

6.1. Что такое программное обеспечение?

Под программным обеспечением (Software) понимается совокупность программ, выполняемых вычислительной системой.

К программному обеспечению (ПО) относится также вся область деятельности по проектированию и разработке ПО:

- технология проектирования программ (например, нисходящее проектирование, структурное и объектно-ориентированное проектирование и др.);
- методы тестирования программ [[ссылка](#), [ссылка](#)];
- методы доказательства правильности программ;
- анализ качества работы программ;
- документирование программ;
- разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения, и многое другое.

Программное обеспечение — *неотъемлемая часть компьютерной системы*. Оно является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него ПО.

Сам по себе компьютер не обладает знаниями ни в одной области применения.
Все эти знания сосредоточены в выполняемых на компьютерах программах.

Программное обеспечение современных компьютеров включает миллионы программ — от игровых до научных.

6.2. Как классифицируется программное обеспечение?

В первом приближении все программы, работающие на компьютере, можно условно разделить на **три категории** (рис. 6.1):

1. прикладные программы, непосредственно обеспечивающие выполнение необходимых пользователям работ;
2. системные программы, выполняющие различные вспомогательные функции, например:
 - управление ресурсами компьютера;
 - создание копий используемой информации;
 - проверка работоспособности устройств компьютера;
 - выдача справочной информации о компьютере и др.;
3. инструментальные программные системы, облегчающие процесс создания новых программ для компьютера.



Рис. 6.1. Категории программного обеспечения

При построении классификации ПО нужно учитывать тот факт, что стремительное развитие вычислительной техники и расширение сферы приложения компьютеров резко ускорили процесс эволюции программного обеспечения.

Если раньше можно было по пальцам перечислить основные категории ПО — операционные системы, трансляторы, пакеты прикладных программ, то сейчас ситуация коренным образом изменилась.

Развитие ПО пошло как вглубь (появились новые подходы к построению операционных систем, языков программирования и т.д.), так и вширь (прикладные программы перестали быть прикладными и приобрели самостоятельную ценность).

Соотношение между требующимися программными продуктами и имеющимися на рынке меняется очень быстро. Даже классические программные продукты, такие, как операционные системы, непрерывно развиваются и наделяются интеллектуальными функциями, многие из которых ранее относились только к интеллектуальным возможностям человека.

Кроме того, появились нетрадиционные программы, классифицировать которые по устоявшимся критериям очень трудно, а то и просто невозможно, как, например, программа — *электронный собеседник*.

На сегодняшний день можно сказать, что **более или менее определённо сложились следующие группы программного обеспечения:**

- [операционные системы](#) и [оболочки](#);
- [системы программирования](#) (трансляторы, библиотеки подпрограмм, отладчики и т.д.);
- инструментальные системы;
- [интегрированные пакеты программ](#);
- динамические [электронные таблицы](#);
- системы машинной графики;
- системы управления базами данных ([СУБД](#));
- прикладное программное обеспечение.

Структура программного обеспечения показана на рис. 6.2. Разумеется, **эту классификацию нельзя считать исчерпывающей**, но она более или менее наглядно отражает направления совершенствования и развития программного обеспечения.

6.3. Какие программы называют прикладными?

Прикладная программа — это любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Например, там, где на компьютер возложена задача контроля за финансовой деятельностью какой-либо фирмы, прикладной будет программа подготовки платежных ведомостей.

Прикладные программы могут носить и общий характер, например, обеспечивать составление и печатание документов и т.п.

В противоположность этому, операционная система или инструментальное ПО не вносят прямого вклада в удовлетворение конечных потребностей пользователя.

Прикладные программы могут использоваться либо автономно, то есть решать поставленную задачу без помощи других программ, либо в составе программных комплексов или пакетов.

6.4. Какова роль и назначение системных программ?

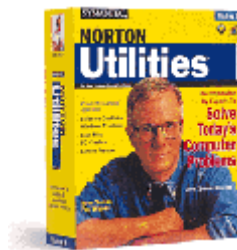
Системные программы выполняются вместе с прикладными и служат для управления ресурсами компьютера — центральным процессором, памятью, вводом-выводом.

