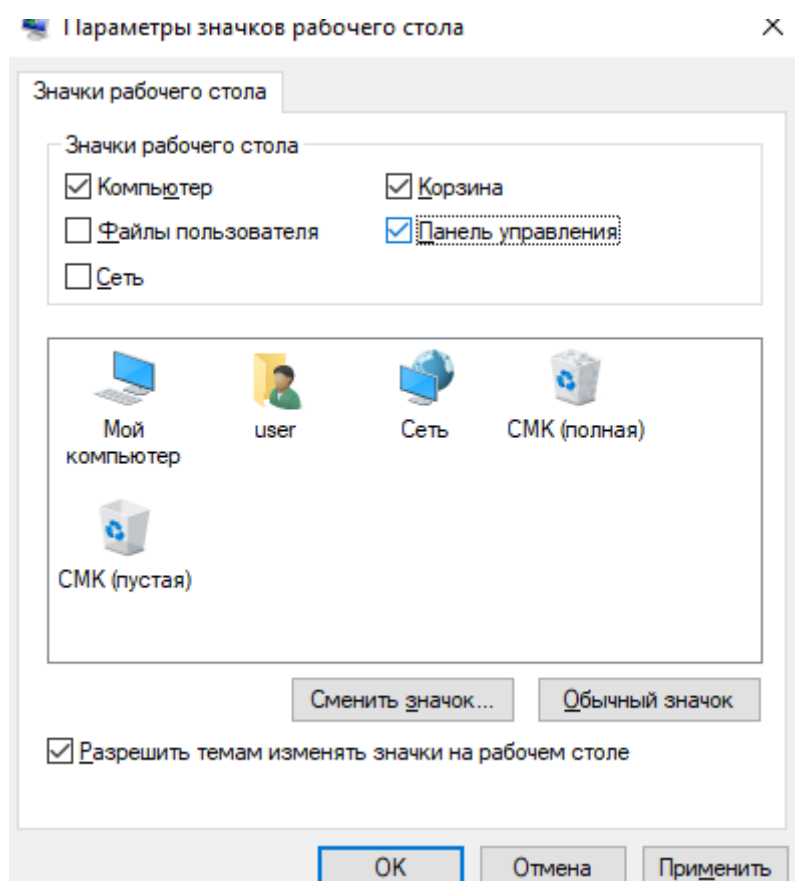
4. Выберите фото для экрана блокировки



Отключите подсказки на экране блокировки

5. Смените курсор мыши в темах на рукописный ввод

6. Настройте параметры значков рабочего стола, добавьте панель управления



7. Установите шрифт Cambria или Segoe UI, отфильтровав предварительно кириллические шрифты.

8. В меню пуск доюавьте отображение наиболее часто используемых приложений.

9. Установить положение панели задач сверху экрана

10. Внести в отчет информацию о каждой группе настроек панели управления: система, устройства, сеть и интернет, приложения, учетные записи, время и язык. Сами настройки не менять. Сделать выводы по работе.

Практическая работа № 5

Мониторинг процессов

Цель работы: изучение работы приложения для мониторинга процессов и приобретение практических навыков его использования.

1. Основные теоретические сведения

Process Monitor — это расширенный инструмент мониторинга для Windows, который в реальном времени показывает файловую систему, реестр и активность процессов/поток. Он сочетает в себе функции двух устаревших утилит Sysinternals, *Filemon* и *Regmon*, и добавляет обширный список улучшений, включая богатую и неразрушающую фильтрацию, комплексные свойства событий, такие как идентификаторы сеансов и имена пользователей, надежную информацию о процессах, полные стеки потоков со встроенными символами. поддержка каждой операции, одновременная запись в файл и многое другое. Его уникальные мощные функции сделают Process Monitor основной утилитой в наборе инструментов для устранения неполадок вашей системы и поиска вредоносных программ.

Обзор возможностей монитора процессов

Process Monitor включает в себя мощные возможности мониторинга и фильтрации, в том числе:

- Больше данных, собираемых для входных и выходных параметров операции.
- Неразрушающие фильтры позволяют устанавливать фильтры без потери данных.
- Захват стеков потоков для каждой операции во многих случаях позволяет определить основную причину операции.
- Надежный сбор сведений о процессе, включая путь к образу, командную строку, идентификатор пользователя и сеанса.
- Настраиваемые и перемещаемые столбцы для любого свойства события.
- Фильтры можно установить для любого поля данных, включая поля, не настроенные как столбцы.
- Усовершенствованная архитектура журналирования масштабируется до десятков миллионов зафиксированных событий и гигабайт данных журналов.
- Инструмент дерева процессов показывает взаимосвязь всех процессов, на которые есть ссылки в трассировке.
- Собственный формат журнала сохраняет все данные для загрузки в другой экземпляр Process Monitor.
- Всплывающая подсказка процесса для удобного просмотра информации об изображении процесса.
- Подробная подсказка обеспечивает удобный доступ к отформатированным данным, которые не помещаются в столбец.
- Отменяемый поиск
- Регистрация всех операций во время загрузки

Лучший способ ознакомиться с функциями Process Monitor — прочитать файл справки, а затем просмотреть каждый из его пунктов меню и опций в работающей системе.

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time of D...	Process Name	PID	Operation	Path	Result	Detail
10:49:15,6...	ctfmon.exe	11784	RegQueryKey	HKLM	SUCCESS	Query: H
10:49:15,6...	ctfmon.exe	11784	RegOpenKey	HKLM\Software\Microsoft\Input\Locale...	SUCCESS	Desired /
10:49:15,6...	ctfmon.exe	11784	RegQueryValue	HKLM\SOFTWARE\Microsoft\Input\Lo...	SUCCESS	Type: RE
10:49:15,6...	ctfmon.exe	11784	RegCloseKey	HKLM\SOFTWARE\Microsoft\Input\Lo...	SUCCESS	
10:49:15,6...	ctfmon.exe	11784	RegQueryKey	HKLM	SUCCESS	Query: H
10:49:15,6...	ctfmon.exe	11784	RegOpenKey	HKLM\Software\Microsoft\Input\Settings	SUCCESS	Desired /
10:49:15,6...	ctfmon.exe	11784	RegQueryKey	HKCU	SUCCESS	Query: H
10:49:15,6...	ctfmon.exe	11784	RegOpenKey	HKCU\Software\Microsoft\Input\Settings	NAME NOT FOUND	Desired /
10:49:15,6...	ctfmon.exe	11784	RegQueryKey	HKLM\SOFTWARE\Microsoft\Input\Se...	SUCCESS	Query: H
10:49:15,6...	ctfmon.exe	11784	RegOpenKey	HKLM\SOFTWARE\Microsoft\Input\Se...	SUCCESS	Desired /
10:49:15,6...	ctfmon.exe	11784	RegQueryValue	HKLM\SOFTWARE\Microsoft\Input\Se...	SUCCESS	Type: RE
10:49:15,6...	ctfmon.exe	11784	RegCloseKey	HKLM\SOFTWARE\Microsoft\Input\Se...	SUCCESS	
10:49:15,6...	ctfmon.exe	11784	RegCloseKey	HKLM\SOFTWARE\Microsoft\Input\Se...	SUCCESS	
10:49:15,6...	ctfmon.exe	11784	RegQueryKey	HKLM	SUCCESS	Query: H
10:49:15,6...	ctfmon.exe	11784	RegOpenKey	HKLM\Software\Microsoft\Input\Settings	SUCCESS	Desired /

Showing 7 491 of 126 500 events (5.%) Backed by virtual memory

Process Tree

☐ Only show processes still running at end of current trace
☒ Timelines cover displayed events only

Process	Description	Image Path	Life Time	Company	Owner	Comr
Idle (0)	Idle					
System (4)	System				NT AUTHORITY\...	
Registry (124)	Registry				NT AUTHORITY\...	
smss.exe (492)	Диспетчер сеан...	C:\Windows\Syst...		Microsoft Corporat...	NT AUTHORITY\...	\Syste
MemCompression (2136)	MemCompression				NT AUTHORITY\...	
csrss.exe (672)	Процесс исполн...	C:\Windows\syst...		Microsoft Corporat...	NT AUTHORITY\...	%Syste
wininit.exe (756)	Автозагрузка пр...	C:\Windows\syst...		Microsoft Corporat...	NT AUTHORITY\...	winini
services.exe (836)	Приложение слу...	C:\Windows\syst...		Microsoft Corporat...	NT AUTHORITY\...	C:\Wi
svchost.exe (524)	Хост-процесс дл...	C:\Windows\syst...		Microsoft Corporat...	NT AUTHORITY\...	C:\Wi
unsecapp.exe (9496)	Sink to receive as...	C:\Windows\syst...		Microsoft Corporat...	NT AUTHORITY\...	C:\Wi
mousocoreworker.exe (1...	MoUSO Core Wor...	C:\Windows\Syst...		Microsoft Corporat...	NT AUTHORITY\...	C:\Wi
wmiprvse.exe (7504)	WMI Provider Host	C:\Windows\syst...		Microsoft Corporat...	NT AUTHORITY\...	C:\Wi
StartMenuExperienceHo...		C:\Windows\Syst...			MK303-1\user	"C:\W
RuntimeBroker.exe (158)	Runtime Broker	C:\Windows\Syst...		Microsoft Corporat...	MK303-1\user	C:\Wi
SearchApp.exe (6732)	Search application	C:\Windows\Syst...		Microsoft Corporat...	MK303-1\user	"C:\W
RuntimeBroker.exe (944)	Runtime Broker	C:\Windows\Syst...		Microsoft Corporat...	MK303-1\user	C:\Wi
RuntimeBroker.exe (480)	Runtime Broker	C:\Windows\Syst...		Microsoft Corporat...	MK303-1\user	C:\Wi
ShellExperienceHost.exe	Windows Shell Ex...	C:\Windows\Syst...		Microsoft Corporat...	MK303-1\user	"C:\W

Description: Автозагрузка приложений Windows
Company: Microsoft Corporation
Path: C:\Windows\system32\wininit.exe
Command: wininit.exe
User: NT AUTHORITY\СИСТЕМА
PID: 756 Started: 17.10.2023 9:37:48

Go To Event Include Process Include Subtree

Close

Задание

1. Скачать монитор процессов и установить на ПК
<https://download.sysinternals.com/files/ProcessMonitor.zip>
2. Выполнить команды на панели инструментов Capture-Autoscroll-Clear- Capture. Какие команды они выполняют? Внести информацию в отчет.
3. Выполнить фильтрацию процессов:
 - - процессы с 32-разрядной архитектурой
 - - процессы NE с 32-разрядной архитектурой
 - - процессы содержат работу с сетью Network
 - - процессы разработчика Yandex LLC
 - - процессы других разработчиков кроме компании Yandex LLC
4. Вывести дерево процессов.
5. Вывести процессы, включая процессы windows.
6. Вывести только процессы, выполнявшие записи в реестр (registry activity)
7. Вывести только процессы file system activity
8. Вывести только процессы network activity
9. Вывести только процессы process and thread activity
10. Вывести только процессы profiling events
11. Через меню tools вывести системную информацию, суммарную активность процессов

Отчет должен содержать:

- титульный лист;
- цель работы;
- ход работы - протокол диалога пользователя с ЭВМ при выполнении задания.
- выводы по работе

3. Заключительная часть.

Проверьте свои знания, ответив на контрольные вопросы:

7. Понятие операционной системы, ее назначение.
8. Что такое процесс? Поток?
9. Какие состояния процессов вам известны?
10. Что такое ID процесса? PID?
11. Перечислите возможности программы Process Monitor.

Монитор ресурсов памяти

Цель работы: знакомство с функциями и структурами данных, используемыми для управления памятью.

Теоретические сведения:

Память является важнейшим ресурсом, требующим тщательного управления со стороны мультипрограммной операционной системы. Особая роль памяти объясняется тем, что процессор может выполнять инструкции протравы только в том случае, если они находятся в памяти. Память распределяется как между модулями прикладных программ, так и между модулями самой операционной системы.

Откачка и подкачка – механизм сброса образов неактивных процессов на диск и, при их активизации, считывания обратно в основную память. Файл откачки – пространство на диске, где система хранит образы откачанных процессов.

Смежное распределение памяти – отведение памяти для всех процессов из одной смежной области памяти. Регистр перемещения используется для преобразования логического адреса в физический (суммируется с логическим адресом).

При распределении памяти в общем случае ОС хранит список свободных и занятых областей. Возникает общая задача распределения памяти – разработка оптимального алгоритма удовлетворения запроса на выделение области памяти заданной длины. Три стратегии решения этой задачи – методы первого подходящего, наиболее подходящего и наименее подходящего. Первая и вторая эффективнее, третья позволяет уменьшить фрагментацию.

Фрагментация – дробление свободной памяти на мелкие несмежные области. Внешняя фрагментация – ситуация, при которой имеется большая область свободной памяти, но она не является непрерывной. Внутренняя фрагментация – выделение "лишних" участков свободной памяти вследствие применения стратегии выделения памяти с точностью до страницы. Внешняя фрагментация может быть ликвидирована с помощью компактировки – перемешивания или сдвига свободной памяти с целью преобразования ее в одну смежную область.

Страничная организация – метод управления памятью, при котором логическая и физическая память делится на страницы одинаковой длины (степень двойки). Физические страницы (фреймы), выделяемые для логических страниц процесса, могут располагаться произвольным образом. Для трансляции логических адресов в физические используется таблица страниц процесса. Логический адрес состоит из номера страницы и смещения внутри страницы. ОС хранит список свободных фреймов в основной памяти.

Сегментная организация памяти – стратегия распределения памяти сегментами переменной длины, каждый из которых выполняет определенную логическую функцию в программе как модуль программы или данных – стек, массив, подпрограмма и т.д.

При сегментной организации логический адрес имеет вид: (номер сегмента, смещение внутри сегмента). Организуется системная таблица сегментов, каждый элемент которой содержит базовый адрес сегмента, его длину и признаки защиты – validation-бит, определяющий корректность номера сегмента, биты защиты от записи, чтения и исполнения. Базовый регистр таблицы сегментов содержит начальный адрес таблицы сегментов, регистр длины таблицы сегментов содержит ее длину.

Перемещение при сегментной организации осуществляется динамически, во время выполнения программы.

Возможен общий доступ нескольких процессов к одному и тому же сегменту с одним логическим номером (разделяемые сегменты).

Стратегии распределения памяти при сегментной организации – методы первого подходящего и наиболее подходящего. Возможна внешняя фрагментация.

Для борьбы с фрагментацией в некоторых системах применяется смешанная, сегментно-страничная организация памяти, при которой для каждого сегмента организуется собственная таблица страниц.

Ход работы:

1. Вывести сведения из панели управления о системе и выделить сведения об оперативной памяти.

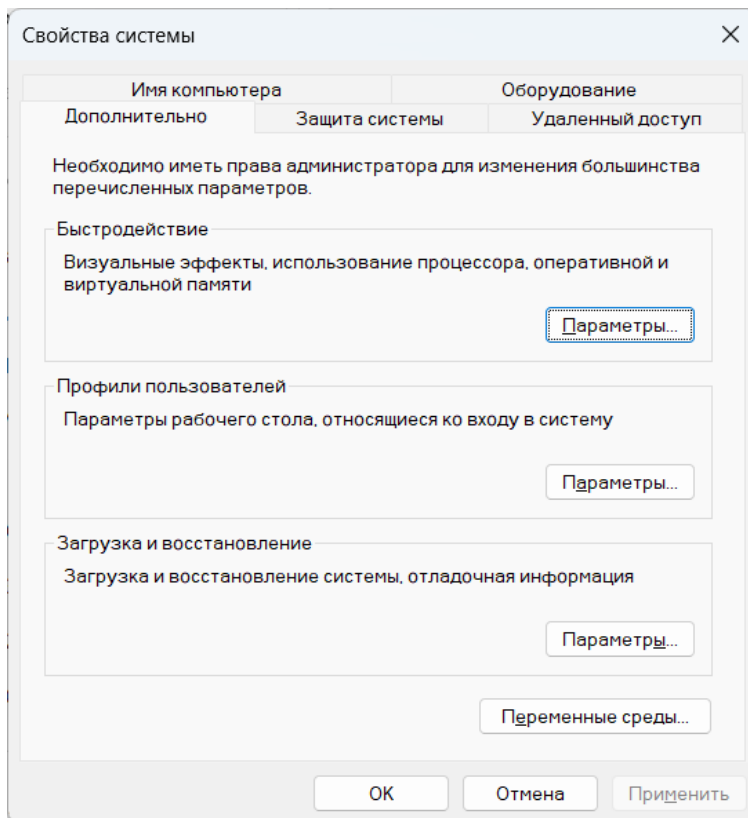
Характеристики устройства

Копировать

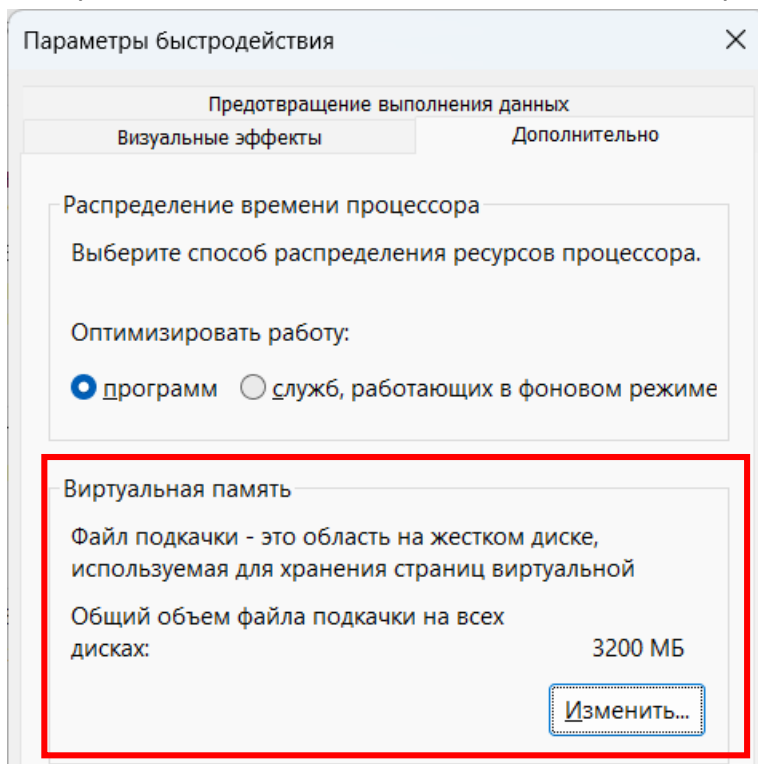
^

Имя устройства	
Процессор	
Оперативная память	16,0 ГБ (доступно: 15,7 ГБ)
Код устройства	
Код продукта	
Тип системы	64-разрядная операционная система, процессор x64
Перо и сенсорный ввод	Для этого монитора недоступен ввод с помощью пера и сенсорный ввод

2. Почему 32-разрядной системе столько памяти недоступно?
3. Вызовите окно выполнения и вызовите оснастку конфигурация системы msconfig. Внесите в отчет все вкладки окна конфигурации.
4. На вкладке загрузка вызовите окно дополнительных параметров, убедитесь, что НЕ задействован режим «Максимум памяти» (в противном случае на нужды системы будет выделен максимальный объем). Внесите информацию в отчет.
5. Вызовите дополнительные параметры системы.



6. На вкладке дополнительно в разделе быстродействие выбираем параметры быстродействия. Внесите в отчет все вкладки параметров быстродействия системы.



7. Опишите параметры, влияющие на быстродействие системы.
8. Опишите минимальный, текущий, максимальный доступный размер файла подкачки для вашей системы. Не меняйте его значение!!!
9. Используя сведения о системе в интерпретаторе командной строки, выведите сведения о:
 - Полный объем физической памяти:

- Доступная физическая память:
- Виртуальная память: Макс. размер:
- Виртуальная память: Доступна:
- Виртуальная память: Используется:
- Расположение файла подкачки:

10. Выведите сведения о памяти из диспетчера задач системы.

Ответьте на контрольные вопросы:

1. Определение виртуальная память.
2. Как реализована сегментная организация памяти?
3. Как реализована страничная организация памяти? Каким может быть размер одной страницы виртуальной памяти?
4. Для чего выполняется подкачка на диск?
5. Какие алгоритмы замещения страниц вам известны?
6. Как в операционной системе выполняется преобразование адресов данных?
7. Основное назначение кэш-памяти (СОЗУ) в компьютере?
8. Каков объем кэш-памяти первого уровня? второго уровня?

ЛИТЕРАТУРА (рекомендуемая)

1. Лекция управление памятью
2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил. — (Серия «Классика computer science»). ISBN 978-5-496-01395-6 — Текст : электронный // Образовательные ресурсы Интернета [сайт]. — URL: <https://www.ss-20.ru/>
3. <https://docs.microsoft.com/en-us/sysinternals/downloads/vmmap>
4. <https://learn.microsoft.com/ru-ru/windows/win32/cimwin32prov/win32-operatingsystem>
5. <https://learn.microsoft.com/ru-ru/visualstudio/profiling/memory-usage?view=vs-2022>

Практическое занятие №7

Работа с программой «Проводник». Работа с файловыми системами и дисками.

Цель работы: ознакомление с файловыми системами операционной системы windows, получение практических навыков работы с объектами файловой системы.

Теоретические сведения:

Файловая система (file system) — это способ организации, хранения и именования данных на носителях информации в компьютерах. Файловые системы используются также в другом электронном оборудовании: в цифровых фотоаппаратах и диктофонах, в мобильных телефонах, и т. п.

Файловые системы хранятся на дисках. Большинство дисков может быть разбито на один или несколько разделов, на каждом из которых будет независимая файловая система. Сектор 0 на диске называется главной загрузочной записью (Master Boot Record (MBR)) и используется для загрузки компьютера. В конце MBR содержится таблица разделов. Из этой таблицы берутся начальные и конечные адреса каждого раздела. Один из разделов в этой таблице помечается как активный. При загрузке компьютера BIOS (базовая система ввода-вывода) считывает и выполняет MBR. Первое, что делает программа MBR, — находит расположение активного раздела, считывает его первый блок, который называется загрузочным, и выполняет его. Программа в загрузочном блоке загружает операционную систему, содержащуюся в этом разделе. Для достижения единообразия каждый раздел начинается с загрузочного блока, даже если он не содержит загружаемой операционной системы. Кроме того, в будущем он может содержать какую-нибудь операционную систему.

Во всем остальном, кроме того, что раздел начинается с загрузочного блока, строение дискового раздела значительно различается от системы к системе. Наиболее часто используемое представление файловой системы представлены на рисунке.

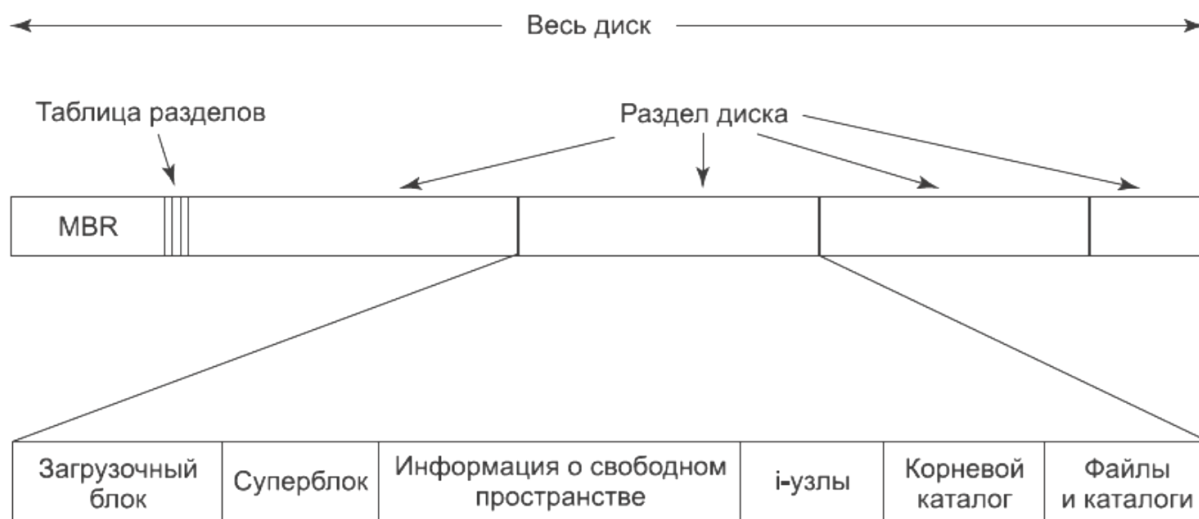


Рисунок 1 – Структура диска и дискового раздела

Первым элементом является суперблок. В нем содержатся все ключевые параметры файловой системы, которые считываются в память при загрузке компьютера или при первом обращении к файловой системе. Обычно в информацию суперблока включаются «магическое» число, позволяющее идентифицировать тип файловой системы, количество блоков в файловой системе, а также другая важная административная информация. Далее может находиться информация о свободных блоках файловой системы, к примеру в виде битового массива или списка указателей. За ней могут следовать i-узлы, массив структур данных — на каждый файл по одной структуре, в которой содержится вся информация о файле (свойства и атрибуты файла, а также дисковые адреса его блоков). Затем может размещаться корневой каталог, содержащий вершину дерева файловой системы. И наконец, оставшаяся часть диска содержит все остальные каталоги и файлы.