Это программы общего пользования, которые **предназначены для всех пользователей компьютера**. Системное программное обеспечение разрабатывается так, чтобы компьютер мог эффективно выполнять прикладные программы.



Рис. 6.2. Структура программного обеспечения компьютера

Среди десятков тысяч системных программ особое место занимают **операционные системы**, которые обеспечивают управление ресурсами компьютера с целью их эффективного использования.



Важными классами системных программ являются также программы вспомогательного назначения — **утилиты** (лат. *utilitas* — польза). Они либо **расширяют и дополняют соответствующие возможности операционной системы**, либо **решают самостоятельные важные задачи**. Кратко опишем некоторые разновидности утилит:

- **программы контроля, тестирования и диагностики**, которые используются для проверки правильности функционирования устройств компьютера и для обнаружения неисправностей в процессе эксплуатации; указывают причину и место неисправности;
- **программы-драйверы**, которые расширяют возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д.; с помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся;
- **программы-упаковщики** (архиваторы), которые позволяют записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл;
- **антивирусные программы**, предназначенные для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусами;

Компьютерный вирус — это специально написанная небольшая по размерам программа, которая может "приписывать" себя к другим программам для выполнения каких-либо вредных действий — портит файлы, "засоряет" оперативную память и т.д.

- **программы оптимизации и контроля качества дискового пространства** ;
- **программы восстановления информации, форматирования, защиты данных** ;
- **коммуникационные программы**, организующие обмен информацией между компьютерами;
- **программы для управления памятью**, обеспечивающие более гибкое использование оперативной памяти;
- **программы для записи CD-ROM, CD-R** и многие другие.

Часть утилит входит в состав операционной системы, а другая часть функционирует независимо от нее, т.е. автономно.

6.5. Что такое операционная система?

Операционная система — это комплекс взаимосвязанных системных программ, назначение которого — организовать взаимодействие пользователя с компьютером и выполнение всех других программ.

Операционная система выполняет роль связующего звена между аппаратурой компьютера, с одной стороны, и выполняемыми программами, а также пользователем, с другой стороны.

Операционная система обычно хранится во внешней памяти компьютера — *на диске*. При включении компьютера она считывается с дисковой памяти и размещается в *ОЗУ*.

Этот процесс называется *загрузкой операционной системы*.

В функции операционной системы входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

Операционную систему можно назвать программным продолжением устройства управления компьютера. Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера.

В зависимости от количества одновременно обрабатываемых задач и числа пользователей, которых могут обслуживать ОС, различают четыре основных класса операционных систем:

1. **однопользовательские однозадачные**, которые поддерживают одну клавиатуру и могут работать только с одной (в данный момент) задачей;
2. **однопользовательские однозадачные с фоновой печатью**, которые позволяют помимо основной задачи запускать одну дополнительную задачу, ориентированную, как правило, на вывод информации на печать. Это ускоряет работу при выдаче больших объёмов информации на печать;
3. **однопользовательские многозадачные**, которые обеспечивают одному пользователю параллельную обработку нескольких задач. Например, к одному компьютеру можно подключить несколько принтеров, каждый из которых будет работать на "свою" задачу;
4. **многопользовательские многозадачные**, позволяющие на одном компьютере запускать несколько задач нескольким пользователям. Эти ОС очень сложны и требуют значительных машинных ресурсов.

В различных моделях компьютеров используют операционные системы с разной архитектурой и возможностями. Для их работы требуются разные ресурсы. Они предоставляют разную степень сервиса для программирования и работы с готовыми программами.

Операционная система для персонального компьютера, ориентированного на профессиональное применение, должна содержать следующие основные компоненты:

- программы управления вводом/выводом;
- программы, управляющие файловой системой и планирующие задания для компьютера;
- процессор командного языка, который принимает, анализирует и выполняет команды, адресованные операционной системе.

Каждая операционная система имеет свой командный язык, который позволяет пользователю выполнять те или иные действия:

- обращаться к каталогу;
- выполнять разметку внешних носителей;
- запускать программы;
- ... другие действия.

Анализ и исполнение команд пользователя, включая загрузку готовых программ из файлов в оперативную память и их запуск, осуществляет командный процессор операционной системы.