Файл — это именованная область на носителе внешней памяти. Файлы идентифицируются именами. Пользователи дают файлам символьные имена, при этом учитываются определенные ограничения ОС. Разные файлы могут иметь одинаковые символьные имена, в этом случае файл однозначно идентифицируется так называемым составным именем (путем), представляющим собой последовательность символьных имен каталогов.

Каталог — это группа файлов, объединенных пользователем исходя из некоторых соображений, содержащая системную информацию о их содержимом. В каталоге содержится список файлов, входящих в него, и устанавливается соответствие между файлами и их характеристиками (атрибутами).

Каталоги могут непосредственно содержать значения характеристик файлов или ссылаться на таблицы, содержащие эти характеристики. Каталоги могут образовывать иерархическую

структуру за счет того, что каталог более низкого уровня может входить в каталог более высокого уровня.

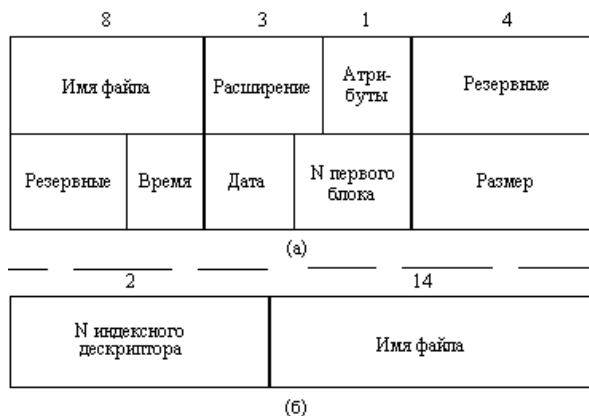


Рисунок 2 - Структура каталогов:
а - структура записи каталога windows (32 байта);
б - структура записи каталога linux

Иерархия каталогов может быть деревом или сетью. Каталоги образуют дерево, если файлу разрешено входить только в один каталог, и сеть - если файл может входить сразу в несколько каталогов. В windows каталоги образуют древовидную структуру, а в linux - сетевую. Как и любой другой файл, каталог имеет символьное имя и однозначно идентифицируется составным именем, содержащим цепочку символьных имен всех каталогов, через которые проходит путь от корня до данного каталога.

Общая модель файловой системы

Функционирование любой файловой системы можно представить многоуровневой моделью, в которой каждый уровень предоставляет некоторый интерфейс (набор функций) вышележащему уровню, а сам, в свою очередь, для выполнения своей работы использует интерфейс (обращается с набором запросов) нижележащего уровня.

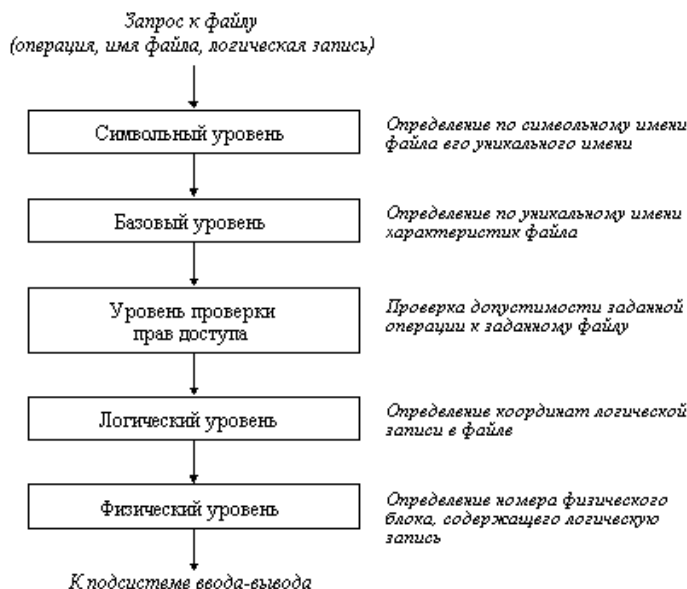


Рисунок 3 - Модель функционирования файловой системы

Задачей символьного уровня является определение по символьному имени файла его уникального имени. В файловых системах, в которых каждый файл может иметь только одно символьное имя (например, MS-DOS и MS Windows), этот уровень отсутствует, так как символьное имя, присвоенное файлу пользователем, является одновременно уникальным и может быть использовано операционной системой. В других файловых системах, в которых один и тот же файл может иметь несколько символьных имен, на данном уровне просматривается цепочка каталогов

для определения уникального имени файла. В файловой системе linux, например, уникальным именем является номер индексного дескриптора файла (i-node).

На следующем, базовом уровне по уникальному имени файла определяются его характеристики: права доступа, адрес, размер и другие.

Следующим этапом реализации запроса к файлу является проверка прав доступа к нему. Если запрашиваемый вид доступа разрешен, то выполнение запроса продолжается, если нет, то выдается сообщение о нарушении прав доступа.

На логическом уровне определяются координаты запрашиваемой логической записи в файле, то есть требуется определить, на каком расстоянии (в байтах) от начала файла находится требуемая логическая запись.

На физическом уровне файловая система определяет номер физического блока, который содержит требуемую логическую запись, и смещение логической записи в физическом блоке.

После определения номера физического блока, файловая система обращается к системе ввода-вывода для выполнения операции обмена с внешним устройством. В ответ на этот запрос в буфер файловой системы будет передан нужный блок, в котором на основании полученного при работе физического уровня смещения выбирается требуемая логическая запись.

Основными функциями файловой системы являются:

- размещение и упорядочивание на носителе данных в виде файлов;
- определение максимально поддерживаемого объема данных на носителе информации;
- создание, чтение и удаление файлов;
- назначение и изменение атрибутов файлов (размер, время создания и изменения, владелец и создатель файла, доступен только для чтения, скрытый файл, временный файл, архивный, исполняемый, максимальная длина имени файла и т.п.);
- определение структуры файла;
- поиск файлов;
- организация каталогов для логической организации файлов;
- защита файлов при системном сбое;
- защита файлов от несанкционированного доступа и изменения их содержимого.

Ход работы:

11. Изучите диск C, на своем компьютере и, используя фигуры MS Word или онлайн-ресурсы, составьте многоуровневую структуру вида:



Внесите информацию в отчет.

12. Изучите свойства и атрибуты файла и каталога в проводнике. Внесите информацию в отчет в виде таблицы:

Отличия свойств файлов и папок	Схожие свойства файлов и папок
...	...

13. Создайте папку Проводник в каталоге вашей учетной записи пользователя user. Скопируйте в эту папку все имеющиеся у вас файлы с отчетами по практическим работам по дисциплине, архив с программой process monitor и архив vmmap.

14. Отсортируйте файлы по возрастанию: по имени, по типу, по дате создания; по убыванию: по дате изменения, по размеру, по автору. Внесите информацию в отчет.
15. Выполните отображение содержимого папки проводник в ВИДе: огромных значков, крупных значков, средних значков, мелких значков, списка, таблицы, плитки, содержимого. Внесите информацию в отчет.
16. Выведите свойства локального диска с:. Внесите в отчет скриншоты всех вкладок свойств диска. Какие проверки можно выполнять для диска в ОС windows?

Ответьте на контрольные вопросы:

1. Что такое файловая система?
2. Какие файловые системы используются в Windows? Linux?
3. Что такое файл? Каталог? Диск?
4. Опишите структуру диска в виде тома(ов).
5. Опишите модель функционирования файловой системы.

ЛИТЕРАТУРА (рекомендуемая)

6. Лекция Файловые системы
7. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил. — (Серия «Классика computer science»). ISBN 978-5-496-01395-6— Текст : электронный // Образовательные ресурсы Интернета [сайт]. — URL: https://library-it.com/os/linux/sovremennyye-operacionnyye-sistemy_4e-izd-2015/
8. Рудаков, А. В. Операционные системы и среды : учебник / А.В. Рудаков. — Москва : КУРС : ИНФРА-М, 2029. — 304 с. — Глава 6.3 с.178-188

ПРАКТИЧЕСКАЯ РАБОТА №8

«Oracle VirtualBox. Установка операционной системы Windows. Настройка параметров»

Цель работы: *практическое изучение виртуальных машин на примере oracle VB, их создание, настройка, виртуализация в операционных системах, установка операционной системы Windows.*

1. Теоретические сведения

Технология виртуализации предназначена для осуществления возможности одновременного запуска на одном компьютере нескольких (в том числе, различных) ОС.

VirtualBox:

- является кросс платформенным приложением виртуализации и работает на компьютерах с процессорами Intel или AMD под управлением операционных систем Windows, Mac, Linux или Solaris;
- расширяет возможности вашего компьютера тем, что позволяет работать множеству операционных систем одновременно (внутри виртуальных машин);
- позволяет установить и запустить виртуальных машин столько, сколько вам нужно -- ваша фантазия ограничивается только размером свободного дискового пространства и памяти.

В ОС, реализованной на виртуальной машине, имеется возможность выполнять различные малоизученные или потенциально опасные для нее операции, не беспокоясь о последствиях: поскольку система является виртуальной, ее крах или частичное повреждение не скажется на работе реальной ОС.

Основные преимущества такого подхода состоят в следующем:

- появляется возможность инсталляции на одном компьютере нескольких ОС без необходимости соответствующего конфигурирования физических жестких дисков;
- возможно осуществлять работу с несколькими ОС одновременно с динамическим переключением между ними без перезагрузки реальной системы;
- сокращается время изменения состава и конфигурации установленных виртуальных ОС;
- осуществляется изоляция реального оборудования от нежелательного воздействия программного обеспечения, работающего в среде виртуальной ОС;
- появляется возможность моделирования вычислительной сети на единственном автономном компьютере.

Задачи, которые можно решать без перезагрузки системы и без опасения нанести ей какой-либо ущерб:

- освоение новой, альтернативной ОС;
- запуск специализированных приложений, предназначенных для работы в среде конкретной ОС;
- тестирование одного приложения под управлением различных ОС;
- установка и удаление оценочных или демонстрационных версий новых приложений;
- тестирование потенциально опасного программного обеспечения, относительно которого имеется подозрение на вирусное заражение;
- управление правами доступа пользователей к данным и программам в пределах виртуальной среды.

Основная терминология

Операционная система хоста (ОС хоста, host OS) – это операционная система физического компьютера, в которой установлен VirtualBox.

Гостевая операционная система (Гостевая ОС, guest OS) – это операционная система работающая внутри виртуальной машины.

Теоретически, в VirtualBox могут работать операционные системы DOS, Windows, OS/2, FreeBSD, OpenBSD.

Виртуальная машина (ВМ, VM) – это специальная среда которую VirtualBox создает для работы гостевой операционной системы.

Обычно, ВМ существует в виде окна на рабочем столе вашего компьютера, но в зависимости от используемого интерфейса она может занимать весь ваш экран или отображаться на экране другого компьютера.

VirtualBox воспринимает ВМ как множество параметров, которые определяют ее поведение. Эти параметры включают:

- аппаратные настройки (размер памяти выделенной для ВМ, типы жестких дисков, подключенные CD и т.д.);
- текущее состояние (ВМ работает, сохранена и т.п.).

Эти настройки представлены в окне менеджера VirtualBox, а также могут управляться через программу командной строки VBoxManage .

Гостевые дополнения (Guest Additions) – это пакет программных продуктов, поставляемых с VirtualBox и разработанных для установки внутри ВМ, способствующих улучшению производительности гостевой ОС и предоставления дополнительной функциональности.

ОСНОВНЫЕ ЭТАПЫ УСТАНОВКИ ОС:

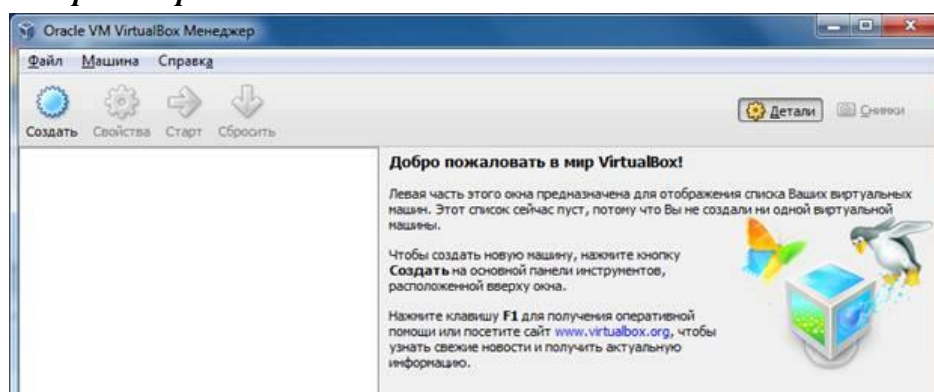
Независимо от того, какую операционную систему Вы решили установить на свой компьютер (Windows или Linux), предполагается выполнение следующих этапов:

1. Подготовка жесткого диска.
2. Загрузка с установочного диска.
3. Определение параметров установки.
4. Следование инструкциям мастера установки.
5. Установка драйверов.
6. Активация ОС.
7. Настройка ОС.
8. Установка дополнительного ПО.

2. Основная часть

Задание

1. Открыть приложение Oracle VM VirtualBox под Windows



- 2.
3. **Создать виртуальную машину со следующими параметрами:**
64-разрядную под windows10 с именем: для группы п/к-2Х-1 имя ХХ№, для группы пв/к-2Х-2 имя УУ№, (где № - номер студента четко по классному журналу)
1-2 ядра ЦП, 2048Мб ОП, виртуальный жесткий диск VDI 50Гб динамического типа.
4. **Просмотреть настройки виртуальной машины (см. Приложение А).**
5. **Выполнить установку операционной системы Windows 10 на виртуальную машину. Дистрибутивы образов установки находятся на C:/VM.**
6. **Создайте общую папку между хостом и гостевой виртуальной ОС ([теория](#)). Поместите в общую папку установочный объект системы программы оболочки Double Commander ([скачать](#), не забудьте о разрядности операционной системы)**
7. **Выполните установку программы оболочки Double Commander в вашу гостевую операционную систему.**
8. **Запустить cmd от имени Администратора и выполнить команды для работы с учетными записями пользователей:**
 - a) `net user`
 - b) `net user Администратор`
 - c) `net user fio1 /add`
 - d) `net user`
 - e) `net user fio /active:no`
 - f) `net user fio /delete`
 - g) `net user`
 - h) `for /l %i in (1,1,10) do net user user%i password%i /add`

¹ Ваше ФИО на английском, например sushkovaja

i) *net user*

j) *а теперь, тоже в цикле, удалите все созданных 10 учетных записей user_i.*

9. *Составьте электронный отчет и предъявите работу преподавателю.*

Контрольные вопросы:

1. Для чего применяют технологию виртуализации в операционных системах?
2. Какие задачи можно решать на виртуальных машинах?
3. В чем различие ОС хоста и гостевой ОС?
4. Опишите алгоритм создание/настройки виртуальной машины.
5. Можете ли вы получать доступ к сети Интернет из браузера ОС установленной на виртуальной машине и просматривать, например, свою почту?
6. Какие команды служат для управления учетными записями в CMD?
7. Какой формат имеет команда цикла в CMD?
8. Что будет выполнено в результате работы команд?

for %t in (computer phone) do echo My %t

for /l %x in (0,4,20) do echo %x

ПРАКТИЧЕСКАЯ РАБОТА №9

«Oracle VirtualBox. Установка операционной системы Linux. Настройка параметров»

Цель работы: *практическое изучение виртуальных машин на примере oracle VB, их создание, настройка, виртуализация в операционных системах, установка операционной системы Linux.*

ХОД РАБОТЫ:

Задание

1. Открыть приложение Oracle VM VirtualBox
2. Найти образ установки Linux на диске C:/VM.
3. Создать виртуальную машину (VM) заданной конфигурации и установить на VM ОС Linux Ubuntu.

Условия выполнения задания:

В папке класса создать папку VM, в которой с помощью дополнительных подкаталогов, вести ход работы. Для выполнения задания установку ПО и ОС производить только на локальный диск C: в вышеупомянутую директорию.
Выбор параметров осуществлять исходя из системных требований ОС Ubuntu.
Тип нового виртуального диска задать рационально, руководствуясь сохранностью свободного места раздела. Осуществить возможность использования физического привода.
При установке системы отказаться от обновлений. Язык установки русский.
Предусмотреть раздел для хранения личных данных (home)
Имя пользователя задать userX (где X – номер рабочего места). Имя компьютера CompX (где X – номер рабочего места). Задать пароль администратора: 55555. Вход в систему установить автоматически.

Оптимизировать скорость загрузки ОС.
Настроить поддержку интеграции мыши. Также система должна поддерживать общий буфер данных.

3. Настроить параметры и права доступа в ОС на VM

Условия выполнения задания:

Оптимизировать разрешение рабочего стола по сравнению с разрешением по умолчанию.
Задать комбинацию клавиш для переключения языка ввода Ctrl+Shift.
Создать на Рабочем столе ОС Ubuntu общую для обеих систем папку PublicX (где X – номер рабочего места) с помощью которой можно будет копировать файлы с хост-системы на виртуальную машину и обратно.
Поместить в эту папку текстовый файл info.txt, в котором отобразить полностью информацию о теме работы и фио автора.
Поместить в папку графический файл screen.jpg, содержащий скриншот всего Рабочего стола.
Активировать переключение рабочих столов Linux.
Установить двух панельный файловый менеджер для графического рабочего стола.
Вывести ярлык файлового менеджера на рабочий стол гостевой ОС.

Оформить электронный отчет по занятию.

Отчет должен содержать:

- титульный лист;
- цель работы;
- ход работы
- выводы по работе

Задание 3 выполнять с использованием общей папки между хостом и виртуальной ОС ([теория](#))

3. Заключительная часть.

Проверьте свои знания, ответив на контрольные вопросы:

12. Дайте определение операционная система, дистрибутив, драйвер.
13. Каков алгоритм установки ОС Linux?
14. Как настроить ПК для установки нескольких ОС на одном ПК?
15. Как создать новую виртуальную машину Oracle VirtualBox? Какие параметры виртуальной машины вы изменяли в работе?

Практическая работа №1

Сравнительный обзор современных ЭВМ.

Тема: История развития вычислительной техники. Классификации ЭВМ. Принципы организации ЭВМ

Учебная цель: изучение комплектации рабочей станции; изучение основных блоков персонального компьютера.

Учебные задачи:

1. Изучить классификацию ПК
2. Научиться подбирать комплектующие в соответствии с классификацией.

Студент должен

уметь:

- определять оптимальную конфигурацию оборудования и характеристики устройств для конкретных задач;
- идентифицировать основные узлы персонального компьютера, разъемы для подключения внешних устройств.

знать:

- построение цифровых вычислительных систем и их архитектурные особенности;
- принципы работы основных логических блоков системы;
- повышение производительности многопроцессорных и многоядерных систем;

Задачи практического занятия №1

1. Повторить краткие теоретические сведения по теме практического задания
2. Подобрать комплектующие ПК по вариантам.
3. Подготовить отчет по практической работе.

Обеспеченность занятия (средства обучения)

1. Учебно-методическая литература:
 - Методические указания по выполнению практических занятий;.
2. Справочная литература:
 - Трутнев Д. Р. Архитектуры информационных систем. Основы проектирования: Учебное пособие. - СПб.: НИУ ИТМО, 2012. - 66 с.;
3. Технические средства обучения:
 - Персональный компьютер
4. Программное обеспечение: ОС Windows, браузер, текстовый процессор MS Word

Краткие теоретические и учебно-методические материалы по теме практического занятия

Архитектура компьютера обычно определяется совокупностью ее свойств, существенных для пользователя. Основное внимание уделяется структуре и функциональным возможностям машины, которые можно разделить на основные и дополнительные.

Основные функции определяют назначение ЭВМ: обработка и хранение информации, обмен информацией с внешними объектами. Дополнительные функции повышают эффективность выполнения основных функций: обеспечивают эффективные режимы ее работы, диалог с пользователем, высокую надежность и др. названные функции ЭВМ реализуются с помощью ее компонентов: аппаратных и программных средств.

Персональный компьютер - это настольная или переносная ЭВМ, удовлетворяющая требованиям общедоступности и универсальности применения.

Достоинствами ПК являются:

- малая стоимость, находящаяся в пределах доступности для индивидуального покупателя;
- автономность эксплуатации без специальных требований к условиям окружающей среды;
- гибкость архитектуры, обеспечивающая ее адаптивность к разнообразным применениям в сфере управления, науки, образования, в быту;
- «дружественность» операционной системы и прочего программного обеспечения, обуславливающая возможность работы с ней пользователя без специальной профессиональной подготовки;
- высокая надежность работы (более 5 тыс. ч наработки на отказ).

Вопросы для закрепления теоретического материала к практическому занятию

1. Какие виды конфигураций вы знаете?
2. С чего начинается сборка компьютера?
3. Каким основным критериям нужно следовать для подбора комплектующих?

Варианты заданий для практического занятия №1

1. Подобрать ПК по следующей классификации: Офисный компьютер.
2. Подобрать ПК по следующей классификации: компьютер для видео-монтажа.
3. Подобрать ПК по следующей классификации: компьютер для аудио-монтажа.
4. Подобрать ПК по следующей классификации: компьютер для издательства.
5. Подобрать ПК по следующей классификации: компьютер для чертежных работ.
6. Подобрать ПК по следующей классификации: компьютер для работы с Flash-анимацией.
7. Подобрать ПК по следующей классификации: компьютер для разработчика игр и приложений.
8. Подобрать ПК по следующей классификации: компьютер с высокой производительностью.
9. Подобрать ПК по следующей классификации: компьютер для дизайнера.
10. Подобрать ПК по следующей классификации: компьютер для торгового представителя.
11. Подобрать ПК по следующей классификации: сервер для локальной сети до 20 рабочих станций.
12. Подобрать ПК по следующей классификации: сервер баз данных.
13. Подобрать ПК по следующей классификации: Офисный компьютер.
14. Подобрать ПК по следующей классификации: компьютер для видео-монтажа.
15. Подобрать ПК по следующей классификации: компьютер для аудио-монтажа.
16. Подобрать ПК по следующей классификации: компьютер для издательства.
17. Подобрать ПК по следующей классификации: компьютер для чертежных работ.
18. Подобрать ПК по следующей классификации: компьютер для работы с Flash-анимацией.
19. Подобрать ПК по следующей классификации: компьютер для разработчика игр и приложений.
20. Подобрать ПК по следующей классификации: компьютер с высокой производительностью.
21. Подобрать ПК по следующей классификации: компьютер для дизайнера.
22. Подобрать ПК по следующей классификации: компьютер для торгового представителя.
23. Подобрать ПК по следующей классификации: сервер для локальной сети до 20 рабочих станций.
24. Подобрать ПК по следующей классификации: сервер баз данных.

Инструкция по выполнению заданий практического занятия №1

1. Прочитать краткие теоретические сведения. При необходимости обратиться к справочным материалам.
2. Внимательно прочитать задание и приступить к выполнению.
3. Результат представить в виде таблицы:

Наименование комплектующих	Модель и характеристики	Цена
Итого		

Методика анализа результатов, полученных в ходе практического занятия

После выполнения практического задания проверить все комплектующие на сочетаемость между собой, проверить мощность элементов и блока питания.

Порядок выполнения отчета по практическому занятию

1. Обязательно указать цели и задачи практического занятия.
2. Выписать задание
3. Результат выполнения задания оформить в таблице.
4. Написать вывод о проделанной работе.

Практическая работа № 2. Анализ конфигурации вычислительной машины

Цели работы: формирование у обучающихся умений получать информацию о параметрах компьютерной системы и анализировать конфигурацию вычислительной машины, развивать умения и навыки самостоятельной работы с компьютерной системой.

Порядок выполнения работы

Практическая работа содержит отчет. Отчет должен содержать:

- название и цель работы;
- формулировки практических упражнений;
- наименование и номера таблиц, заполненные таблицы;
- вывод к практической работе;
- ответы на контрольные вопросы.

Практическое задание № 1.

Выполнить анализ конфигурации вычислительной машины.

Задание № 1

Заполнить таблицу 1 для конкретного рабочего компьютера (в таблицу следует заносить только реальные данные по конфигурации компьютера, в случае отсутствия какого-либо устройства ставится прочерк).

Таблица 1. Конфигурация рабочего ПК

№ п/п	Наименование параметра	Значение параметра
1	Тип и модель монитора	
2	Форм-фактор корпуса системного блока	

3	Клавиатура, интерфейс подключения	
4	Вид манипулятора "мыши", интерфейс ее подключения	
5	Интерфейсы подключения периферийных устройств на задней панели системного блока (наименование и количество)	
6	Интерфейсы подключения периферийных устройств на лицевой панели системного блока (наименование и количество)	
7	Процессор, модель и тактовая частота	
8	Объем оперативной памяти	
9	Тип модема и сетевого интерфейса	
10	Наименование и скорость привода для чтения оптических дисков	
11	Модель и объем памяти накопителя на жестких магнитных дисках	
12	Видеоадаптер, модель и объем видеопамати	
13	Модель звукового адаптера	
14	Версия операционной системы	
15	Другие периферийные устройства (принтер, сканер и т.д.)	

Задание № 2

Изучите основные сведения о данном компьютере. Для этого откройте соответствующее окно (**«Просмотр основных сведений о вашем компьютере»**) Создайте иллюстрацию: скопируйте содержимое экрана в буфер нажатием на клавиатуре клавиши PrintScreen. После этого вставьте содержимое буфера в документ MS Word, сохраните документ.

Задание № 3

Изучите периферийные устройства данного компьютера. Для этого откройте соответствующее окно. Создайте иллюстрацию: скопируйте содержимое экрана в буфер нажатием на клавиатуре клавиши PrintScreen. После этого вставьте содержимое буфера в документ MS Word, сохраните документ.

Практическое упражнение № 2

Составьте конфигурацию основных блоков ПК и периферийных устройств для вычислительных машин различной функциональной направленности:

1. Офисный ПК.
2. Домашний мультимедиа ПК.
3. Игровой ПК и мультимедиацентр.

Задание № 1

Таблица 2. Конфигурация ПК

№ п/п	Наименование параметра	Значение параметра
1	Тип и модель монитора	
2	Форм-фактор корпуса системного блока	
3	Клавиатура, интерфейс подключения	
4	Вид манипулятора "мыши", интерфейс ее подключения	
5	Интерфейсы подключения периферийных устройств на задней панели системного блока (наименование и количество)	
6	Интерфейсы подключения периферийных устройств на лицевой панели системного блока (наименование и количество)	
7	Процессор, модель и тактовая частота	
8	Объем оперативной памяти	
9	Тип модема и сетевого интерфейса	
10	Наименование и скорость привода для чтения оптических дисков	
11	Модель и объем памяти накопителя на жестких магнитных дисках	
12	Видеоадаптер, модель и объем видеопамати	
13	Модель звукового адаптера	
14	Версия операционной системы	
15	Периферийные устройства (принтер, сканер и т.д.)	

Заполните таблицу для каждой конфигурации.

Заполните Таблицу 2 для конфигурации Офисный ПК.

Заполните Таблицу 2 для конфигурации домашнего мультимедиа ПК.

Заполните Таблицу 2 для конфигурации игрового ПК и мультимедиа центра.

Задание № 2

1. Создайте отчет.
2. Обоснуйте ваш выбор конфигураций.
3. Сделайте **Выводы**

Контрольные вопросы

1. Что понимается под конфигурацией вычислительной машины?
2. Что включает базовая конфигурация компьютера?
3. В какой последовательности проводится анализ конфигурации вычислительной машины?
4. Что понимается под профилем оборудования? Каковы преимущества системы с настраиваемым профилем оборудования?
5. Какие инструменты операционной системы Windows используются для анализа конфигурации компьютера?

Практическая работа № 3

Классификация и типовая структура микропроцессоров.

Характеристики микропроцессоров

Цели работы: формирование у обучающихся умений получать и анализировать информацию о параметрах центрального процессора компьютерной системы, развивать умения и навыки самостоятельной работы с компьютерной системой.

Порядок выполнения работы

Практическая работа содержит отчет в виде презентации. Отчет должен содержать:

- титульную часть;
- название и цель работы;
- формулировки практических заданий;
- наименование и номера таблиц, заполненные таблицы;
- описание параметров, содержащихся в таблицах;
- выводы к практической работе;
- ответы на контрольные вопросы.

Задание № 1

1. Выполните анализ центрального процессора (ЦП) конфигурации вычислительной машины.
2. Изучите основные сведения о центральном процессоре вашего компьютера. Для этого откройте соответствующее окно (**«Просмотр основных сведений о вашем компьютере»**) Создайте иллюстрацию: скопируйте содержимое экрана в буфер нажатием на клавиатуре клавиши PrintScreen. После этого вставьте содержимое буфера в документ Microsoft Office PowerPoint, сохраните документ.
3. Для найденной информации выполните поиск информации основных характеристик ЦП.
4. Создайте слайд с таблицей по образцу таблицы №1. Заполните таблицу найденными параметрами. Таблица №1. Параметры центрального процессора.

№	Наименование параметра	Значение параметра ЦП
1.	Наименование	
2.	Ядро	
3.	Количество ядер	
4.	Техпроцесс, нм	
5.	Разъем	
6.	Частота, МГц	
7.	Множитель	
8.	FSB/HT/QPI	
9.	кэш L1, КБ	
10.	кэш L2, КБ	
11.	кэш L3, КБ	
12.	Напряжение питания, В	
13.	TDP, Вт	
14.	Кол-во транзисторов, млн	
15.	Площадь кристалла, кв. мм	
16.	Предельная температура, °C	
17.	Набор инструкций	
18.	Прочие особенности	

- В следующих слайдах презентации детально опишите каждый значимый параметр центрального процессора. Краткое описание параметра, его определение, назначение и характеристика.
- Сделайте выводы по работоспособности центрального процессора. Приведите наиболее точные отзывы и результаты тестирования.

Задание № 2

- Согласно варианту задания выполните [поиск в интернете](#) для указанного типа ЦП.
- Создайте слайд с таблицей по образцу таблицы №2. Заполните таблицу найденными параметрами.
- В следующих слайдах выполните сравнительный анализ характеристик двух

ЦП. Таблица 2. Сравнительный анализ характеристик центральных процессоров.

№	Наименование параметра	Значение параметра ЦП	Значение параметра ЦП(по варианту)
1.	Наименование		
2.	Ядро		
3.	Количество ядер		
4.	Техпроцесс, нм		
5.	Разъем		
6.	Частота, МГц		
7.	Множитель		
8.	FSB/HT/QPI		
9.	кэш L1, КБ		
10.	кэш L2, КБ		
11.	кэш L3, КБ		

12.	Напряжение питания, В		
13.	TDP, Вт		
14.	Кол-во транзисторов, млн		
15.	Площадь кристалла, кв. мм		
16.	Предельная температура, °C		
17.	Набор инструкций		
18.	Прочие особенности		

Задание № 3

1. Создайте отчет.
2. Сделайте **Выводы**
3. Ответьте на контрольные вопросы.
4. Загрузите презентацию на аккаунт платформы do.sevsu.ru

Контрольные вопросы

1. Процессор ЭВМ, его компоненты и их назначение.
2. Какой элемент процессора предназначен для указания адреса команды, которую следует выбирать из памяти для выполнения?
3. Назначение регистра — счетчика команд, почему он так называется?
4. Для чего в процессоре нужно устройство управления?
5. Для чего нужен регистр команд процессора?
6. Для чего нужен регистр состояния процессора?
7. Для чего нужен в процессоре регистр — указатель стека?
8. Регистры общего назначения; для чего они используются?
9. Что такое разрядность ЭВМ, параметры каких элементов ЭВМ она определяет?
10. Регистр адреса памяти, регистр данных памяти, где они расположены, их назначение.

Варианты заданий.

1	Intel Core i9-11900K
2	Intel Core i9-11900
3	Intel Core i9-10900
4	Intel Core i9-9900K
5	Intel Core i7-11700K
6	Intel Core i7-11700
7	Intel Core i7-10700K
8	Intel Core i7-10700
9	Intel Core i7-9700K
10	Intel Core i7-8700K
11	Intel Core i5-11600K
12	Intel Core i5-11600
13	Intel Core i5-11500T
14	Intel Core i5-11500
15	Intel Core i5-11400F
16	Intel Core i5-11400
17	Intel Core i5-10600
18	Intel Core i5-10500
19	Intel Core i5-10400
20	Intel Core i5-9600K

21	Intel Core i5-8600K
22	Intel Core i5-8400
23	Intel Core i3-10325
24	Intel Core i3-10320
25	Intel Core i3-10305
26	Intel Core i3-10300
27	Intel Core i3-10105
28	Intel Core i3-10100
29	Intel Core i3-8350K
30	Intel Core i3-8100

Практическая работа №1**Информационные ресурсы общества. Роль информационной деятельности в современном обществе**

ЦЕЛЬ: СОЗДАТЬ ОТЧЕТ С ОТВЕТАМИ НА ВОПРОСЫ В ВИДЕ ТЕКСТА И ТАБЛИЦ, ИСПОЛЬЗУЯ ИНТЕРНЕТ-РЕСУРСЫ.

I. В настоящее время существуют следующие виды информационных ресурсов:

- средства массовой информации;
- электронные библиотеки;
- электронные базы данных;
- сайты;
- информационные сервисы;
- информационный портал.

Сформулируйте определения данных ресурсов и запишите их в тетрадь.

II. Используя информационные ресурсы и интернет, заполните таблицу ответами на вопросы и указанием ресурса

ТАБЛИЦА РЕЗУЛЬТАТОВ ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ

№	Вопрос	Результат поиска	Инф ресурс
1	Длина экватора Земли в километрах		
2	Ускорение свободного падения тела		
3	5-ая планета Солнечной системы		
4	Количество букв в первой русской азбуке		
5	Годы жизни Пифагора		
6	У планеты Марс есть 2 спутника. В переводе с греческого – "страх" и "ужас". Назовите эти спутники		
7	Скорость обращения Луны вокруг Земли		
8	Три знаменитых романа Ивана Александровича Гончарова		
9	Ядро атома состоит из ...		
10	Закон Ома		
11	Автор повести "Левша"		
12	Бронза – это металл или сплав?		
13	Открытие первой в России		

№	Вопрос	Результат поиска	Инф ресурс
	железной дороги состоялось ...		
14	16-я буква русского алфавита		
15	Годы правления Екатерины Великой		
16	Оборона Севастополя продолжалась ... дней		
17	В чем измеряется солёность моря		
18	Температура кипения воды		
19	Самая высокая горная вершина в России (м)		
20	На берегу, какой реки располагается самый маленький город России		
21	Первый экземпляр манипулятора мышь		
22	Автор приема "лассо"		
23	Формулы вычисления корней квадратного уравнения		
24	Теорема Виетта		

III. С помощью одной из поисковых систем найдите информацию и занесите ее в таблицу:

№	Личности 20 века		
	Фамилия, имя	Годы жизни	Род занятий
1.	Джеф Раскин		
2.	Лев Ландау		
3.	Юрий Гагарин		
4.			
5.			

IV. Сделайте вывод о проделанной практической работе (не менее 50 слов).

Практическая работа №2

Вычисление информационного веса символов и объёма сообщений. Решение задач кодирования информации.

Цель работы: изучить способы представления текстовой, графической, звуковой информации и видеоинформации, научиться записывать информацию в различных кодировках.

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

Вся информация, которую обрабатывает компьютер, должна быть представлена двоичным кодом с помощью двух цифр 0 и 1. Эти два символа принято называть двоичными цифрами или

битами. С помощью двух цифр 0 и 1 можно закодировать любое сообщение. Это явилось причиной того, что в компьютере обязательно должно быть организовано два важных процесса: кодирование и декодирование.

Кодирование – преобразование входной информации в форму, воспринимаемую компьютером, то есть двоичный код.

Декодирование – преобразование данных из двоичного кода в форму, понятную человеку.

Дискретизация – это преобразование непрерывных изображений и звука в набор дискретных значений в форме кодов.

С точки зрения технической реализации использование двоичной системы счисления для кодирования информации оказалось намного более простым, чем применение других способов. Действительно, удобно кодировать информацию в виде последовательности нулей и единиц, если представить эти значения как два возможных устойчивых состояния электронного элемента:

0 – отсутствие электрического сигнала;

1 – наличие электрического сигнала.

Эти состояния легко различать. Недостаток двоичного кодирования – длинные коды. Но в технике легче иметь дело с большим количеством простых элементов, чем с небольшим числом сложных.

Способы кодирования и декодирования информации в компьютере, в первую очередь, зависят от вида информации, а именно, что должно кодироваться: числа, текст, графические изображения или звук.

Информация, в том числе графическая и звуковая, может быть представлена в аналоговой или дискретной форме. При аналоговом представлении физическая величина принимает бесконечное множество значений, причем ее значения изменяются непрерывно. При дискретном представлении физическая величина принимает конечное множество значений, причем ее величина изменяется скачкообразно.

Преобразование графической и звуковой информации из аналоговой формы в дискретную производится путем дискретизации, то есть разбиения непрерывного графического изображения и непрерывного (аналогового) звукового сигнала на отдельные элементы. В процессе дискретизации производится кодирование, то есть присвоение каждому элементу конкретного значения в форме кода.

Кодовые таблицы

Текстовая информация представляет собой набор символов, которыми могут быть буквы, цифры, знаки препинания. В компьютерной технике символы закодированы с помощью чисел, каждый символ кодируется своим набором цифр — **кодом**.

Существуют специальные таблицы кодировок, в которых хранятся такие наборы кодов символов.

Кодовая таблица — это представление символов в компьютерной технике на внутреннем уровне.

Самой первой системой кодирования текстовой информации была ASCII (американский стандартный код для обмена информацией). Однако этот стандарт рассчитан на передачу текстовой информации, которая состоит из букв английского алфавита. Таблица ASCII-кодов состоит из двух частей: стандартная (содержит коды от 0 до 127) и расширенная (содержит символы с кодами от 128 до 255).

Известно, что числа в ЭВМ представляются в двоичной форме, в виде набора нулей и единиц. Для этого разработаны специальные приемы перевода числовых значений в двоичную последовательность. Для представления буквы в компьютере ее заменяют числовым эквивалентом, а затем переводят в двоичный код. Каждой букве соответствует своя цифра. Все буквы с их числовыми эквивалентами сведены в кодовую таблицу символов, которая может называться ASCII, Unicode, КОИ-7, КОИ-8, Windows-1251.

Национальные версии таблицы ASCII

Таблица ASCII в интервале символов от 0 до 127 остается неизменной для любых программ. Диапазон кодовых значений от 128 до 255 может варьироваться в зависимости от языковых и национальных особенностей.

Для кодирования текстов на русском языке (т.е. букв кириллицы) наиболее широко применяются следующие кодовые страницы:

- Windows-1251, она же Microsoft code page 1251 (CP1251) — в системах Windows;
- Семейство кодовых страниц KOI-8;
- Альтернативная кодировка, она же IBM code page 866 — в системах DOS, а также в текстовых окнах Microsoft Windows;
- MacCyrillic — на компьютерах Macintosh.

Общим недостатком всех однобайтовых кодовых таблиц (в них для кодировки используются восьмиразрядные двоичные числа) является отсутствие в коде символа какой-либо информации, которая подсказывает компьютеру, какая в данном случае используется кодовая таблица.

Сообществом фирм Unicode предложена в качестве стандарта другая система кодировки символов. В этой системе для представления (кодирования) одного символа используются два байта (16 битов), и это позволяет включить в код символа информацию о том, какому языку принадлежит символ и как его нужно воспроизводить на экране монитора или на принтере. Два байта позволяют закодировать 65 536 символов. Объем информации, занимаемой одним и тем же текстом при этом, увеличивается вдвое. Зато тексты всегда будут «читаемыми» независимо от использованного национального языка и операционной системы.

2. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение понятиям кодирование, декодирование, дискретизация.
2. Чем отличается непрерывный (аналоговый) сигнал от дискретного(цифрового)?
3. Что такое кодовая таблица, код?
4. Как называется самая первая системой кодирования текстовой информации?
5. Перечислите однобайтовые кодовые таблицы для кодирования текстов на кириллице.
6. Какая кодовая таблица используется для описания символов двумя байтами?

3. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ

Задание 1.

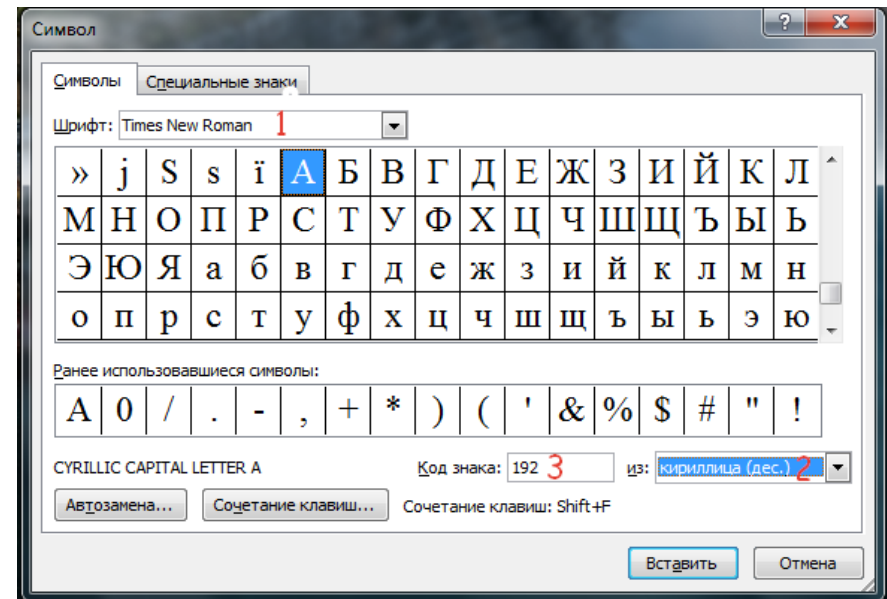
Используя таблицу символов, записать последовательностью десятичных числовых кодов в кодировке Windows свои данные:

- а) ФИО;
- б) название улицы, по которой проживаете;
- в) дату рождения.

Таблица символов отображается в редакторе MS Word с помощью команды вкладка Вставка>Символ>Другие символы. В поле Шрифт выбираете Times New Roman.⁽¹⁾

Набор символов, на котором основан код символа из: кириллица(дес.)⁽²⁾

Например, для буквы «А» (русской заглавной) код знака– 192.⁽³⁾



Пример:

И	В	А	Н	О	В		А	Р	Т	Е	М		П	Е	Т	Р	О	В	И	Ч
200	194	192	205	206	194		192	208	210	197	204		207	197	210	208	206	194	200	215

Задание 2.

Используя стандартную программу БЛОКНОТ, определить, какая фраза в кодировке Windows закодирована, закончить предложение по смыслу, продолжив код.

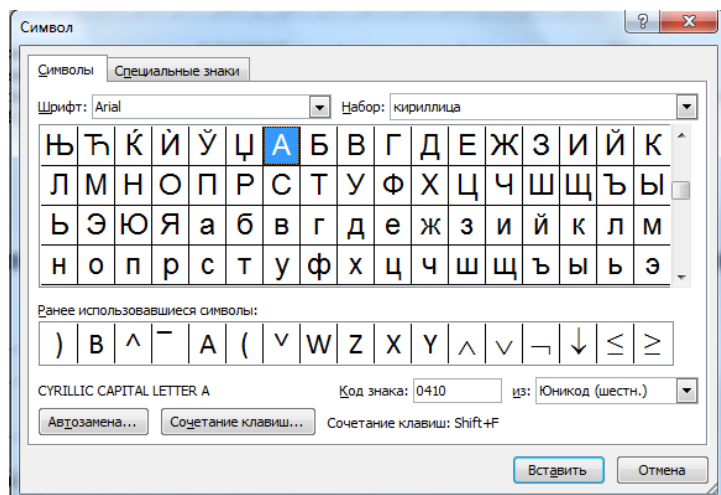
1. Запустить БЛОКНОТ.
2. С помощью дополнительной цифровой клавиатуры при нажатой клавише ALT ввести код, отпустить клавишу ALT.
3. В документе появиться соответствующий символ.
4. Фразу и коды оформить в виде таблицы.

0223		0243	0247	0243	0241	0252		0226		0209	0204	0202		0239	0238	
0241	0239	0229	0246	0232	0224	0235	0252	0237	0238	0241	0242	0232	058			
																.
																046

Задание 3.

Используя таблицу символов редактора MS Word в кодировке Юникод шрифтом Arial записать:

1. название колледжа
2. Фамилию Имя Отчество куратора группы
3. ваш № и модель телефона
4. Информацию и соответствующие ей коды оформить в виде таблицы.



Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

Практическая работа №1

Создание и нормализация таблиц базы данных

Цель работы : научиться определять структуру таблицы и устанавливать свойства ее полей в режиме конструктора таблиц; определять простой и составной первичный ключ; приводить таблицы нормальной форме; уметь устанавливать межтабличные связи.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предьявляется реализованная средствами Microsoft Access база данных для заданной предметной области

Краткие теоретические сведения:

Понятие нормализации схем баз данных неразрывно связано с понятием ограничений, накладываемых системой функциональных зависимостей, и во многом следует из этого понятия.

Исходной точкой любого проектирования базы данных является представление предметной области в виде одного или нескольких отношений, и на каждом шаге проектирования производится некоторый набор схем отношений, обладающих «улучшенными» свойствами. Таким образом, процесс проектирования представляет собой процесс нормализации схем отношений, причем каждая следующая нормальная форма обладает свойствами, в некотором смысле лучшими, чем предыдущая.

Каждой нормальной форме соответствует определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. Примером может служить ограничение первой нормальной формы – значения всех атрибутов отношения атомарны.

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц). Метод нормальных форм (НФ) состоит в сборе информации о объектах решения задачи в рамках одного отношения и последующей декомпозиции этого отношения на несколько взаимосвязанных отношений на основе процедур нормализации отношений.

Цель нормализации: исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении, редактировании и удалении кортежей (строк таблицы).

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- 1) первая нормальная форма (1 NF);
- 2) вторая нормальная форма (2 NF);
- 3) третья нормальная форма (3 NF);
- 4) нормальная форма Бойса – Кодда (BCNF);

- 5) четвертая нормальная форма (4 NF);
 6) пятая нормальная форма, или нормальная форма проекции-соединения (5 NF или PJ/NF). Основные свойства нормальных форм состоят в следующем:
 1) каждая следующая нормальная форма в некотором смысле лучше предыдущей нормальной формы;
 2) при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

Рассмотрим первые три нормальные формы

Первая нормальная форма

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

Вторая нормальная форма

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК).

Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

Третья нормальная форма

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Иными словами, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Порядок выполнения работы

Задание 1. Создайте базу данных о студентах, поступивших в ВУЗ, состоящую из двух таблиц:

«Группа» и «Студент». Дана исходная таблица 1. Привести ее к нормальной форме. Таблица 1.