Для управления внешними устройствами компьютера используются специальные системные программы — драйверы. Драйверы стандартных устройств образуют в совокупности базовую систему ввода-вывода (BIOS), которая обычно заносится в постоянное ЗУ компьютера.

6.6. Что такое файловая система ОС?

Файл (англ. *file* — папка) — это именованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и др.

Файловая система — это средство для организации хранения файлов на каком-либо носителе.

Файлы физически реализуются как *участки памяти на внешних носителях* — магнитных дисках или CD-ROM. Каждый файл занимает некоторое количество блоков дисковой памяти. Обычная длина блока — 512 байт.

Обслуживает файлы специальный модуль операционной системы, называемый драйвером файловой системы. Каждый файл имеет имя, зарегистрированное в каталоге — оглавлении файлов.

Каталог (иногда называется *директорией* или *папкой*) доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нем файлы, переносить их содержимое на новое место и удалять.

Каталог может иметь собственное имя и храниться в другом каталоге наряду с обычными файлами: так образуются иерархические файловые структуры. Пример такой структуры — на рис. 6.3.

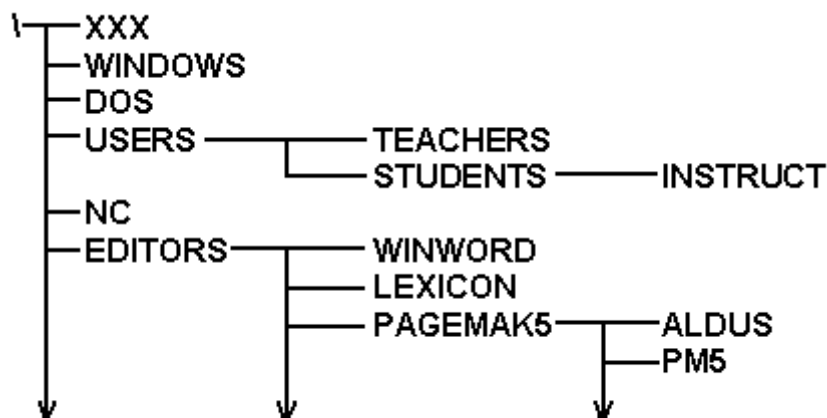


Рис. 6.3. Дерево каталогов на диске

Что происходит, когда пользователь подает операционной системе команду "*открыть файл ...*", в которой указано *имя файла* и *имя каталога*, в котором размещён этот файл?

Для выполнения этой команды драйвер файловой системы обращается к своему *справочнику*, выясняет, какие блоки диска соответствуют указанному файлу, а затем передает запрос на считывание этих блоков драйверу диска.

При выполнении команды "*сохранить файл*" драйвер файловой системы ищет на диске незанятые блоки, отмечает их, как распределённые для вновь созданного файла, и передаёт драйверу диска запрос на запись в эти блоки данных пользователя.

Драйвер файловой системы обеспечивает доступ к информации, записанной на магнитный диск, по имени файла и распределяет пространство на магнитном диске между файлами.

Для выполнения этих функций драйвер файловой системы хранит на диске не только информацию пользователя, но и свою собственную служебную информацию. **В служебных областях диска хранится список всех файлов и каталогов**, а также различные дополнительные справочные таблицы, служащие для повышения скорости работы драйвера файловой системы.

К файловой системе имеет доступ также и любая прикладная программа, для чего во всех языках программирования имеются специальные процедуры.

Понятие файла может быть обращено на любой источник или потребитель информации в машине, например, в качестве файла для программы могут выступать *принтер, дисплей, клавиатура* и др.

Структура файловой системы и структура хранения данных на внешних магнитных носителях определяет удобство работы пользователя, скорость доступа к файлам и т.д.

6.7. Какова структура операционной системы MS DOS?

Операционная система MS DOS (Microsoft Disk Operating System) — самая распространенная ОС на 16-разрядных персональных компьютерах. Она состоит из следующих основных модулей (рис. 6.4):

- базовая система ввода/вывода ([BIOS](#));
- блок начальной загрузки (Boot Record);
- модуль расширения базовой системы ввода/вывода (IO.SYS);
- модуль обработки прерываний (MSDOS.SYS);
- командный процессор (COMMAND.COM);
- [утилиты](#) MS DOS.