Группа	Кол-во студенто в в группе	ФИО	Пол	Номер студен та	Год рождения	Адрес	Проходно й балл	Код специаль ности
11	25	Иванова	Ж	2	05.02.87		4,5	611
11	25	Сидоров	М	17	10.06.87		4,8	611
11	25	Петров	М	10	11.12.86		4,3	611
12	20	Мазуров	М	11	29.10.86		4,5	615
12	20	Бойко	Ж	1	06.01.87		5,0	615
12	20	Пашкова	Ж	9	15.12.86		4,3	615
13	20	Степанов	М	13	27.09.86		4,9	3514
13	20	Бородина	Ж	5	20.04.87		5,0	3514

Для создания структуры созданных таблиц необходимо определить основные характеристики полей таблиц, свойства полей, назначить первичный ключ. В таблицах 2 и 3 заполнить основные характеристики полей полученных таблиц. Заполненные таблицы 2 и 3 занести в отчет.

Таблица 2. Описание свойств полей таблицы «Группа»

Имя Поля	Ключ. поле	Тип данных	Размер, байт	Число десятич. знаков	Подпись поля	Условие на значение
						Сообщение об ошибке
		Текстовый	5			
		Текстовый	6			
		Числовой	Байт	0		
		Числовой	Одинарное с пл. точкой	2		

Таблица 3. Описание свойств полей таблицы «Студент»

Имя Поля	Ключ. поле	Тип данных	Размер, Байт	Число десятич. знаков	Подпись поля	Условие на значение
						Сообщение об ошибке
		Текстовый	5			
		Текстовый	2			
		Текстовый	15			
		Подстановкой (м, ж)	1			
		Дата/Время	Краткий форм. Д.			
		Текстовый	25			
		Числовой	Одинарное с пл. точкой	2		

Задание 2. Создание структуры базы данных

1. Запустите программу Microsoft Access.
2. В первом диалоговом окне Access установите переключатель «**Новая база данных**».
3. В следующем окне выберите папку и укажите в качестве имени файла новой базы данных

свою фамилию. Щелкните на кнопке **Создать**.

4. В окне новой базы данных из списка объектов выберите объект **Таблицы**. Создайте структуру таблицы Группа в режиме Конструктора. При задании **Условия на значение** для полей пользуйтесь Построителем выражений.

5. Задайте ключевое поле.
6. Сохраните структуру первой таблицы под именем **Группа**.
7. Создайте структуру таблицы Студент. Условие задавайте Построителем выражений.
8. Задайте составной уникальный ключ, состоящий из двух полей: **НГ** (номер группы) и **НС** (номер студента в группе). Для выделения полей пользуйтесь клавишей CTRL.

9. Сохраните таблицу под именем **Студент**. **Задание 3.**

Ввод данных в таблицы

10. В окне базы данных откройте таблицу **Группа** двойным щелчком на ее значке.
11. Введите записи.
12. Закройте таблицу **Группа**.
13. Откройте таблицу **Студент**.
14. Введите записи.
15. Закройте таблицу **Студент**.

Задание 4. Установка связей между таблицами

16. Выполните команду **Работа с базами данных-Схема данных**.
17. В окне Схемы данных расположите все таблицы.
18. Установите между таблицами **Группа**• **Студент** связь по простому ключу **НГ**.
19. В открывшемся окне обратите внимание, что тип отношения установится *один-ко-многим*. Установите все флажки, обеспечивающие целостность во взаимосвязанных таблицах.
20. Установите все флажки, обеспечивающие целостность во взаимосвязанных таблицах.
21. Сохраните схему данных и закройте окно.
22. Откройте таблицы **Группа** и **Студент**. Проверьте, как влияют установленные параметры поддержания связной целостности:
 - в таблице **Группа** измените номер группы (что произошло в подчиненной таблице **Студент**?

Ответ запишите в отчет);

- в подчиненную таблицу **Студент** попытайтесь ввести запись с номером группы, не представленным в главной таблице **Группа**; Что произошло? Ответ запишите в отчет.
- в главной таблице **Группа**, удалите запись, которая вводилась последней (выделите ее и выполните команду **Правка-Удалить**); что произошло в подчиненной таблице **Студент**? Ответ запишите в отчет.

23. Закройте таблицы **Группа** и **Студент**.

IV. Ввод и корректировка записей во взаимосвязанных таблицах

24. Откройте таблицу **Группа**.
25. Создайте заново удаленную запись.
26. Используя значки «+» в строке записи, введите несколько записей в подчиненную таблицу **Студент**. (если «плюсы» не отображены, выполните команду **Вставка-Подтаблица**).
27. Покажите работу преподавателю. Закройте окно Базы данных и Access.

Контрольные вопросы

1. Что такое нормализация базы данных?
2. Какие нормальные формы вы знаете? Дайте им определения.
3. Что такое декомпозиция?

Практическая работа №2

Проектирование структуры базы данных.

Цель работы: Изучить этапы проектирования реляционной БД; изучить и практически освоить процесс создания новой БД средствами СУБД MS Access, включая разработку макета таблиц в режиме Конструктора и построение схемы БД.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предъявляется реализованная средствами Microsoft Access база данных для заданной предметной области

Краткие теоретические сведения:

Прежде чем приступить к созданию таких объектов базы данных, как таблицы, формы и отчеты, нужно разработать их проект. Главное назначение проекта—выработка четкого пути, по которому нужно следовать при его реализации. База данных—достаточно сложный объект, и время, затраченное на ее планирование, может значительно сократить сроки ее разработки. Отсутствие продуманной структуры базы данных приводит к необходимости постоянной переделки и перенастраиванию объектов базы данных, таких, как формы и таблицы.

Проектирование базы данных целесообразно начать с краткого описания отчетов, списков и других документов, которые необходимо получить с помощью БД. Далее следует разработать эскиз объектов, требуемых для получения необходимых результатов и определит связи между этими объектами.

При разработке эскиза необходимо ответить на следующие вопросы:

Какими данными мы располагаем?

Какие данные будут содержать таблицы?

Какой тип и какие свойства должны иметь данные в каждом поле таблицы?

Как эти таблицы будут связаны друг с другом?

Законченный план должен содержать подробное описание всех таблиц (имена полей, типы данных и их свойства), а также связей между ними.

Проектирование предусматривает этапы создания проекта базы данных от концепции до реального воплощения. Этапы проектирования базы данных:

1. Исследование предметной области и формулировка основных допущений (накладываемых условий). На этом этапе составляется список всех форм и отчетов, которые могут быть затребованы пользователями вашей БД.

2. Анализ данных. Составить перечень всех элементов данных, входящих в формы и отчеты и сгруппировать их в таблицы БД.

3. Установить, какие взаимосвязи существуют между элементами данных. Определить первичные и вторичные (внешние) ключи отношений. Организовать поля данных в таблицах, причем это необходимо сделать, следуя 4-м правилам нормализации:

Правило 1: Каждое поле таблицы должно представлять уникальный тип информации. Это правило означает, что необходимо избавиться от повторяющихся полей и разделить составные поля на отдельные элементы данных.

Правило 2: Каждая таблица должна иметь уникальный идентификатор или первичный ключ, который может состоять из одного или нескольких полей.

Правило 3: В таблице не должно быть данных, не относящихся к объекту, определяемому первичным ключом.

Правило 4: Независимость полей. Это правило означает возможность изменять значения любого поля (не входящего в первичный ключ) без воздействия на данные других полей.

Результатом 3 этапа должна явиться группа таблиц, удовлетворяющих правилам нормализации. На этом же этапе необходимо установить связи между таблицами.

Порядок выполнения работы

В качестве предметной области выбрана деятельность фирмы по продаже автомобилей марки Toyota. Поставлена задача: упорядочить информацию о клиентах фирмы, ассортименте продукции и сформированных заказах.

Этапы 1-3 проектирования БД изучить теоретически, 4-5 выполнить практически.

Оформить отчет о выполнении всех этапов задания

Ответить на контрольные вопросы

1-й этап. Определение цели проектирования БД.

Проектируемая реляционная БД должна содержать структурированную информацию о клиентах, продаваемых автомобилях и заказах.

Проектируемая БД должна поддерживать выполнение, как минимум, следующих основных функций: ввод и обновление информации, просмотр и удаление.

2-й этап. Разработка информационно-логической модели предметной области.

Вся информация о предметной области может быть логично разделена на 3 таблицы:

Клиенты, Автомобили, Заказы.

При этом выполняются основные требования к содержанию таблиц:

1. Каждая таблица содержит информацию только на одну тему.
2. Информация в таблицах не дублируется.
3. Для связи между таблицами заданы **первичные ключи**, однозначно определяющие каждую запись в таблице.

Содержание базовых таблиц приведено ниже:

Таблица Клиенты	Таблица Автомобили	Таблица Заказы
1. Код клиента (ключ)	1. Код модели (ключ)	1. Код заказа (ключ)
2. Фамилия	2. Модель	2. Код клиента
3. Имя	3. Мощность двигателя	3. Код Модели
4. Отчество	4. Цвет	4. Дата заказа
5. Адрес	5. Количество дверей	5. Скидка, %
6. Телефон	6. Заводская цена	6. Оплачено
	7. Издержки (транспортные, предпродажные)	
	8. Специальная модель	
	9. Дополнительное оснащение	

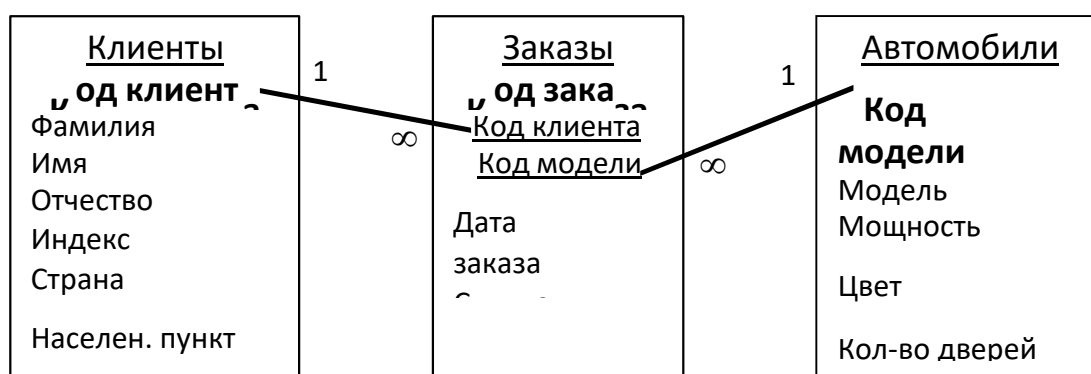
При разработке полей для каждой таблицы необходимо учитывать:

- Каждое поле должно быть связано с темой таблицы.
- Не включать в таблицу данные, которые являются результатом вычисления.
- Информацию следует разбивать на наименьшие логические единицы (Например, поля «Индекс», «Страна», «Населенный пункт», «Почтовый адрес», а не общее поле «Адрес»).

3-й этап. Определение отношений между таблицами.

Поскольку для проектируемой БД выполнены требования нормализации, между таблицами **Клиенты-Заказы** и **Автомобили-Заказы** могут быть установлены **одно-многочленные отношения** ($1 : \infty$), которые поддерживаются реляционной СУБД.

Связь между таблицами устанавливается с помощью **ключей** Код клиента и Код модели, которые в главных таблицах Клиенты и Автомобили являются первичными, а в таблице-связке Заказы - внешними.



4-й этап. Создание таблиц БД средствами СУБД MS Access.

4.1. Загрузить СУБД MS Access. Создать Новую базу данных.

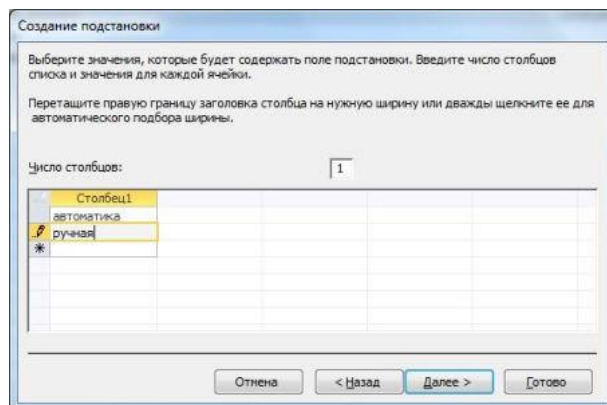
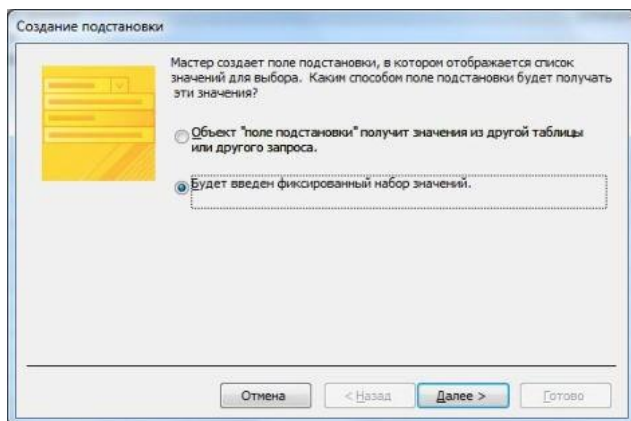
4.2. Создать макет таблицы Автомобили в режиме Конструктора, используя нижеприведенные данные об именах полей, их свойствах и типах данных.

*Все поля, за исключением поля Другое оснащение, должны быть обязательными для заполнения (Свойство Обязательное поле: Да).

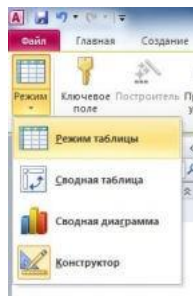
Имя поля	Тип данных	Описание	Свойства поля
Код модели	Числовой	Ключевое поле, код модели по заводскому каталогу	Индексированное поле: Да/Совпадения не допускаются Ключевое поле задается в меню Правка/Ключевое поле
Модель	Текст	Тип кузова	Размер поля: 20, Значение по умолчанию: Corolla Индексированное поле: Да/Совпадения допускаются (одна и та же модель может встречаться в БД многократно с различными вариантами оснащения)
Мощность	Текст	Мощность двигателя (кВт/л.с.)	Размер поля: 10 Индексированное поле: Нет
Цвет	Текст	Цвет кузова	Размер поля: 20 Индексированное поле: Нет
Количество дверей	Числовой	2 или 4	Размер поля: Байт Формат: Основной Число десятичных знаков: 0 Значение по умолчанию: 4 Условие на значение: 2 Or 4 Сообщение об ошибке: Данное поле может содержать

Имя поля	Тип данных	Описание	Свойства поля
			только 2 или 4 Индексированное поле: Нет
Коробка передач	*Мастер подстановок, фиксированный набор значений	Автоматика или ручная	Размер поля: 12 Значение по умолчанию: Ручная Условие на значение: “Ручная” От “Автоматика” Сообщение об ошибке: Допустимы только значения “Ручная” или “Автоматика” Индексированное поле: Нет
Обивка	*Мастер подстановок, фиксированный набор значений	Велюр, кожа, ткань	Размер поля: 10 Индексированное поле: Нет
Другое оснащение	Мето	Дополнительные аксессуары	Значение по умолчанию: Радио/плеер, раздвижная крыша
Заводская цена	Денежный	Заводская продажная нетто-цена	Формат: Денежный Число десятичных знаков: Auto Индексированное поле: Нет
Транспортные издержки	Денежный	Издержки на доставку	Формат: Денежный Число десятичных знаков: Auto Индексированное поле: Нет
Предпродажные издержки	Денежный	Издержки на предпродажную подготовку	Формат: Денежный Число десятичных знаков: Auto Значение по умолчанию: 105 Индексированное поле: Нет
Специальная модель	Логический	Спец. модель или стандартная	Формат: Да/Нет

*Мастер подстановок позволяет создать для указанных полей список, из которого выбирается значение.



4.3. Перейти в режим Таблицы, сохранив созданный макет таблицы под именем Автомобиля



4.4. Добавить в таблицу Автомобили 3 записи:

Код модели	12580	12653	12651
Модель	Corolla Liftback	Corolla CompactGT	Corolla CompactXL
Мощность	69/90	100/139	90/135
Цвет	Бутылочное стекло	Черный	Небесно-голубой
Количество дверей	4	2	2
Коробка передач	Автоматика	Ручная	Ручная
Обивка	Ткань	Кожа	Велюр
Другое оснащение	Радио/плеер, раздвижная крыша, лаковое покрытие “Металлик”	Радио/плеер, раздвижная крыша, алюмин. дворники	Электро-подъемник окон, раздвижная крыша
Заводская цена	39200	41100	37900
Транспортные издержки	1200	975	1050
Предпродажные издержки	105	105	105
Специальная модель	Нет	Да	Да

4.5. Создать макет таблицы Клиенты в режиме Конструктора.

*Обязательные поля Код клиента, Фамилия, Страна.

Имя поля	Тип данных	Описание	Свойства поля (определяют правила сохранения, отображения и обработки данных в поле)
Код клиента	Счетчик	Ключевое поле, уникальный номер клиента в БД	Индексированное поле: Да/Совпадения не допускаются <i>Ключевое поле задается в меню Правка/Ключевое поле</i>
Фамилия	Текст	Фамилия	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
Имя	Текст	Имя	Размер поля: 20, Индексированное поле: Да/Совпадения допускаются
Отчество	Текст	Отчество	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
Индекс	Числовой	Почтовый индекс	Размер поля: Длинное целое, Индексированное поле: Да/Совпадения допускаются
Страна	Текст	Название страны	Размер поля: 20, Индексированное поле: Да/Совпадения допускаются
Населенный пункт	Текст	Название населенного пункта	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются

Почтовый адрес	Текст	Почтовый адрес	Размер поля: 50, Индексированное поле: Нет
Телефон	Текст	Контактный телефон	Размер поля: 20, Индексированное поле: Нет

4.6. Добавить в таблицу Клиенты 3 записи. (Перейти в режим Таблицы, сохранив макет таблицы под именем Клиенты)

4.7. Создать в режиме Конструктора макет таблицы Заказы.

*Все поля, за исключением поля Скидка, являются обязательными для заполнения.

Имя поля	Тип данных	Описание	Свойства поля (определяют правила сохранения, отображения и обработки данных в поле)
Код заказа	Счетчик	Ключевое поле, уникальный номер заказа	Индексированное поле: Да/Совпадения не допускаются <i>Ключевое поле задается в меню Правка/Ключевое поле</i>
Код модели	Числовой, *Мастер подстановок	Внешний ключ, для связи с таблицей Автомобиля	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
Код клиента	Числовой, *Мастер подстановок	Внешний ключ, для связи с таблицей Клиенты	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
Дата заказа	Дата/время	Дата формирования заказа ДД.ММ.ГГ	Формат: Краткий формат даты Индексированное поле: Да/Совпадения допускаются
Скидка	Числовой	Размер скидки в %	Размер поля: Одинарное с плавающей точкой Формат: Процентный Условие на значение: Between 0 And 1

* Используя Мастер подстановок, сформировать для полей Код клиента и Код модели список выбора из таблиц Клиенты и Автомобиля (сбросить флажок Скрыть ключевое поле).

4.8. Добавить 5 записей в таблицу Заказы.

5-й этап. Создание схемы данных БД (связей между таблицами).

5.1. Выполнить опцию Схема данных из вкладки Работа с базами данных. В диалогом окне Добавление таблицы последовательно добавить все три таблицы. Закрывать диалоговое окно.

5.2. Установить связь между таблицами Клиенты-Заказы, Автомобиля-Заказы: выделить ключевое поле в главной таблице (Клиенты или Автомобиля) и перетащить его на соответствующее поле таблицы-связки Заказы. Обеспечить целостность данных.

5.3. Сохранить макет схемы данных.

Контрольные вопросы

1. Перечислите основные этапы проектирования реляционной БД. Кратко поясните содержание каждого этапа.
2. Какие требования предъявляют к содержанию таблиц реляционной БД?
3. Приведите порядок создания макета таблицы в режиме Конструктора.
4. Для чего в каждой таблице задается первичный ключ? В чем различие между первичным и внешним ключом?

Практическая работа № 1

Тема: «Нормативно-правовые документы и стандарты в области защиты информации и информационной безопасности»

Цель работы: Получить практические навыки работы со стандартами в области защиты информации и информационной безопасности.

Ход работы:

1. Найти в сети Internet серию стандартов, регламентирующих информационные технологии. Номер серии записать в тетрадь.
2. Выписать в тетрадь 10 стандартов и их реквизиты (номер, название, дата введения в действие, кем издан)
3. В виде таблицы или схемы выписать из стандарта стадии и этапы создания АС.
4. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется в соответствии с ходом работы)

Практическая работа № 2

Тема: «Системы менеджмента качества»

Цель работы: Ознакомиться и изучить содержания международных стандартов в области ИТ: ISO/IEC 9126, ISO/IEC 14598 и ИСО/МЭК 9126-1
Получить практические навыки работы с данными стандартами.

Ход работы:

1. Найти в сети Internet серию стандартов, регламентирующих информационные технологии. Номер серии записать в тетрадь.
2. Найти определения следующих понятий;
 - качество;
 - надежность;

- совместимость;
- взаимозаменяемость;
- безопасность;
- соответствие;
- дефект.

Для сравнения дайте определения этим же понятиям, принятым в отечественной практике и по результатам работы заполните таблицу.

Пример таблицы

№ п/п	Понятие	Определения международных стандартов	Определения, принятые в отечественной практике

3. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется, в соответствии с ходом работы)

Практическая работа № 3

Тема: «Стандарты и спецификации в области информационной безопасности»

Цель работы: Изучить международные и национальные стандарты и спецификации в области ИБ — от "Оранжевой книги" до ISO 15408.

Получить навыки определения сильных и слабых стороны этих документов.

Ход работы:

1. Ознакомиться со стандартами и спецификациями в области информационной безопасности.
2. Разработать интерфейс пользователя «Оранжевая книга» как оценочный стандарт».
3. Ответить на контрольные вопросы.

Контрольные вопросы:

1. Уровень безопасности С, согласно "Оранжевой книге", характеризуется:
 - произвольным управлением доступом
 - принудительным управлением доступом
 - верифицируемой безопасностью
2. Согласно рекомендациям X.800, аутентификация может быть реализована на:
 - сетевом уровне
 - транспортном уровне
 - прикладном уровне
3. "Общие критерии" содержат следующие виды требований:
 - функциональные
 - доверия безопасности
 - экономической целесообразности
4. Уровень безопасности В, согласно "Оранжевой книге", характеризуется:
 - произвольным управлением доступом
 - принудительным управлением доступом
 - верифицируемой безопасностью
5. Согласно рекомендациям X.800, целостность с восстановлением может быть реализована на:
 - сетевом уровне
 - транспортном уровне
 - прикладном уровне
6. В число классов функциональных требований "Общих критериев" входят:
 - анонимность
 - приватность
 - связь
7. Уровень безопасности А, согласно "Оранжевой книге", характеризуется:
 - произвольным управлением доступом
 - принудительным управлением доступом
 - верифицируемой безопасностью

8. Согласно рекомендациям X.800, неотказуемость может быть реализована на:

- сетевом уровне
- транспортном уровне
- прикладном уровне

9. В число классов требований доверия безопасности "Общих критериев" входят:

- разработка
- оценка профиля защиты
- сертификация

Практическая работа № 4

Тема: «Основные виды технической и технологической документации».

Цель работы: Познакомиться с технической и технологической документацией и получить практические навыки составления маршрутной технологической карты.

Ход работы:

1. Записать в тетрадь определение технологической документации.
2. Найти в сети Internet и изучить виды технологической документации. Записать их в тетрадь с кратким описанием.
3. Составить маршрутную технологическую карту на процесс производства компьютера.
4. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется обучающимся, в соответствии с ходом работы)

Практическая работа № 5

Тема: «Изготовление и оформление технической документации.
Экспертиза ценности научно-технических документов».

Цель работы: Познакомиться с правилами изготовления и оформления технической документации.

Ход работы:

1. Найти в сети Internet и ознакомиться с правилами изготовления и оформления технической документации. Записать их в тетрадь с кратким описанием.
2. Выписать, как создаётся экспертная комиссия по работе с научно-техническими документами.
3. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется обучающимся, в соответствии с ходом работы)

Практическая работа №1.

Тема: Вычисление погрешностей результатов арифметических действий над приближенными числами.

Приближение числа. Погрешности приближённых значений чисел.

Пусть x_0 – некоторая величина, истинное значение которой неизвестно и равно x . Число x_0 , которое можно принять за значение величины x , мы и будем называть приближенным числом.

Определение: Абсолютной погрешностью e_x приближенного значения числа X называется модуль разности между точным числом X его приближенным значением x , т.е. $e_x = |X - x|$.

Определение: Число x называется приближённым значением точного числа X с точностью до Δx , если абсолютная погрешность приближённого значения a не превышает Δx , т.е. $|X - x| \leq \Delta x$ (1).

Определение: Число Δx называется границей абсолютной погрешности приближённого значения числа x . $\Delta x = x - x_0 \Rightarrow x = x_0 + \Delta x$

Число Δx на практике стараются подобрать как можно меньше и простое по записи.

Из неравенства (1) найдём границы, в которых заключено точное значение числа X :

$$x - \Delta x \leq X \leq x + \Delta x.$$

$НГ_x = x - \Delta x$ - нижняя граница приближения величины X . $ВГ_x = x + \Delta x$ -

верхняя граница приближения величины X .

Определение: Относительной погрешностью δx приближенного числа x числа X называется отношение абсолютной погрешности Δx этого приближения к числу x , т.е.

$$\delta x = \frac{\Delta x}{|x|}$$

Если первая значащая цифра в относительной погрешности δx меньше 5, то граница относительной погрешности определяется из неравенства $\delta x \leq \frac{1}{2} \cdot 10^{-n}$, где n - количество верных цифр.

Определение: Число x_0 называется приближенным по недостатку, если оно меньше истинного значения ($x_0 < x$), и по избытку, если оно больше истинного значения ($x_0 > x$).

Например, число 3,14 является приближенным значением числа π по недостатку, а

2,72 – приближенным значением числа e по избытку.

Пример: $\pi = 3,14$ $x = 3,1416$; $x_0 = 3,14$ $\Delta x = 0,0016$

$$\delta x = 0,00051$$

$$\delta x = 0,051 \%$$

Относительная погрешность выражается в %.

Действия над приближенными числами.

Погрешность результата может быть выражена через погрешности первоначальных данных при помощи следующих теорем:

Теорема 1. Абсолютная погрешность алгебраической суммы равна сумме алгебраической погрешности слагаемых.

Теорема 2. Относительная погрешность алгебраической суммы заключена между наибольшим и наименьшим с относительной погрешностью слагаемых.

Теорема 3. Относительная погрешность произведения или частного равна сумме или разности относительных погрешностей соответственно делимого и делителя.

Абсолютная погрешность:

$$A + B = A_1 + B_1 \quad \begin{matrix} 0 \\ 0 \end{matrix} \quad \begin{matrix} + \\ + \end{matrix} \Delta A + \Delta B \Rightarrow \text{СУММА}$$

$$\Delta(A + B) = \Delta A + \Delta B$$

$$A \cdot B = A_0 B_0 + A_0 \Delta B + B_0 \Delta A + \Delta A \Delta B \quad \Rightarrow \text{УМНОЖЕНИЕ}$$

$$\Delta(AB) = A_0 \Delta B + B_0 \Delta A$$

$$3) \frac{A}{B} = \frac{A_0 + \Delta A}{B_0 + \Delta B} \Rightarrow \text{ЧАСТНОЕ}$$

$$\frac{\Delta A}{B} = \frac{\Delta A}{B_0} - \frac{A_0 \Delta B}{B_0^2}$$

Относительная погрешность:

$$1. \delta(A+B) = \frac{\delta A + \delta B}{A_0 + B_0}$$

$$2. \delta AB = \frac{\delta A}{A} + \frac{\delta B}{B} = \delta A + \delta B$$

$$3. \delta \frac{A}{B} = \frac{\delta A}{A_0} - \frac{\delta B}{B_0} = \delta A - \delta B$$

Пример.

1) Сложить приближенные числа 117,55 и 24,71

$$A+B=A_0+B_0+\Delta A+\Delta B$$

$$A=117,55 \quad B=24,71$$

$$A_0=117 \quad \Delta A=0,55$$

$$B_0=24 \quad \Delta B=0,71$$

$$A+B=117+24+0,55+0,71=141+1,26=142,26$$

$$2) \quad A \cdot B = (A_0 + \Delta A)(B_0 + \Delta B) = A_0 B_0 + A_0 \Delta B + \Delta A B_0 + \Delta A \Delta B$$

$$4,26 \text{ и } 7,12$$

$$\Rightarrow 4 \cdot 7 + 4 \cdot 0,12 + 7 \cdot 0,26 + 0,26 \cdot 0,12 = 28 + 0,48 + 1,82 = 30,33 \quad 117,55; 24,71;$$

$$18,88; 2,40 \text{ и } 17,50 - \text{сложить}$$

$$A+B+C+D+E=A_0+B_0+C_0+D_0+E_0+\Delta A+\Delta B+\Delta C+\Delta D+\Delta E=117+24+18+2+17+0,55+0,71+0,88+0,4+0,5=141+20=17+3,04=181,04$$

$$4) -0,906-0,330=1,23$$

$$5) 1,84 \cdot 0,672 = 1 \cdot 0 + 1 \cdot (+0,672) + 0,84 \cdot 0 + 0,84 \cdot (+0,672) = 0 + (+0,672) + 0 + (+0,564) = +1,236$$

$$A \cdot B = 8,46 \cdot 5,3 = 8 \cdot 5 + 8 \cdot 0,3 + 0,46 \cdot 5 + 0,46 \cdot 0,3 = 40 + 2,4 + 2,3 + 0,138 = 44,8$$

$$6) 5,73; 4,104; 2,60; 1,894$$

$$A+B+C+D=A_0+B_0+C_0+D_0+\Delta A+\Delta B+\Delta C+\Delta D=5+4+2+1+0,73+0,104+0,6+0,894=12+1,598=14,33$$

$$7) 3,5 \cdot 51,2 + 8,25 \cdot 12,7$$

Пример.

$$1) \text{ Умножить } 3,5 \cdot 51,2 \Rightarrow A_0=3 \quad B_0=51; \Delta A=0,5 \quad \Delta B=0,2.$$

$$3 \cdot 51 + 3 \cdot 0,2 + 51 \cdot 0,5 + 0,5 \cdot 0,2 = 153 + 0,6 + 25,5 + 0,1 = 179,2$$

$$2) 8,25 \cdot 12,7 \Rightarrow A_0=8 \quad B_0=12; \Delta A=0,25 \quad \Delta B=0,7$$

$$8 \cdot 12 + 8 \cdot 0,7 + 12 \cdot 0,25 + 0,25 \cdot 0,7 = 96 + 5,6 + 3 + 0,175 = 104,77$$

$$A_0=179 \quad \Delta A=0,2 \quad B_0=104 \quad \Delta B=0,77$$

$$A+B=283,97$$

$$8) (19,55+1,87) \cdot 0,42$$

$$A+B=19+1+0,55+0,87=21,42$$

$$A \cdot B = 21 \cdot 0 + 21 \cdot 0,42 + 0,42 + 0,42 \cdot 0 = 8,82 + 0,17 = 8,9$$

Значащие цифры. С помощью абсолютной погрешности определяется так называемое верное значение цифры.

Опр. Значащая цифра приближенного значения числа x , находящейся в разряде, в котором выполняется условие: абсолютная погрешность Δx не превосходит половину единицы этого разряда, называется *верной*.

Опр. Значащие цифры разрядов, где данное условие не выполняется, называется сомнительными. Все значащие цифры, расположены слева от верной, также будут верными и справа от сомнительной будут сомнительными.

Пример: 1) Для приближенного числа $x = 72,356$ известна абсолютная погрешность $\Delta x = 0,04$. Определить погрешность верно значащей цифры.

7	$\frac{10}{2} = 5 \geq 0,04$	верное число
2	$\frac{1}{2} = 0,5 \geq 0,04$	
3	$\frac{0,1}{2} = 0,05 \geq 0,04$	
5	$\frac{0,01}{2} = 0,005 <$	сомнительное число

$\Rightarrow x \approx 72,4$

2) $x = 3,73$; $\Delta x = 0,056$ $x \approx 3,7$

3	$\frac{1}{2} = 0,5 \geq 0,056$ – верное
2	
7	$\frac{0,1}{2} = 0,05 \leq 0,056$ – неверное
2	

Задания.

Определить абсолютную погрешность и указать верные цифры:

- 1) $1,936 \pm 12,29$;
- 2) $3,73 \pm 0,056$;
- 3) $3,672 \pm 0,0008$;
- 4) $4,732 \pm 0,06$;
- 5) 561274 ± 500 ;

Практическая работа № 2.

Тема: Решение алгебраических и трансцендентных уравнений методом половинного деления и методом итераций.

Определение. Трансцендентная функция – аналитическая функция, не являющаяся алгебраической. Простейшими примерами трансцендентных функций служат показательная функция, тригонометрические функции, логарифмическая функция, степенная функция.

Общая постановка задачи. Найти корни уравнения $f(x)=0$, где $f(x)$ -это алгебраическая или трансцендентная функция. Решение проводится в следующей последовательности:

- 1) отделение (локализация) корня;
- 2) приближенное вычисления корня до заданной точности.

Отделение действительного корня уравнения $f(x)=0$ – это нахождения отрезка $[a,b]$, которому принадлежит этот корень.

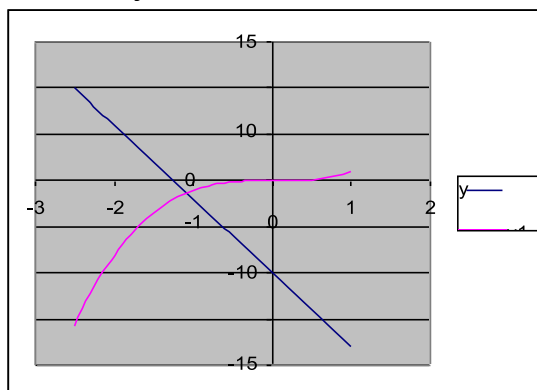
Определение. Отрезок $[a,b]$ называется отрезком локализации корня или изоляцией. Наиболее удобным методом определения отрезка является *графический* метод: строится функция $f(x)$, абсциссы точек пересечения графика функции с осью ОХ есть корни уравнения $f(x)=0$. Если функция $f(x)$ -сложная, то ее можно представить в следующем виде $f(x)=\varphi_1(x)-\varphi_2(x)$, так как $f(x)=0$, то $\varphi_1(x)=\varphi_2(x)$. Строим графики функций $y=\varphi_1(x)$ и $y=\varphi_2(x)$, абсциссы точек пересечения графиков этих функций будут корнями уравнения $f(x)=0$.

Пример. Найти корни уравнения:

$$y = x^3 + 8x + 10x^3 + 8x + 10 = 0$$

$$y_1 = x^3$$

$$y_2 = -8x - 10$$



Абсцисса точки пересечения графика функций $y_1 = x^3$ и $y_2 = -8x - 10$

$$x \in [-2; -1].$$

Уточнение корня производится следующими методами:

- 1) метод половинного деления (бисекций);
- 2) метод итераций;
- 3) метод хорд (секущих);
- 4) метод касательных (Ньютона);
- 5) комбинированный метод.

Метод половинного деления.

При решении уравнения методом половинного деления задается интервал $[a, b]$, на котором существует только одно решение и желаемая точность ε .

Такой метод можно применить, если функция $y=f(x)$ непрерывна на отрезке

$[a, b]$

и принимает разные значения на концах этого отрезка, то есть выполняется условие:

$$f(a) \cdot f(b) < 0$$

(*)

Разделим отрезок $[a, b]$ пополам точкой

$$C_1 = \frac{a+b}{2} \text{ и получим два отрезка } [a, c_1] \text{ и } [c_1, b].$$

Из этих двух отрезков выбираем тот, для которого выполняется условие (*). Пусть это отрезок $[c_1, b]$ и пусть для него выполняется условие:
 $f(c_1) \cdot f(b) < 0$

Снова делим отрезок $[c_1, b]$ пополам, находим

$$C_2 = \frac{C_1 + b}{2}. \text{ И продолжаем до тех}$$

пор, пока не достигнется заданная точность ε , то есть пока не выполнится условие $|c_{i+1} - c_i| < \varepsilon$.

Пример: Найти корни уравнения $y = x^3 + 8x + 10$, отрезок изоляции $[-2; -1]$. Проверим условие (1):

$$f(-2) = (-2)^3 + 8(-2) + 10 = -8 - 16 + 10 = -14$$

$$f(-1) = (-1)^3 + 8(-1) + 10 = -1 - 8 + 10 = 1$$

$$f(-2) \cdot f(-1) = -14 \cdot 1 = -14 < 0$$

$$C_1 = \frac{-2 + (-1)}{2} = -1,5 \Rightarrow [-2; -1,5] \quad \text{и} \quad [-1,5; -1]$$

Проверим условие (*) для отрезка $[-1,5; -1]$

$$f(-1,5) = (-1,5)^3 + 8(-1,5) + 10 < 0 \quad f(-1,5) \cdot f(-1) < 0$$

$$[-1,5; -1]: \quad C_2 = \frac{-1,5 + (-1)}{2} = -1,25$$

$$\varepsilon = 0,001$$

$$|-1,25 + (-1,5)| = 0,25 > 0,001 \Rightarrow \text{условие не выполняется} \quad C_{10} = -1,0889$$

$$C_{11} = -1,0884$$

$$|-1,0884 + (-1,0889)| = 0,0005 < 0,001 \Rightarrow \text{заданная точность достигнута и корень уравнения}$$

$$x = -1,0884 \approx -1,088$$

Метод итераций.

При решении нелинейных уравнений методом итераций запишем уравнение в виде $x = f(x)$. Задаются начальное значение аргумента x_0 и точность ε . Первое приближение решения x_1 , находим из выражения $x_1 = f(x_0)$; второе $x_2 = f(x_1)$ и т.д. В общем случае $(i+1)$ приближения найдем по формуле: $x_{i+1} = f(x_i)$. Указанную процедуру повторяем пока $|f(x_i)| > \varepsilon$. Условие сходимости $|f(x)| < 1$.

Задания

Решить уравнения методом половинного деления и методом итераций:

1. $x^3 + 2x - 7 = 0$; $x \in [1; 2]$; $\varepsilon = 0,01$

2. $x^3 + 8x + 10 = 0$; $x \in [-2; -1]$; $\varepsilon = 0,001$

3. $x^3 - 12x + 7 = 0$; $x \in [0; 1]$; $\varepsilon = 0,01$

4. $x^3 - 2x - 5 = 0$; $x \in [2; 3]$; $\varepsilon = 0,01$

5. $x^3 + 7x - 3 = 0$; $x \in [0; 1]$; $\varepsilon = 0,1$

6. $x^3 - x + 8 = 0$; $x \in [-3; -2]$; $\varepsilon = 0,1$

Практическая работа № 3.

Тема: Решение алгебраических и трансцендентных уравнений методами хорд и касательных.

Метод Хорд.

При решении уравнений методом Хорд, задаются интервал $[a; b]$, на котором существует только одно решение и точность, затем через две точки с координатами $(a, F(a))$ и $(b, F(b))$ проводим отрезок по прямой линии с осью ОХ (точка с).

Если при этом $F(a) \cdot F(c) < 0$, то правую границу интервала переносим в точку с ($b=c$). Если указанное условие не выполняется, то в точку с переносится левая граница интервала ($a=c$). Поиск решения прекращается при достижении заданной точности $|F(c)| < \varepsilon$.

Этот метод применяется для решения уравнений вида $f(x)=0$ с отрезком изоляции $[a; b]$, то есть $x \in [a; b]$ и выполняется условие:

1) $f(a) \cdot f(b) < 0$

2) $f'(x)$ сохраняет знак на отрезке $[a, b]$, то есть функция возрастает или убывает на этом отрезке.

Первое приближение корня находится по формуле:

$$x_1 = a - \frac{(b-a) \cdot f(a)}{f(b) - f(a)}$$

и из отрезка $[a, x_1]$ и $[x_1, b]$ выбираем тот, на котором выполняется условие (1). Второе приближение корня:

$$x_2 = a - \frac{(x_1 - a) \cdot f(a)}{f(x_1) - f(a)}, \text{ если } x \in [a; x_1] \quad \text{и по формуле}$$

$$x_2 = x_1 - \frac{(b - x_1) \cdot f(x_1)}{f(b) - f(x_1)}, \text{ если } x \in [x_1; b]$$

Так продолжается до тех пор, пока не достигнет заданной точности.

Метод касательных.

При решении уравнений методом касательных, задается начальным значением аргумента x_0 и точность ε . Затем в точке $(x_1, F(x_0))$ проводим касательную к графику $f(x)$, и определяем точку пересечения касательной с осью абсцисс x_1 .

В точке $(x_1, F(x))$ снова строим касательную; находим следовательно приближенную искомого решение и т.д. Указанную процедуру повторим пока $|F(x_i)| > \varepsilon$.

Метод касательных применяется для уравнений вида $f(x)=0$, если корень x отделенный, $x \in [a; b]$, и выполняется условия:

1) $f(a) \cdot f(b) < 0$;

2) f' и f'' сохраняет свои знаки на отрезке $[a, b]$, то есть функция либо возрастает, либо убывает на отрезке и сохраняет на нем направление выпуклости.

На отрезке $[a, b]$ выбирается такое число x_0 , для которого добивается условие

$$f'(x_0) \cdot f''(x_0) > 0$$

Таким образом, выбирается точка с абсциссой x_0 , которая касается графика функции $f(x)$, пересекает ось ОХ на отрезке $[a, b]$, выбираем один из отрезков $[a, b] \Rightarrow$ выбираем x_0

Первое приближения корня находится по формуле:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$
$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

вычисляется до тех пор, пока не выполнится условие $|x_n - x_{n-1}| < \varepsilon$.

Комбинированный метод хорд и касательных.

Этот метод применяется, если выполняется два условия:

- 1) $f(a) \cdot f(b) < 0$;
- 2) $f'(x)$ и $f''(x)$ сохраняет знак на отрезке $[a, b]$, то есть график функции возрастает, либо убывает на отрезке $[a, b]$ и сохраняет направление выпуклости.

Схема нахождения корня уравнения по комбинированному методу:

1. Найти знак $f(a)$ и $f(b)$;
2. Проверить выполнения условия (1);
3. Найти производные $f'(x)$ и $f''(x)$;
4. Проверить постоянство знака производной на отрезке $[a; b]$;
5. Найти приближения корня для метода касательных. За x_0 выбираем

тот из концов отрезка $[a; b]$, для которого выполняется условие $f(x_0) \cdot f''(x_0) > 0$;

6. а) по методу касательных $x_{11} = x_0 - \frac{f(x_0)}{f'(x_0)}$ - первое приближение корня;

$$x_{11} = x_0 - \frac{f(x_0)}{f'(x_0)}$$

б) по методу хорд
$$x_{12} = a - \frac{(b-a) \cdot f(a)}{f(b) - f(a)}$$

7. $\xi = \frac{x_{11} + x_{12}}{2}$; _____

8. Проверим, выполняется ли условие $|\xi - x_{11}| < \varepsilon$, где ε - заданная точность. Если условие не выполняется, то продолжаем применять метод по схеме 1-8.

Задания

Решить уравнения методом хорд, методом касательных и комбинированным методом:

1. $x^3 + 2x - 7 = 0$; $x \in [1; 2]$; $\varepsilon = 0,01$
2. $x^3 + 8x + 10 = 0$; $x \in [-2; -1]$; $\varepsilon = 0,001$
3. $x^3 - 12x + 7 = 0$; $x \in [0; 1]$; $\varepsilon = 0,01$
4. $x^3 - 2x - 5 = 0$; $x \in [2; 3]$; $\varepsilon = 0,01$
5. $x^3 + 7x - 3 = 0$; $x \in [0; 1]$; $\varepsilon = 0,1$
6. $x^3 - x + 8 = 0$; $x \in [-3; -2]$; $\varepsilon = 0,1$

Практическая работа № 4.

Тема: Решение систем линейных уравнений приближенными методами. Метод простой

итерации

Как отмечалось ранее, итерационные методы используются для решения уравнений и систем любой природы. Рассмотрим, как это делается применительно к системам линейных алгебраических уравнений.

Приведём систему линейных алгебраических уравнений (2.7)

[illegible]

к равносильной ей системе вида $x = Ax$:

$$\begin{cases} x_1 = \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n + \beta_1, \\ x_2 = \alpha_{21}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n + \beta_2, \\ \\ x_n = \alpha_{n1}x_1 + \alpha_{n2}x_2 + ... + \alpha_{nn}x_n + \beta_n \end{cases} \quad (2.10)$$

В сокращенной форме:

$$x_i = \sum_{j=1}^n \alpha_{ij} x_j + \beta_i, \quad i = \overline{1, n}$$

О системе (2.10) говорят, что она «приведена к нормальному виду».

$$F: y_i = \sum_{j=1}^n \alpha_{ij} x_j + \beta_i \quad (2.11),$$

Правая часть системы определяет отображение

переводящее точку $\bar{x}(x_1, x_2, \dots, x_n)$ в точку $\bar{y}(y_1, y_2, \dots, y_n)$. Используя отображения (2.11) и

выбрав начальную точку $\bar{x}^0(x_1^0, x_2^0, \dots, x_n^0)$ можно построить итерационную последовательность точек.

Если отображение F является сжимающим, то эта последовательность сходится и её предел является решением системы (2.10) а, следовательно, и решением исходной системы (2.7).

Замечание: Отображение является *сжимающим*, если расстояние между образами меньше, чем расстояние между исходными точками.

Для отображения (2.11) необходимым и достаточным условием сжимаемости

$$\max_{1 \leq j \leq n} \sum_{i=1}^n |\alpha_{ij}| < 1$$

является следующее: $\sum_{i=1}^{1 \leq j \leq n} |a_{ji}|$, (2.12), т.е. максимальная из сумм модулей коэффициентов при неизвестных в правой части системы (2.10), взятых по столбцам, должен быть меньше 1.

Практическая схема решения с.л.у. методом простой итерации

С.л.у. (2.7) необходимо привести к нормальному виду (2.10).

Для обеспечения сходимости итерационной последовательности необходимо,

чтобы коэффициенты α_{ij} при неизвестных в правой части системы были существенно меньше 1.

Этого можно достичь, если исходную систему (2.7) с помощью равносильных преобразований привести к системе, у которой абсолютная величина коэффициентов, стоящих на главной диагонали, больше абсолютных величин каждого из других

коэффициентов, стоящих при неизвестных в соответствующих уровнях (такую систему называют *системой с преобладающими диагональными коэффициентами*). Если теперь разделить все уравнения на соответствующие диагональные коэффициенты и выразить из каждого уравнения неизвестное с коэффициентом, равным 1, будет получена система (2.10), у которой все $|\alpha_{ij}| < 1$.

Для проверки точности решения используем условие (2.12).

Пример: Решить систему линейных уравнений

$$\begin{cases} 2,34x_1 - 4,21x_2 - 11,61x_3 = 14,41, \\ 8,04x_1 + 5,22x_2 + 0,27x_3 = -6,44, \\ 3,92x_1 - 7,99x_2 + 8,37x_3 = 55,56. \end{cases}$$

методом простой итерации с точностью $\varepsilon = 1 \cdot 10^{-4}$.

Решение:

Построим систему с преобладающими диагональными коэффициентами.

В качестве 1-ого уравнения возьмем 2-ое, в качестве 3-его уравнения – 1-ое, в качестве 2-ого уравнения – сумму 1-го и 2-го уравнений:

$$\begin{cases} 8,04x_1 + 5,22x_2 + 0,27x_3 = -6,44, \\ 6,26x_1 - 12,2x_2 - 3,24x_3 = 69,97, \\ 2,34x_1 - 4,21x_2 - 11,61x_3 = 14,41. \end{cases}$$

Разделим каждое из полученных уравнений на диагональный коэффициент и выразим из каждого уравнения диагональные элементы:

$$\begin{cases} x_1 = -0,649x_2 - 0,034x_3 - 0,801, \\ x_2 = 0,573x_1 - 0,266x_3 - 5,735, \\ x_3 = 0,202x_1 - 0,363x_2 - 1,241. \end{cases}$$

Метод Зейделя

Будем снова рассматривать систему линейных уравнений (2.7) и эквивалентную ей систему (2.10).

При решении системы (2.10) методом простой итерации каждый шаг итерационного процесса состоит в переходе от уже имеющегося приближения значений неизвестных к новому (очередному) приближению.

Обозначим элементы имеющегося приближения через $x_1, x_2, \dots, x_n$, а элементы очередного (вычисляемого) приближения через $y_1, y_2, \dots, y_n$.

Вычислительные формулы имеют вид:

$$y_i = \sum_{j=1}^n \alpha_{ij} x_j + \beta_i, \quad i = 1, 2, \dots, n$$

Основная идея метода Зейделя состоит в том, что на каждом шаге итерационного процесса при вычислении значения y_i учитываются уже полученные значения $y_1, y_2, \dots, y_{i-1}$. Выпишем соответствующие вычислительные формулы:

$$y_1 = \sum_{j=1}^n \alpha_{1j} x_j + \beta_1,$$

$$y_2 = \alpha_{21} y_1 + \sum_{j=2}^n \alpha_{2j} x_j + \beta_2,$$

$$y_i = \sum_{j=1}^{i-1} \alpha_{ij} y_j + \sum_{j=i}^n \alpha_{ij} x_j + \beta_i$$

$$y_n = \sum_{j=1}^{n-1} \alpha_{nj} y_j + \alpha_{nn} x_n + \beta_n$$

Справедливо следующее **утверждение**:

Если для матрицы коэффициентов системы (2.10) выполняется условие (2.12), то итерационный процесс метода Зейделя сходится к решению системы при любом выборе начального приближения $x^{(0)}$.

Преимущество этого метода состоит в том, что он обеспечивает более быструю сходимость, чем метод простой итерации.

Пример: Решить систему линейных уравнений

$$10x_1 + x_2 + x_3 = 12$$

$$2x_1 + 10x_2 + x_3 = 13$$

$$2x_1 + 2x_2 + 10x_3 = 14$$

$$x_1 = 1,2 - 0,1x_2 - 0,1x_3 \quad x_2 = 1,3 - 0,2x_1 -$$

$$0,1x_3 \quad x_3 = 1,4 - 0,2x_1 - 0,2x_2$$

$$x^{(0)} = 1,2; x^{(0)}_2 = 0; x^{(0)}_3 = 0$$

$$x^{(1)}_1 = 1,2 - 0,1 \cdot 0 - 0,1 \cdot 0$$

$$\begin{cases} x^{(1)}_1 = 1,2 - 0,1 \cdot 0 - 0,1 \cdot 0 \\ x^{(1)}_2 = 1,3 - 0,2 \cdot 1,2 - 0,1 \cdot 0 \\ x^{(1)}_3 = 1,4 - 0,2 \cdot 1,2 - 0,2 \cdot 0 \end{cases}$$

$$x^{(1)}_1 = 1,2; x^{(1)}_2 = 1,06; x^{(1)}_3 = 1,16$$

$i = 2$ (i – приближение)

$$x^{(2)}_1 = 1,2 - 0,1 \cdot 1,06 - 0,1 \cdot 1,16$$

$$\begin{cases} x^{(2)}_1 = 1,2 - 0,1 \cdot 1,06 - 0,1 \cdot 1,16 \\ x^{(2)}_2 = 1,3 - 0,2 \cdot 1,2 - 0,1 \cdot 1,16 \\ x^{(2)}_3 = 1,4 - 0,2 \cdot 1,2 - 0,2 \cdot 1,06 \end{cases}$$

$$x^{(2)}_1 = 0,978; x^{(2)}_2 = 0,944; x^{(2)}_3 = 0,948$$

Задания

Методом Зейделя решить с точностью 0,001 систему линейных уравнений, приведя ее к виду, удобному для итераций.

$$\begin{cases} 1,7x_1 + 2,8x_2 + 1,9x_3 = 0,7; \\ 2,1x_1 + 3,4x_2 + 1,8x_3 = 1,1; \\ 4,2x_1 - 1,7x_2 + 1,3x_3 = 2,8. \end{cases}$$

1.

$$\begin{aligned} 2. \quad & \begin{cases} 2,7x_1 + 3,3x_2 + 1,3x_3 = 2,1; \\ 3,5x_1 - 1,7x_2 + 2,8x_3 = 1,7; \\ 4,1x_1 + 5,8x_2 - 1,7x_3 = 0,8. \end{cases} \end{aligned}$$

$$\begin{aligned} 3. \quad & \begin{cases} 3,1x_1 + 2,8x_2 + 1,9x_3 = 0,2; \\ 1,9x_1 + 3,1x_2 + 2,1x_3 = 2,1; \\ 7,5x_1 + 3,8x_2 + 4,8x_3 = 5,6. \end{cases} \end{aligned}$$

$$\begin{aligned} 4. \quad & \begin{cases} 9,1x_1 + 5,6x_2 + 7,8x_3 = 9,8; \\ 3,8x_1 + 5,1x_2 + 2,8x_3 = 6,7; \\ 4,1x_1 + 5,7x_2 + 1,2x_3 = 5,8. \end{cases} \end{aligned}$$

$$\begin{aligned} 5. \quad & \begin{cases} 3,3x_1 + 2,1x_2 + 2,8x_3 = 0,8; \\ 4,1x_1 + 3,7x_2 + 4,8x_3 = 5,7; \\ 2,7x_1 + 1,8x_2 + 1, x_3 = 3,2. \end{cases} \end{aligned}$$

$$\begin{aligned} 6. \quad & \begin{cases} 7,6x_1 + 5,8x_2 + 4,7x_3 = 10,1; \\ 3,8x_1 + 4,1x_2 + 2,7x_3 = 9,7; \\ 2,9x_1 + 2,1x_2 + 3,8x_3 = 7,8. \end{cases} \end{aligned}$$

Практическая работа № 5.

Тема: Составление интерполяционных формул Лагранжа, Ньютона.

Постановка задачи аппроксимации функций

В вычислительной математике нередки случаи, когда одну функцию приходится заменять другой, более другой и удобной для дальнейшей работы. Такую задачу называют **аппроксимацией** функций. Поводом для аппроксимации функции может послужить, в частности, табличный способ её задания. Предположим, что результате некоторого эксперимента для конечного набора значений величины x из отрезка $[a;b]$: $a = x_0 < x_1 < \dots < x_n = b$

получен набор значений y_i величины y

Таблица (3.1)

x	x_0	x_1	$\dots$	x_i	$\dots$	x_n
$F(x)$	y_0	y_1	$\dots$	y_i	$\dots$	y_n

Допустим, существует функциональная зависимость $y=F(x)$. Необходимо задать $F(x)$ аналитически.

Точки $x_0, \dots, x_n$ называют **узлами аппроксимации**. Классический подход к численному решению подобных задач заключается в том, чтобы, опираясь на информацию о функции F , по некоторому алгоритму подобрать аппроксимирующую функцию G , в определенном смысле «близкую» к F .

Чаще всего задача аппроксимации решается с помощью многочленов. Вычисления значений многочлена легко автоматизировать, производная и интеграл от многочлена, в свою очередь, также являются многочленами.

Для оценки «близости» функций выбирают тот или иной **критерий согласия**.

Для функций, заданных таблично, достаточно распространенным является **критерий Чебышева**, который определяет расстояние ρ между аппроксимируемой и аппроксимирующей функциями как максимум величины отклонения

$$\rho = \max_i |F(x_i) - G(x_i)| \quad (3.1)$$

между этими функциями в узлах:

Если $\rho = 0$, т.е. $F(x_i) = G(x_i) = y_i$ (в узлах значения совпадают), то соответствующий способ аппроксимации называют **интерполяцией**, а процедуру вычисления значений $F(x)$ с помощью $G(x)$ в точках, не являющихся узлами сетки, **-интерполированием**.

Часто процедура аппроксимации связана с другим критерием согласия:

$$\rho = \min_i \sum_{i=0}^n (F(x_i) - G(x_i))^2$$

Применяемый на его основе способ аппроксимации называется **методом наименьших квадратов**.

Существование и единственность интерполяционного многочлена

Пусть известны значения некоторой функции $F(x)$:

x	x_0	x_1	$\dots$	x_i	$\dots$	x_n
$F(x)$	y_0	y_1	$\dots$	y_i	$\dots$	y_n

Будем решать задачу интерполирования этой функции с помощью построения интерполяционного многочлена n -ой степени.

$G(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ (3.2), который в узлах принимает значения y_i

По формуле (3.9) получаем:

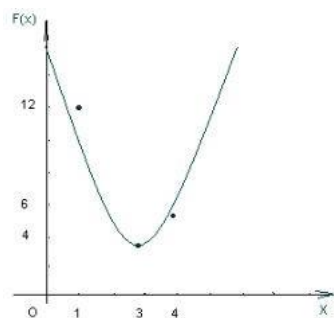
$$L_2(x) = 12 \frac{(x-3)(x-4)}{(1-3)(1-4)} + 4 \frac{(x-1)(x-4)}{(3-1)(3-4)} + 6 \frac{(x-1)(x-3)}{(4-1)(4-3)} =$$

$$= 12 \frac{(x^2 - 7x + 12)}{-2(-3)} + 4 \frac{(x^2 - 5x + 4)}{2(-1)} + 6 \frac{(x^2 - 4x + 3)}{3 \cdot 1} =$$

$$= 2(x^2 - 7x + 12) - 2(x^2 - 5x + 4) + 2(x^2 - 4x + 3) = 2x^2 - 12x + 22.$$

Таким образом, интерполяционный многочлен для заданной функции имеет вид $L_2(x) = 2x^2 - 12x + 22$.

Построим график $L_2(x)$ и точки в одной координатной плоскости.



Задания

Задание 1. По заданной таблице значений функции

x	x_0	x_1	x_2	x_3
y	y_0	y_1	y_2	y_3

составить формулу интерполяционного многочлена Лагранжа. Построить его график и отметить на нем узловые точки.

Задание 2. Вычислить с помощью калькулятора одно значение заданной функции для промежуточного значения аргумента с помощью интерполяционного многочлена Лагранжа и оценить погрешность интерполяции.

	<i>Вариант</i>									
	1	2	3	4	5	6	7	8	9	10
x_0	-1	2	0	7	-3	1	-1	2	-4	-1
x_1	0	3	2	9	-1	2	-1	4	-2	1,5
x_2	3	5	3	13	3	4	2	5	0	3
x_3	4	6	5	15	5	7	4	7	3	5
y_0	-3	4	-1	2	7	-3	4	9	2	4
y_1	5	1	-4	-2	-1	-7	9	-3	8	-7
y_2	2	7	2	3	4	2	1	6	5	1
y_3	-6	2	-8	-4	-6	8	6	-2	10	-8
x	3,8	3,5	0,5	4,8	4,1	3,9	3,3	4	2,9	5,3

Практическая работа № 6.

Тема: Нахождение интерполяционных многочленов сплайнами.

Интерполяция сплайнами

При большом количестве узлов интерполяции сильно возрастает степень интерполяционных многочленов, что делает их неудобными для вычислений.

Высокой степени многочлена можно избежать, разбив отрезок интерполяции на несколько частей, с последующим построением на каждой части самостоятельного интерполяционного многочлена.

Однако такое интерполирование наталкивается на существенный недостаток: в точках стыка разных интерполяционных многочленов бывает разрывной их первая производная.

В этом случае удобно пользоваться особым видом кусочно-полиномиальной интерполяции - интерполяции **сплайнами**.

Суть этого подхода заключается в следующем:

Определение: Функция $S_m(x)$ называется *интерполяционным сплайном* порядка m для функции $f(x)$, заданной таблицей:

x	x_0	x_1	$\dots$	x_i	$\dots$	x_n
y	y_0	y_1	$\dots$	y_i	$\dots$	y_n
	0	1		i		

если:

1. на каждом отрезке $[x_i; x_{i+1}]$ ($i=0, \dots, n-1$) $S(x)$ является многочленом порядка m ;
2. $S(x)$ и её производная до $(m-1)$ -го порядка включительно непрерывны на $[x_0; x_n]$;
3. $S(x_i)=y_i$ ($i=0, \dots, n$) - непосредственно условие интерполяции.

Остановимся на построении наиболее популярных в практике аппроксимации функций *кубических сплайнов*.

По определению кубический сплайн $S(x)$ можно представить в виде

$$S(x) = \begin{cases} P_1(x), & x \in [x_0; x_1]; \\ P_2(x), & x \in [x_1; x_2]; \\ \\ P_n(x), & x \in [x_{n-1}; x_n]. \end{cases} \quad (3.17)$$

Где каждый из $P_i(x)$ - многочлен третьей степени:

$$P_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, i = \overline{1, n} \quad (3.18)$$

Коэффициенты a_i найдем из условия: $y_i = S(x_i) = P_i(x_i) = a_i$, т.е. $a_i = y_i$ (3.19)

Условие непрерывности $S(x)$ в каждом узле приводит к равенствам:

$$P_i(x_i) = P_{i+1}(x_i), \quad i = \overline{1, n-1}$$

В развернутом виде с учетом формулы (3.18) эти равенства примут вид:

$$a_i = a_{i+1} + b_{i+1}(x_i - x_{i+1}) + C_{i+1}(x_i - x_{i+1})^2 + d_{i+1}(x_i - x_{i+1})^3 \quad (3.20)$$

Введем обозначения: $h_i = x_i - x_{i-1}$. Понижая в равенстве (3.20) индекс на единицу (меняем i на $i-1$) и,

учитывая (3.19), получим: $h_i b_i - h_i^2 c_i + h_i^3 d_i = y_i - y_{i-1}$ (3.21)

Условие непрерывности первой производной кубического сплайна сводится к

требованию $P_i'(x_i) = P_{i+1}'(x_i), \quad i = \overline{1, n-1}$.

используя, введем обозначения, получим:

Тогда дифференцируя формулу (3.18) и

$$b_{i-1} - b_i + 2h_i c_i - 3h_i^2 d_i = 0 \quad (i = \overline{2, n}) \quad (3.22)$$

Из условия непрерывности второй производной:

$$P_i''(x_i) = P_{i+1}''(x_i) \text{ получим:}$$

) (3.23). Составим систему из равенств (3.21)-(3.23) и, решив её, найдем коэффициенты b_i, C_i, d_i .

Однако, для однозначной ее разрешимости добавим условия непрерывности на концах отрезка:

$$S''(x_0) = S''(x_n) = 0, \quad P_1''(x_0) = 0, \quad P_n''(x_n) = 0 \quad \text{т.е.}$$

$$\begin{cases} C_1 - 3h_1 d_1 = 0 \\ a_n = 0 \end{cases} \quad (3.24)$$

В результате получаем систему уравнений:

$$\begin{cases} C_1 - 3h_1 d_1 = 0; \\ a_n = 0; \\ h_i b_i - h_i^2 c_i + h_i^3 d_i = y_i - y_{i-1}; \\ b_{i-1} - b_i + 2h_i c_i - 3h_i^2 d_i = 0; \\ C_{i-1} - C_i + 3h_i d_i = 0. \end{cases}$$

Последовательно, исключая переменные получим

$$h_{i+1} C_{i+1} + 2(h_i + h_{i+1}) C_i + h_i C_{i-1} = 3 \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right) \quad (3.25)$$

(это уравнение содержит лишь C_i).

$$d_i = \frac{C_i - C_{i-1}}{3h_i} \quad (3.26)$$

(это уравнение содержит лишь неизвестные d_i).

$$b_i = \frac{y_i - y_{i-1}}{h_i} + h_i C_i - h_i^2 d_i \quad (3.27)$$

(это уравнение содержит лишь неизвестные b_i).

Построив кубический сплайн, найдем оценку погрешности интерполяции:

$$|f(x) - S(x)| \leq \max_{[a;b]} |f^{(4)}(x)|, \quad \text{где } [a;b] \text{ - промежуток интерполяции.}$$

Пример: Построить кубический сплайн для функции $y=f(x)$, заданной таблицей:

x_i	-1	0	1	2
y_i	1/2	1	2	4

с дополнительным условием: $S''(-1) = S''(2) = 0$. Найти с помощью $S(x)$ значения функции при $x=0,3$. (Заметим, что в основу таблицы положена функция $y = 2^x$).

Решение: $C_0 = 0$ (т.к. не используется в функциях) и $C_3 = 0$ (т.к. из условия (3.24): $C_n = 0$).

Шаг таблицы $h_i = 1$.

из (3.25) получаем:

$$\begin{cases} 1 \cdot C_2 + 2(1+1)C_1 + 1 \cdot C_0 = 3 \left(\frac{y_2 - y_1}{1} - \frac{y_1 - y_0}{1} \right), \text{ при } i = 1, \\ 1 \cdot C_3 + 2(1+1)C_2 + 1 \cdot C_1 = 3 \left(\frac{y_3 - y_2}{1} - \frac{y_2 - y_1}{1} \right), \text{ при } i = 2. \end{cases}$$

$$\begin{cases} C_2 + 4C_1 = 3 \left(\frac{2-1}{1} - \frac{1-\frac{1}{2}}{1} \right) = 3 \left(1 - \frac{1}{2} \right) = \frac{3}{2}, \\ D + 4C_2 + C_1 = 3 \left(\frac{4-2}{1} - \frac{2-1}{1} \right) = 3(2-1) = 3. \end{cases}$$

$$\begin{cases} C_2 + 4C_1 = \frac{3}{2}, \\ 4C_2 + C_1 = 3. \end{cases}$$

$$15C_1 = 6 - 3 = 3,$$

$$C_1 = \frac{1}{5}, \quad C_2 = \frac{3}{2} - \frac{4}{5} = \frac{7}{10}.$$

Из (3.26) имеем:

$$d_1 = \frac{C_1 - C_0}{3h} = \frac{\frac{1}{5} - 0}{3 \cdot 1} = \frac{1}{15},$$

$$d_2 = \frac{C_2 - C_1}{3h} = \frac{\frac{7}{10} - \frac{1}{5}}{3} = \frac{1}{6},$$

$$d_3 = \frac{C_3 - C_2}{3h} = \frac{0 - \frac{7}{10}}{3} = -\frac{7}{30}.$$

Из (3.27) имеем:

$$b_1 = \frac{y_1 - y_0}{h} + hC_1 - h^2 d_1 = \frac{1 - \frac{1}{2}}{1} + 1 \cdot \frac{1}{5} - 1^2 \cdot \frac{1}{15} = \frac{1}{2} + \frac{1}{5} - \frac{1}{15} = \frac{19}{30}$$

$$b_2 = \frac{y_2 - y_1}{h} + hC_2 - h^2 d_2 = \frac{2-1}{1} + 1 \cdot \frac{7}{10} - 1 \cdot \frac{1}{6} = 1 + \frac{7}{10} - \frac{1}{6} = \frac{23}{15},$$

$$b_3 = \frac{y_3 - y_2}{h} + hC_3 - h^2 d_3 = \frac{4-2}{1} + 1 \cdot 0 - 1 \cdot \left(-\frac{7}{30} \right) = 2 + \frac{7}{30} = \frac{67}{30}.$$

из формулы (3.28) получаем:

$$P_1(x) = a_1 + b_1(x - x_1) + C_1(x - x_1)^2 + d_1(x - x_1)^3, x \in [x_0; x_1]$$

$$P_1(x) = 1 + \frac{19}{30}(x - 0) + \frac{1}{5}(x - 0)^2 + \frac{1}{15}(x - 0)^3, \text{ т.е. } P_1(x) = 1 + \frac{19}{30}x + \frac{1}{5}x^2 + \frac{1}{15}x^3, x \in [-1; 0]$$

$$P_2(x) = a_2 + b_2(x - x_2) + C_2(x - x_2)^2 + d_2(x - x_2)^3, x \in [x_1; x_2],$$

$$P_2(x) = 2 + \frac{23}{15}(x - 1) + \frac{7}{10}(x - 1)^2 + \frac{1}{6}(x - 1)^3, x \in [x_1; x_2]$$

$$P_3(x) = a_3 + b_3(x - x_3) + C_3(x - x_3)^2 + d_3(x - x_3)^3, x \in [x_2; x_3],$$

$$P_3(x) = 4 + \frac{67}{30}(x - 2) - \frac{7}{30}(x - 2)^3, x \in [1; 2]$$

Следовательно, сплайн $S(x)$ построен:

$$S(x) = \begin{cases} P_1(x) = 1 + \frac{19}{30}x + \frac{1}{5}x^2 + \frac{1}{15}x^3, x \in [-1; 0]; \\ P_2(x) = 2 + \frac{23}{15}(x - 1) + \frac{7}{10}(x - 1)^2 + \frac{1}{6}(x - 1)^3, x \in [0; 1]; \\ P_3(x) = 4 + \frac{67}{30}(x - 2) - \frac{7}{30}(x - 2)^3, x \in [1; 2]. \end{cases}$$

Найдем его значение при $x=0,3$:

: Заметим, что $0,3 \in [0;1]$, поэтому используем многочлен $P_2(x)$

$$P(0.3) = 2 + \frac{23}{15}(0,3-1) + \frac{7}{10}(0,3-1)^2 + \frac{1}{6}(0,3-1)^3 = 1,2125$$

Отметим для сопоставления с той же точностью значение функции, положенной в основу данного примера: $f(x) = 2^x$; $f(0.3) = 2^{0,3} = 1,2311$.

Интерполяция сплайнами сопряжена с немалым объемом вычислительной работы. Весьма необычна и форма окончательного результата, ибо сплайн имеет различные представления на различных частичных отрезках интерполяции. Это осложняет доступ к значениям сплайна в каждой конкретной точке, так как предполагает, прежде всего, поиск параметров, определяющих соответствующую форму сплайна. Эти трудности легко предотвратимы при использовании компьютера, так как упорядоченное хранение всех необходимых параметров организовать нетрудно, а выполнение однотипных процедур по вычислению параметров сплайна и его значений может быть обеспечено специальными процедурами.

Задания.

Задание 1. По заданной таблице значений функции

x	x_0	x_1	x_2	x_3
y	y_0	y_1	y_2	y_3

вычислить коэффициенты и составить формулы кубического сплайна.

Задание 2. Результат интерполирования проверить путем вычисления значений сплайна узловых точках.

	Вариант									
	1	2	3	4	5	6	7	8	9	10
x_0	-1	2	0	7	-3	1	-1	2	-4	-1
x_1	0	3	2	9	-1	2	1	4	-2	0
x_2	1	4	4	11	1	3	3	6	0	1
x_3	2	5	6	13	3	4	5	7	2	2
y_0	-3	4	-1	2	7	-3	4	9	2	4
y_1	5	1	-4	-2	-1	-7	9	-3	8	-7
y_2	2	7	2	3	4	2	1	6	5	1
y_3	-6	2	-8	-4	-6	8	6	-2	10	-8
x	3,8	3,5	0,5	4,8	4,1	3,9	3,3	4	2,9	5,3

ОП.11 Компьютерные сети

Практическая работа №1 Монтаж кабельных сетей технологий Ethernet. Построение одноранговой сети. Построение схемы компьютерной сети.

Преимущества и недостатки проводных сетей

Структурированная кабельная система (СКС) – законченная совокупность кабелей связи и коммутационного оборудования, отвечающая требованиям соответствующих нормативных документов.

СКС включает набор кабелей и коммутационных элементов, и методику их совместного использования, позволяющую создавать регулярные расширяемые структуры связей в локальных сетях различного назначения. СКС – физическая основа инфраструктуры здания, позволяющая свести в единую систему множество сетевых информационных сервисов разного назначения: локальные вычислительные сети и телефонные сети, системы безопасности, видеонаблюдения и т. д.

СКС представляет собой иерархическую кабельную систему, смонтированную в здании или в группе зданий, состоящую из структурных подсистем. В состав СКС входят такие элементы, как главный кросс (МС), кабель магистральной подсистемы первого и второго уровня, промежуточные кроссы (ИС), горизонтальные кроссы (НС) и кабели горизонтальной подсистемы, а также консолидационные точки (СР), многопользовательские телекоммуникационные розетки (МуТОА или МуТО) и телекоммуникационные розетки (ТО) и другие. Система может быть построена на основе медных или оптических кабелей, все элементы СКС интегрируются в единый комплекс (систему) и эксплуатируются согласно определённым правилам.

Преимущества:

- при относительно высокой начальной стоимости оправдывает капиталовложения за счет длительной и безотказной эксплуатации;
- доступная оптическая и медная УТР-технология;
- обеспечивает модульность и возможность наращивания и легкого внесения изменений;
- способность к передаче на высокой скорости и полосе пропускания;
- открытая архитектура допускает одновременное использование различных протоколов и продуктов различных пользователей;
- не зависит от изменений информационной технологии;
- допускает использование уже существующего активного оборудования;
- использует стандартные компоненты и материалы;
- допускает обслуживание минимальным количеством персонала пользователя;
- соответствует всем существующим стандартам по кабельной проводке здания в настоящее время и в будущем.

Недостатки:

- высокая стоимость проектирования и инсталляции.

Простые (неструктурированные) кабельные сети представляют из себя обычные кабельные системы на основе витой пары, проложенные в кабельных каналах. Для передачи данных и офисной телефонии используются разные кабельные системы. Часто

представляют из себя очень печальное зрелище: провода запутаны, просто валяются на полу, под столами.

Преимущества:

- низкая стоимость монтажа по сравнению с СКС;
- сравнительно высокая скорость монтажа;
- высокая надежность.

Недостатки:

- небольшая гарантия на систему;
- сложность расширения системы, дополнительные затраты на расширение.

Монтаж локальной сети

При монтаже проводных локальных сетей нет ничего предельно сложного или требующего каких-то специальных навыков или знаний. Достаточно запомнить несколько простых правил и следовать им.

Основным носителем в подавляющем большинстве проводных сетей Ethernet сейчас является **витая пара**. Этот кабель делится по категориям. Наиболее распространен кабель маркировкой UTP-8 Cat 5 или 5e, состоящий из 4-х пар (8 проводников) (рис. 1).

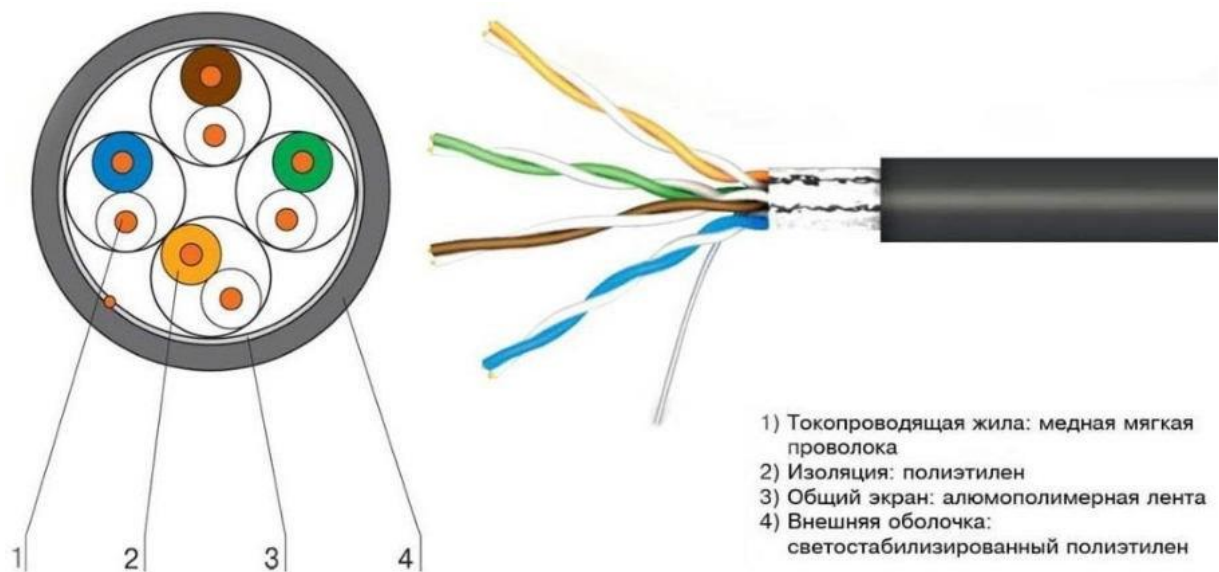


Рис. 1. Витая пара

Каждая пара скручена со своим определённым шагом, что служит защитой от помех. Проводники в парах имеют цветную маркировку: оранжевый+бело-оранжевый, зелёный+бело-зелёный, синий+бело-синий, коричневый+бело-коричневый. Маркировка двухцветных проводов может быть выполнена одним из двух способов: «колечками» — когда дополнительный цвет нанесён на белый проводник через определённое расстояние небольшими колечками, либо «полосками» — когда проводник поделён вдоль пополам, одна половина белого цвета, вторая — дополнительного. Предпочесть следует кабель второго типа по одной простой причине: после зачистки кабеля от изоляции на одном из двухцветных проводов ближайшее колечко может остаться под изоляцией, и мы будем иметь проводник неопределённого белого цвета.

Максимальная длина отрезка кабеля, используемого между двумя устройствами, по стандарту не может быть больше 100 метров. Розетки, переходники и прочие пассивные элементы устройствами в данном контексте не считаются. Устройство принимает сигнал и отправляет дальше, предварительно усилив его. В небольших сетях это концентраторы (hub), коммутаторы (switch hub), мосты (bridge) и маршрутизаторы (router). Ну и, конечно же, сами компьютеры с сетевыми картами. Давайте остановимся на сетевых устройствах чуть более подробно, ведь, в конечном счёте, именно от них в большей степени зависит качество работы и удобство использования нашей будущей сети.

Концентратор – это многопортовый повторитель. То есть он принимает сигнал на один из своих портов (порт – это, проще говоря, гнездо, в которое вставляется сетевой кабель), и, усилив его, передаёт его на все остальные свои порты. Это один из самых дешёвых способов объединить компьютеры в локальную сеть, но у него есть один существенный недостаток. Допустим, в сеть соединены четыре компьютера. Так вот, если с первого компьютера пользователь передаёт файл на второй и, в это же время, с третьего компьютера пользователь передаёт файл на четвёртый, то скорость передачи падает более чем вдвое по сравнению с максимальной. Это происходит из-за того, что сигналы каждого из компьютеров передаются на все порты. Возникают так называемые «коллизии». Для решения этой проблемы были созданы коммутаторы. Внешне они ничем, как правило, не отличаются от концентраторов (да и по стоимости не очень заметно), но работают они принципиально по-другому. При включении компьютера в сеть коммутатор запоминает его физический адрес (MAC-адрес). Таким образом, если с первого компьютера файл передаётся на второй, а с третьего на четвёртый, то коммутатор сигнал, пришедший на первый порт, усиливает и передаёт только на второй порт, а сигнал, пришедший на третий

– только на четвёртый. Таким образом, за счёт этого две передачи файлов происходят одновременно, причём на максимально возможной скорости и безо всяких коллизий.

Мосты используются для связи в одно целое двух отдельных сегментов сети. То есть, допустим, у нас в одном кабинете 4 компьютера, объединённые в локальную сеть, и в другом кабинете 3 компьютера, также объединённые в локальную сеть. Расстояние между кабинетами равно 150 метрам. Нам необходимо объединить эти две сети в одну. Просто протянуть витую пару у нас не получится (как мы помним, её максимальная длина – 100 метров). Значит, надо либо где-то на посторонней территории устанавливать ретранслятор (усилитель сигнала), а это не всегда возможно, либо использовать, например, коаксиальный кабель (его длину стандарт ограничивает уже 200 метрами). Либо, даже при гораздо меньшем расстоянии, но при невозможности проложить вообще какой-либо кабель, использовать беспроводную связь (Wi-Fi). Так вот, мост как раз и даёт возможность соединять сети, построенные на одном носителе (в нашем случае, витой паре), с помощью другого носителя (коаксиального кабеля или радиоволн) в одну сеть. Либо просто объединять в одну сеть два сегмента, построенные на разных носителях (допустим, все стационарные компьютеры подключены к сети с помощью кабеля, а ноутбуки и планшеты – по Wi-Fi).

Маршрутизаторы или роутеры. Сейчас для небольших домашних и офисных сетей это наиболее распространённые устройства. Недостаток коммутатора состоит, прежде всего, в том, что для того, чтобы обеспечить всем выход в Интернет по одной линии (одному подключению), необходимо для этого настроить один из компьютеров, который для этого будет подключён и к локальной сети

и к Интернету. Более того, для обеспечения выхода остальных пользователей в Интернет этот компьютер должен быть постоянно включён даже тогда, когда его пользователь на нём не работает. То же самое относится и к другим общим устройствам, например, принтерам или сканерам (если они используются по сети совместно).

Роутер объединяет в себе функции коммутатора, моста и такого компьютера. То есть он подключается к Интернету, а все остальные подключены к нему как к коммутатору (по кабелю или по Wi-Fi). Наиболее продвинутые модели также позволяют подключать к себе принтеры, сканеры и внешние жёсткие диски при помощи USB. Так же они содержат программное обеспечение, которое может постоянно работать без включения компьютера (например, торрент-клиент). При выборе роутера необходимо учесть все потребности и откинуть всё лишнее (так, для дома, вероятнее всего, вряд ли понадобится 12-и портовый роутер со встроенным принт-сервером. А вот в небольшом офисе это будет практически идеальный вариант. Также выбор модели зависит от способа вашего подключения к провайдеру (наиболее распространены сейчас, по понятным причинам, Ethernet-роутеры и ADSL-роутеры).

Ниже рассмотрим, как создать небольшую локальную сеть, центральным узлом которой будет являться роутер. Для начала надо определиться с его месторасположением. Оно должно отвечать следующим требованиям:

1. Роутер должен быть легко доступен, и к нему без затруднений должен подключаться кабель провайдера Интернет.
2. В непосредственной близости от места размещения роутера (не дальше 1 метра) должна быть электророзетка.
3. Общая длина кабеля, идущего на подключение компьютеров к роутеру, должна быть минимальной.

Кабели обычно проложены либо методом скрытой проводки (в стене), либо в кабель-каналах и под фальшпотолком. На их конце монтируются розетки под разъёмы RJ-45 (рис. 2), а компьютеры в эти розетки подключаются с помощью так называемых патч-кордов. **Патч-корд** – это отрезок гибкой витой пары, длиной от 75 см до 1 м, на обоих концах которого находятся разъёмы RJ-45. Их можно приобрести в любом магазине, торгующем компьютерным оборудованием либо изготовить самостоятельно.

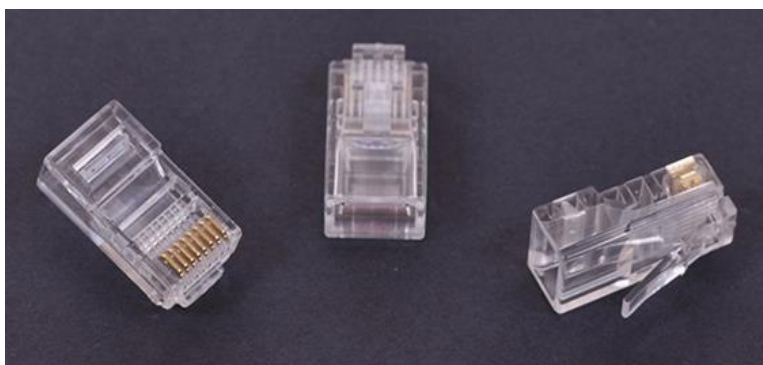


Рис. 2. Разъёмы RJ-45

Для присоединения кабеля «витая пара» к розеткам и для присоединения к кабелю разъёмов RJ-45 нам понадобятся два инструмента:

- **обжимной инструмент RJ-45** (обжимные клещи или кримпер, рис. 3) служит для закрепления на концах кабеля разъёмов

- **устройство для зачистки RJ-45** (куттер, рис. 4) – служит для закрепления кабеля в розетке.



Рис. 4. Устройство для зачистки RJ-45

У разъёма RJ-45 восемь контактов. Нумеруются они следующим образом: если взять разъём и повернуть его фиксирующим язычком

вниз, чтобы металлические контакты были направлены от себя, то контакты слева направо нумеруются с 1 по 8.

Для фиксации разъёма на кабеле («обжима»), необходимо сделать следующие операции (рис. 5):

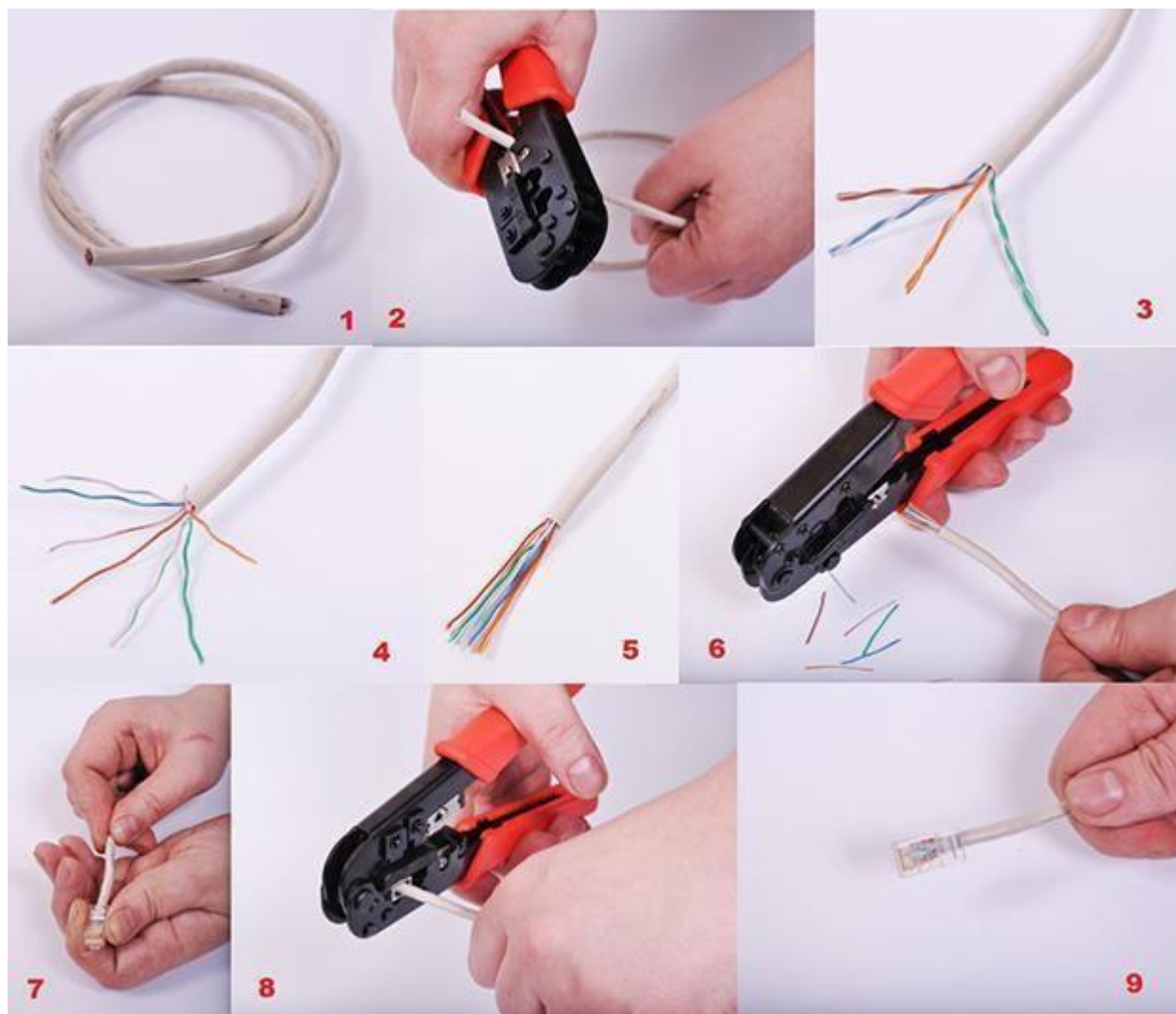


Рис. 5. Операции «обжима» витой пары

- зачистить конец кабеля от изоляции (рис. 5 (1,2));
- раскрутить витые пары и разложить провода в линейку согласно цветовой схемы (рис. 5 (3,4,5));
- срезать лишние провода до необходимой длины (чтобы провода помещались в разъём, а изоляция кабеля входила в него на несколько миллиметров) (рис. 5 (6));
- вставить провода в разъём, внимательно следя за тем, чтобы последовательность проводов не была нарушена (рис. 5 (7));
- вставить разъём в кримпер и сдавливанием рукояток зафиксировать разъём на кабеле (рис. 5 (8));
- извлечь разъём из клещей (рис. 5 (9)).

В настоящее время применяются две схемы обжима: TIA/EIA-568A и TIA/EIA-568B, который используется гораздо чаще. Расположение проводников в них следующее:

№ контакта	TIA/EIA-568A	TIA/EIA-568B	Crossover 1000 Mbit/s
1	Бело-зелёный	Бело-оранжевый	Бело-зелёный
2	Зелёный	Оранжевый	Зелёный
3	Бело-оранжевый	Бело-зелёный	Бело-оранжевый
4	Синий	Синий	Бело-коричневый
5	Бело-синий	Бело-синий	Коричневый
6	Оранжевый	Зелёный	Оранжевый
7	Бело-коричневый	Бело-коричневый	Синий
8	Коричневый	Коричневый	Бело-синий

При соединении компьютера с роутером оба конца кабеля обжимаются одинаково. Для соединения же однотипного оборудования (компьютер соединяется напрямую с компьютером либо, например, два концентратора соединяются между собой), используется так называемая «перекрёстная» схема обжима (crossover). Она также существует в двух вариантах. Первый вариант (который применяется на скоростях до 100 мбит/с и поэтому является наиболее распространённым), один конец кабеля обжимается по схеме «А», а второй – по схеме «В». Для скорости же до 1000 мбит/с применяются следующие схемы обжима: один конец кабеля обжимается по схеме «В», а второй – по схеме «Crossover 1000 Mbit/s».

Обжим розеток производить ещё проще. На розетках уже нанесены цветовые маркировки. Так что, в зависимости от выбранной схемы обжима «А» или «В» вам надо произвести следующие действия:

- с помощью куттера зачистить конец кабеля от изоляции;
- раскрутить пары проводов;
- наложить каждый провод на своё гнездо и с помощью носика куттера сдвинуть их вниз по лезвиям гнезда до упора;
- закрепить розетку в подрозетнике.

После этого необходимо с помощью патч-кордов соединить компьютеры с розетками локальной сети.

Настройка локальной сети

После того, как все компоненты локальной сети физически соединены в одно целое, необходимо их настроить. Первым делом настраиваем центральное устройство нашей сети, то есть роутер. Узнать в подробностях, как это делается, Вы можете здесь. С компьютерами же ещё проще. Обычно все компьютеры под управлением любой версии Windows по-умолчанию настроены как клиенты DHCP.

DHCP – это сервер, который занимается тем, что каждому подключающемуся к сети компьютеру либо другому устройству он выдаёт по заданным правилам все необходимые сетевые настройки, получив которые, компьютер подключается к сети и

может в ней работать. После удачного входа в сеть этот ключ компьютер запоминает, и при последующих подключениях необходимость в его вводе отсутствует.

Но иногда (правда, довольно редко), возникает необходимость в настройке сетевых свойств вручную. Для этого нам необходимо будет зайти в «Центр управления сетями и общим доступом». Проще всего это сделать, кликнув по значку подключения сети в трее и выбрав там «Центр управления сетями и общим доступом» (рис. 6).

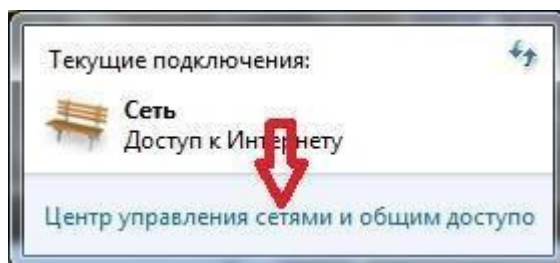


Рис. 6. Центр управления сетями и общим доступом

Либо зайти в меню «Пуск», далее в «Панель управления», в ней войти в раздел «Сеть и Интернет» и там выбрать «Центр управления сетями и общим доступом».

Там, в зависимости от того, какое подключение Вы настраиваете, Вам необходимо будет выбрать либо «Подключение по локальной сети» (1), либо «Беспроводное сетевое соединение» (2) (рис. 7).

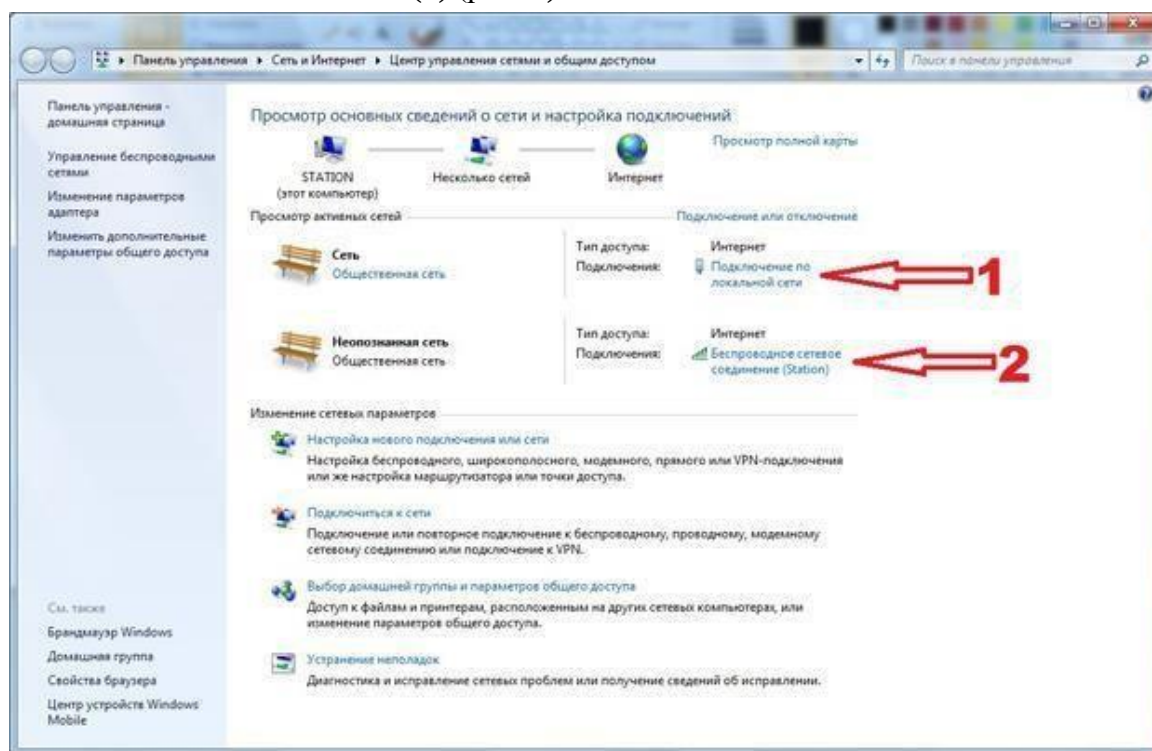


Рис. 7. Настройка сетевых подключений

Ручная настройка сети, вне зависимости от типа подключения, производится следующим образом.

В окне подключения кликнуть по кнопке «Свойства» (рис. 8).

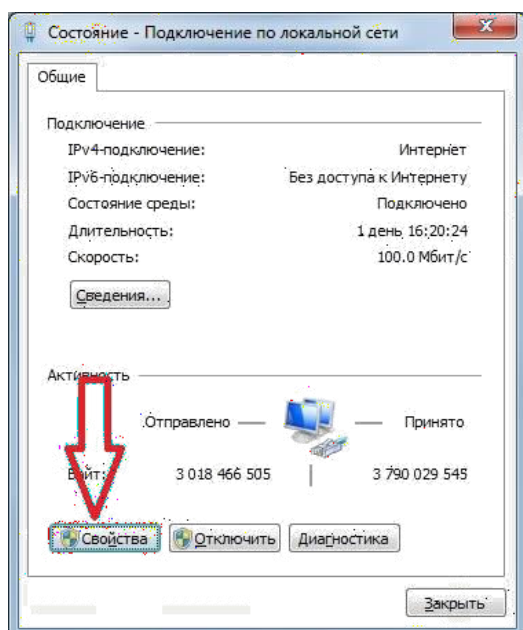


Рис. 8. Окно состояния подключения по локальной сети

Далее необходимо выбрать «Протокол Интернета 4 (TCP/IPv4)» и нажать «Свойства» (рис. 9).

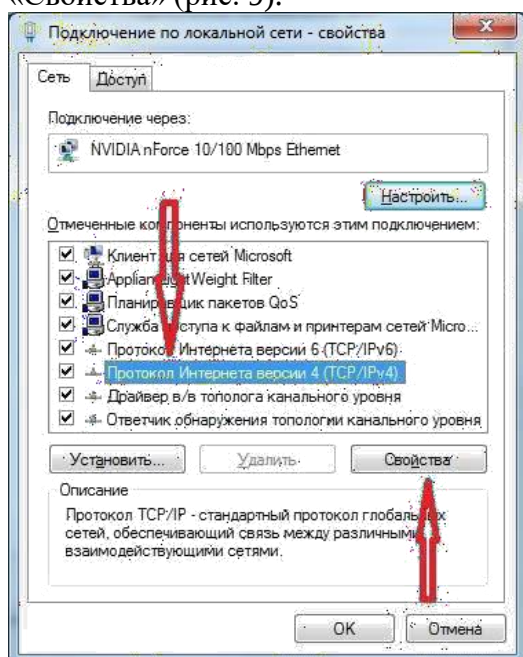


Рис. 9. Настройка параметров подключения по локальной сети

В свойствах необходимо будет ввести IP-адрес Вашего компьютера, маску подсети, шлюз по умолчанию и один либо два адреса (основной и резервный) сервера доменных имён DNS (рис. 10).

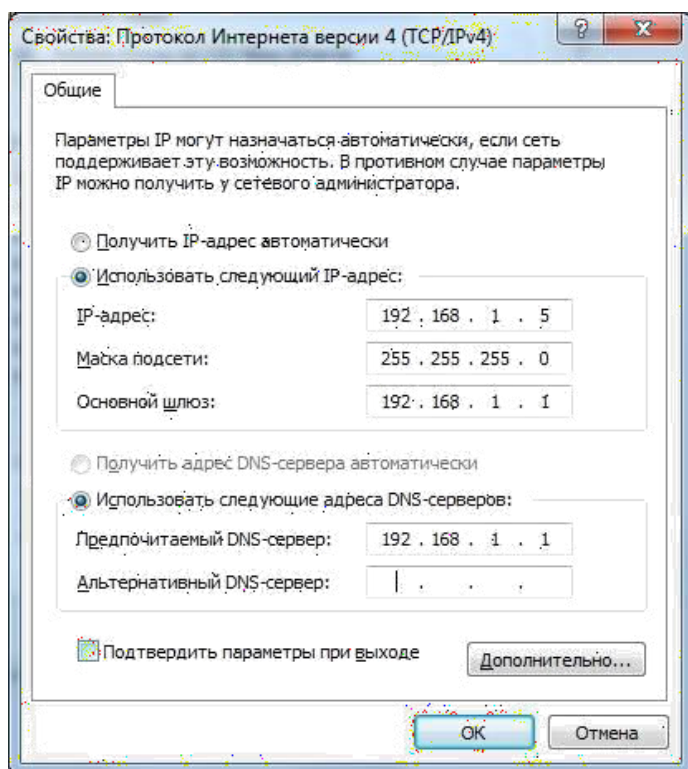


Рис. 10. Настройка параметров протокола TCP/IP

Сервер DNS занимается тем, что по символьному адресу URL, который Вы вводите в браузере, например, www.fixitbook.ru, выдаёт сопоставленный ему IP-адрес, с помощью которого Ваш компьютер может подключиться к нужному и показать Вам тот сайт, который Вам необходим. Сделана эта служба была потому, что названия из слов подавляющее большинство людей запоминает гораздо лучше и проще, нежели последовательность случайных цифр, которую представляет из себя IP-адрес.

Практическая работа №2 «Настройка протокола TCP/IP в операционных системах» Цель: обобщение и систематизация знаний по теме «Межсетевое взаимодействие»

Средства для выполнения работы:

- аппаратные: компьютер, с установленной ОС Windows XP;
- программные: виртуальные машины: VM-1;
- информационные: IP-адрес; маска подсети; основной шлюз; предпочитаемый DNS.

Задания к работе

1. Проверьте работоспособность стека протоколов TCP/IP.

Запустите виртуальную машину VM-1 и загрузите ОС Windows.

Запустите консоль (*Пуск/Программы/Стандартные/Командная строка*).

В командной строке введите `ipconfig /all / more`.

Используя приведенную ниже информацию, создайте в своей папке текстовый документ со следующими данными:

- имя компьютера;
- основной DNS-суффикс;
- описание DNS-суффикса для подключения;
- физический адрес;
- DHCP включен;
- автоконфигурация включена;
- IP-адрес автоконфигурации;
- маска подсети;
- шлюз по умолчанию.

Убедитесь в работоспособности стека *TCP/IP*, отправив эхо-запросы на IP-адреса. Для этого воспользуйтесь командой `ping`:

- отправьте эхо-запросы на локальный адрес компьютера (loopback) `ping 127.0.0.1` (на экране должны появиться сообщения о полученном ответе от узла 127.0.0.1);
- отправьте эхо-запрос по другому IP-адресу, например 172.21.5.1.

2. Настройте стек протоколов TCP/IP для использования статического IP-адреса.

Откройте окно Сетевые подключения (*Пуск/Панель управления/Сетевые подключения*).

Вызовите свойства подключения по локальной сети. Для этого можно воспользоваться контекстным меню.

В появившемся диалоговом окне на вкладке Общие откройте свойства Протокол Интернета TCP/IP.

Щелкните переключатель *Использовать следующий IP-адрес* и введите в соответствующие поля данные: IP_адрес; Маску подсети; Основной шлюз; Предпочитаемый DNS.

Примените параметры кнопкой *ОК*.

Закройте окно свойств подключения кнопкой *ОК* (если потребуется, то согласитесь на перезагрузку компьютера).

Проверьте работоспособность стека протоколов *TCP/IP*.

3. Настройте TCP/IP для автоматического получения IP-адреса.

Откройте окно Сетевые подключения.

Вызовите свойства Подключения по локальной сети.

Откройте свойства Протокол Интернета TCP/IP.

Установите переключатель *Получить IP-адрес автоматически*.

Закройте диалоговое окно Свойства: Протокол Интернета TCP/IP кнопкой *ОК*.

Примените параметры кнопкой *ОК*.

Проверьте настройку стека протоколов *TCP/IP*.

Получите другой адрес для своего компьютера. Для этого:

- запустите консоль (командную строку);
- введите команду для сброса назначенных адресов - `ipconfig /release`;
- введите команду для получения нового адреса `ipconfig / renew`;

Проверьте работоспособность стека протоколов *TCP/IP*.

Контрольные вопросы:

1. Опишите параметры, используемые при настройке статического адреса TCP/IP.
2. Какие преимущества дает применение стека протоколов TCP/IP.
3. Дайте определение понятию стек протоколов TCP/IP.

Практическая работа №3 «Работа с диагностическими утилитами протокола TCP/IP»Цель: обобщение и систематизация знаний по теме «Межсетевое взаимодействие»

Задания к работе

Задание 1. Получение справочной информации по командам.

Выведите на экран справочную информацию по всем рассмотренным утилитам (см. таблицу п.1). Для этого в командной строке введите имя утилиты без параметров и дополните /?.

Сохраните справочную информацию в отдельном файле. Изучите

ключи, используемые при запуске утилит.

Задание 2. Получение имени хоста.

Выведите на экран имя локального хоста с помощью команды hostname. Сохраните результат в отдельном файле.

Задание 3. Изучение утилиты ipconfig.

Проверьте конфигурацию TCP/IP с помощью утилиты ipconfig. Заполните таблицу:

Имя хоста	
IP-адрес	
Маска подсети	
Основной шлюз	
Используется ли DHCP (адрес DHCP-сервера)	
Описание адаптера	
Физический адрес сетевого адаптера	
Адрес DNS-сервера	
Адрес WINS-сервера	

Задание 4. Тестирование связи с помощью утилиты ping.

1. Проверьте правильность установки и конфигурирования TCP/IP на локальном компьютере.
2. Проверьте функционирование основного шлюза, послав 5 эхо-пакетов длиной 64 байта.
3. Проверьте возможность установления соединения с удаленным хостом.
4. С помощью команды ping проверьте адреса (взять из списка локальных ресурсов на сайте aspu.ru) и для каждого из них отметьте время отклика. Попробуйте изменить параметры команды ping таким образом, чтобы увеличилось время отклика. Определите IP-адреса узлов.

Задание 5. Определение пути IP-пакета.

С помощью команды tracert проверьте для перечисленных ниже адресов, через какие промежуточные узлы идет сигнал. Изучите ключи команды.

- a) aspu.ru
- b) mathmod.aspu.ru
- c) yarus.aspu.ru

Задание 6: Просмотр ARP-кэша.

С помощью утилиты arp просмотрите ARP-таблицу локального компьютера. Внести в кэш локального компьютера любую статическую запись.

Задание 7: Просмотр локальной таблицы маршрутизации.

С помощью утилиты route просмотреть локальную таблицу маршрутизации.

Задание 8. Получение информации о текущих сетевых соединениях и протоколах стека TCP/IP.

С помощью утилиты netstat выведите перечень сетевых соединений и статистическую информацию для протоколов UDP, TCP, ICMP, IP.

Контрольные вопросы:

1. Раскрыть термины: хост, шлюз, хоп, время жизни пакета, маршрут, маска сети, авторитетный/неавторитетный (компетентный) DNS-сервер, порт TCP, петля обратной связи, время отклика.
2. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP?
3. Каким образом команда ping проверяет соединение с удаленным хостом?
4. Каково назначение протокола ARP?
5. Как утилита ping разрешает имена узлов в ip-адреса (и наоборот)?
6. Какие могут быть причины неудачного завершения ping и tracert? (превышен интервал ожидания для запроса, сеть недоступна, превышен срок жизни при передаче пакета).
7. Всегда ли можно узнать символьное имя узла по его ip-адресу?
8. Какой тип записи запрашивает у DNS-сервера простейшая форма nslookup?

Практическая работа №4 «Решение проблем с TCP/IP» Цель: обобщение и

систематизация знаний по теме «Межсетевое взаимодействие»

Задания к работе

1. Открыть окно командной строки, ввести команду `ping` с IP адресом машины, при взаимодействии с которой возникают проблемы. Определить, использует ли проблемная машина конфигурацию статичного или динамичного IP адреса. Для этого откройте панель управления и выберите опцию Сетевые подключения. Теперь правой клавишей нажмите на подключении, которое собираетесь диагностировать, затем выберите опцию Свойства в появившемся меню быстрого доступа.
2. Перейдите по спискам элементов, используемых подключением, пока не дойдете до TCP/IP протокола (выбран на рисунке 3). Выберите этот протокол, нажмите на кнопке Свойства, чтобы открыть страницу свойств для Internet Protocol (TCP/IP).
3. Запишите IP конфигурацию машины. Особенно важно сделать заметки следующих элементов:
 - а. Использует ли машина статичную или динамичную конфигурацию?
 - б. Если используется статичная конфигурация, запишите значение IP адреса, маски подсети и основного шлюза?
 - в. Получает ли машина адрес DNS сервера автоматически?
 - г. Если адрес DNS сервера вводится вручную, то какой адрес используется?
4. Если на компьютере установлено несколько сетевых адаптеров, то в панели управления будут перечислены несколько сетевых подключений.
5. Проверьте тип адаптера.
6. Определите, принимает ли Windows такую конфигурацию. Для этого откройте окно командной строки и введите следующую команду: `IPCONFIG /ALL`.
7. Определите правильный сетевой адаптер. В этом случае определение нужного адаптера довольно простое, поскольку в списке есть всего лишь один адаптер.
8. Отправьте `ping` запрос на адрес локального узла. Существует два различных способа того, как это сделать. Одним способом является ввод команды: `PING LOCALHOST`.
9. Введите команду `Nslookup`, за которой должно идти полное доменное имя удаленного узла. Команда `Nslookup` должна суметь разрешить полное доменное имя в IP адрес.
11. Необходимо просканировать клиентскую машину на предмет вредоносного ПО. Если на машине не обнаружено вредоносного ПО, сбросьте DNS кэш путем ввода следующей команды: `IPCONFIG /FLUSHDNS`.

Контрольные вопросы

1. Поясните, что может означать, если время TTL закончилось до получения ответа.
2. Как подтвердить наличие сетевого соединения?
3. Что показывает команда `IPCONFIG /ALL`?
4. Что означает наличие IP адрес со значением 0.0.0.0.?
5. С помощью какой команды можно проверить то, что конфигурация IP адреса работает корректно, и что отсутствуют проблемы с стеком локального протокола TCP/IP?
6. Как производится опрос основного шлюза?
7. Как производится опрос DNS сервера?

Практическая работа №5 «Преобразование форматов IP-адресов. Расчет IP-адреса и маски подсети»

Цель работы: определение класса и расчет IP-адреса и маски подсети.

Общие положения

IP-адрес представляет собой 32-разрядное двоичное число, разделенное на группы по 8 бит, называемых *октетами*.

Наиболее распространенной формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в *десятичной форме* и разделенных точками, например: 128.10.2.30

Этот же адрес может быть представлен в *двоичном формате*: 10000000 00001010 00000010 00011110.

А также в *шестнадцатеричном формате*: 80.0A.02.1D

Следует заметить, что максимальное значение октета равно 11111111 (двоичная система счисления), что соответствует в десятичной системе 255.

Поэтому IP-адреса, в которых хотя бы один октет превышает это число, являются недействительными. Пример: 172.16.123.1 – действительный адрес, 172.16.123.256 – несуществующий адрес, поскольку 256 выходит за пределы допустимого диапазона.

IP-адрес состоит из двух логических частей – **номера подсети (ID подсети)** и **номера узла (ID хоста)** в этой подсети. При передаче пакета из одной подсети в другую используется ID подсети. Когда пакет попал в подсеть назначения, ID хоста указывает на конкретный узел в рамках этой подсети.

Чтобы записать ID подсети, в поле номера узла в IP-адресе ставят нули. Чтобы записать ID хоста, в поле номера подсети ставят нули. Например, если в IP-адресе 172.16.123.1 первые два байта отводятся под номер подсети, остальные два байта – под номер узла, то номера записываются следующим образом:

ID подсети: 172.16.0.0.

ID хоста: 0.0.123.1.

По числу разрядов, отводимых для представления номера узла (или номера подсети), можно определить общее количество узлов (или подсетей) по простому правилу: если число разрядов для представления номера узла равно N , то общее количество узлов равно $2^N - 2$. Два узла вычитаются

вследствие того, что адреса со всеми разрядами, равными нулям или единицам, являются особыми и используются в специальных целях.

Например, если под номер узла в некоторой подсети отводится два байта (16 бит), то общее количество узлов в такой подсети равно $2^{16} - 2 = 65534$ узла.

Для определения того, какая часть IP-адреса отвечает за ID подсети, а какая за ID хоста, применяются два способа:

- 1) с помощью классов
- 2) с помощью масок.

Общее правило: под ID подсети отводятся *первые* несколько бит IP-адреса, оставшиеся биты обозначают ID хоста.

Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса.

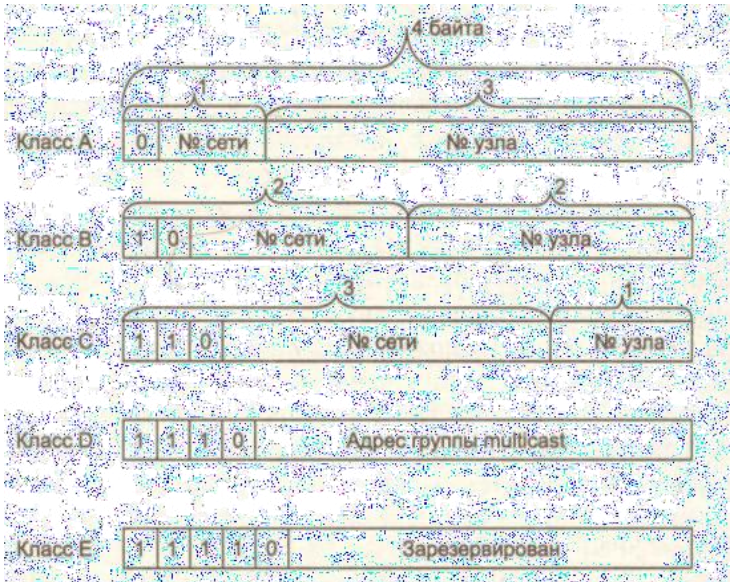


Таблица - Классы IP-адресов

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Количество сетей	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	126	$2^{24} - 2 = 16777214$
B	10	128.0.0.0	191.255.0.0	16384	$2^{16} - 2 = 65534$
C	110	192.0.1.0	223.255.255.0	2097152	$2^8 - 2 = 254$
D	1110	224.0.0.0	239.255.255.255	Групповой адрес	
E	11110	240.0.0.0	247.255.255.255	Зарезервирован	

Адреса **класса А** предназначены для использования в больших сетях общего пользования.

Они допускают большое количество номеров узлов.

Адреса **класса В** используются в сетях среднего размера, например, сетях университетов и крупных компаний.

Адреса класса С используются в сетях с небольшим числом компьютеров.

Адреса класса D используются при обращениях к группам машин.

Адреса класса E зарезервированы на будущее.

Некоторые IP-адреса являются особыми, они не должны применяться для идентификации обычных сетей:

- Если все биты IP-адреса равны нулю, адрес обозначает узел-отправитель и используется в некоторых сообщениях ICMP.
- Если все биты ID сети равны 1, адрес называется *ограниченным широковещательным (limited broadcast)*, пакеты, направленные по такому адресу, рассылаются всем узлам той подсети, в которой находится отправитель пакета.
- Если все биты ID хоста равны 1, адрес называется *широковещательным (broadcast)*, пакеты, имеющие широковещательный адрес, доставляются всем узлам подсети назначения.
- Если все биты ID хоста равны 0, адрес считается идентификатором подсети (subnet ID).

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является *внутренним адресом стека протоколов* компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети. В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со значения 127. Когда программа посылает данные по IP-адресу 127.x.x.x, то данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера, как только что принятые. Маршрут перемещения данных образует «петлю», поэтому этот адрес называется *адресом обратной петли (loopback)*.

Форма *группового IP-адреса - multicast* - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Групповой адрес не делится на номера сети и узла и обрабатывается маршрутизатором особым образом. Основное назначение групповых адресов - распространение информации по схеме «один ко многим». Основное назначение multicast-адресов - распространение информации по схеме «один-ко-многим». Хост, который хочет передавать одну и ту же информацию многим абонентам,

с помощью специального протокола IGMP (Internet Group Manageme Protocol) сообщает о создании в сети новой мультивещательной группы с определенным адресом. Маршрутизаторы, поддерживающие мультивещательность, распространяют информацию о создании новой группы во всех, подключенных к портам этого маршрутизатора. Хосты, которые хотят присоединиться к вновь создаваемой мультивещательной группе, сообщают об этом своим локальным маршрутизаторам и те передают эту информацию хосту, инициатору создания новой группы. Групповая адресация предназначена для экономичного распространения в Internet или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей.

Маска - число, которое служит для выделения частей IP-адреса, чтобы TCP/IP мог отличать номер сети от номера хоста. Используя маску подсети, TCP/IP-хосты могут связаться и определить, где находится хост назначения: в локальной или удаленной сети. Пример маски подсети: 255.255.255.0.

Биты IP-адреса, определяющие номер IP-сети, в маске подсети должны быть равны 1, а биты, определяющие номер узла, в маске подсети должны быть равны 0. Для стандартных классов сетей маски имеют следующие значения:

-класс А - 11111111. 00000000. 00000000. 00000000 (255.0.0.0); -
класс В - 11111111.11111111. 00000000. 00000000 (255.255.0.0);
-класс С-11111111.11111111.11111111. 00000000 (255.255.255.0).

Маски подсетей могут использоваться для маскирования тех частей адреса, которые согласно структуре класса, определяются как адреса сети. На практике разделение на подсети применяется в случае, когда конкретное сетевое адресное пространство разбивается дальше на отдельные подсети.

Подсети являются удобным средством структуризации сетей в рамках одной организации, когда все адресное пространство сети internet может быть разделено на непересекающиеся подпространства - "*подсети*", с каждой из которых можно работать как с обычной сетью TCP/IP. Таким образом единая IP-сеть организации может строиться как объединение подсетей. При этом организация должна получить один сетевой номер.

Выполнение работыЗадание 1. Изучить теоретические основы

IP-адресации

1. Сколько октетов в IP — адресе?
2. Сколько битов в октете?
3. Сколько бит в маске подсети?

Задание 2. Определить IP адрес вашего ПК

Узнайте собственный *IP адрес* компьютера и определите, к какому классу он относится.

Узнать свой собственный *IP адрес* вы можете, если запустите в ОС Windows XP на выполнение команду **Пуск – Программы – Стандартные – Командная Строка** и наберете в ней **ipconfig**.

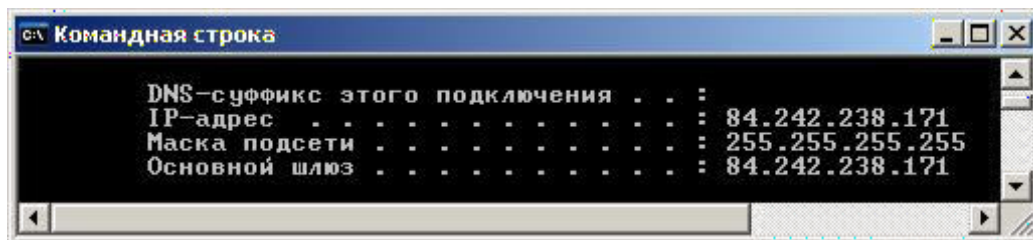


Рис. IP адрес вашего ПК в десятичной системе счисления

Задание 3. Переведите следующие двоичные числа в десятичные, а десятичные в двоичные.

Двоичное значение	Десятичное значение	Десятичное значение	Двоичное значение
10101100.00101000.00000000.00000000		127.1.1.1	
01011110.01110111.10011111.00000000		109.128.255.254	
10010001.0110000.10000000.00011001		131.107.2.89	
01111111.00000000.00000000.00000001		129.46.78.0	

Задание 4. Определение частей IP- адресов.

Заполнить таблицу об идентификации различных классов IP-адресов.

IP- адреса хостов	Класс адреса	Адрес сети	Адреса хостов	Широковещательный (broadcast) адрес	Маска подсети по умолчанию
216.14.55.137					
123.1.1.15					
150.127.221.244					
194.125.35.199					
175.12.239.244					

Задание 5. Дан IP- адрес 142.226.0.15

1. Чему равен двоичный эквивалент второго октета?

2. Какому классу принадлежит этот адрес?
3. Чему равен адрес сети, в которой находится хост с этим адресом?
4. Является ли этот адрес хоста допустимым в классической схеме адресации?

Задание 6

Найти адрес сети, минимальный IP, максимальный IP и число хостов по IP-адресу и маске сети:

IP-адрес: 192.168.215.89

Маска: 255.255.255.0

Задание 7

Найти маску сети, минимальный IP, максимальный IP по IP-адресу и адресу сети:

IP-адрес: 124.165.101.45

Сеть: 124.128.0.0

Задание 8

Найти минимальный IP, максимальный IP по адресу сети и маске:

Маска: 255.255.192.0

Сеть: 92.151.0.0

Задание 9. Определите, какие IP-адреса не могут быть назначены узлам. Объясните, почему такие IP-адреса не являются корректными.

1. 131.107.256.80
2. 222.222.255.222
3. 31.200.1.1
4. 126.1.0.0
5. 190.7.2.0
6. 127.1.1.1
7. 198.121.254.255
8. 255.255.255.255