Каждый из указанных модулей выполняет определенную часть функций, возложенных на ОС. Места постоянного размещения этих модулей различны. Так, *базовая система ввода/вывода находится в постоянном запоминающем устройстве ([ПЗУ](#))*, а не на дисках, как все остальные модули.

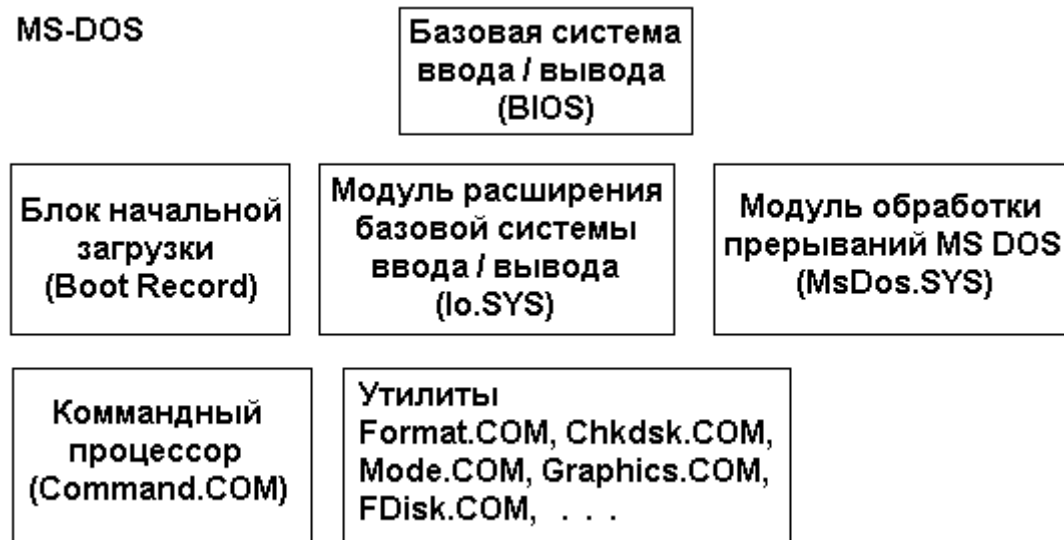


Рис. 6.4. Состав операционной системы MS-DOS

Базовая система ввода/вывода (BIOS) выполняет наиболее простые и универсальные услуги операционной системы, связанные с осуществлением *ввода-вывода*. В функции BIOS входит также *автоматическое тестирование основных аппаратных компонентов* (оперативной памяти и др.) при включении машины и *вызов блока начальной загрузки DOS*.

Блок начальной загрузки (или просто *загрузчик*) — это очень короткая программа, единственная функция которой заключается в считывании с диска в оперативную память двух других частей DOS — модуля расширения базовой системы ввода/вывода и модуля обработки прерываний.

Модуль расширения базовой системы ввода/вывода дает возможность использования *дополнительных драйверов*, обслуживающих новые внешние устройства, а

также драйверов для нестандартного обслуживания внешних устройств.

Модуль обработки прерываний реализует основные высокоуровневые услуги DOS, поэтому его и называют основным.

Командный процессор DOS обрабатывает команды, вводимые пользователем.

Утилиты DOS — это программы, поставляемые вместе с операционной системой в виде отдельных файлов. Они выполняют действия обслуживающего характера, например, разметку дискет, проверку дисков и т.д.

6.8. Что такое программы-оболочки?

Оболочки — это программы, созданные для упрощения работы со сложными программными системами, такими, например, как DOS. Они преобразуют неудобный командный пользовательский интерфейс в дружелюбный графический интерфейс или интерфейс типа "меню". Оболочки предоставляют пользователю удобный доступ к файлам и обширные сервисные услуги.

Самая популярная у пользователей IBM-совместимого ПК оболочка — пакет программ **Norton Commander**. Он обеспечивает:

- создание, копирование, пересылку, переименование, удаление, поиск файлов, а также изменение их атрибутов;
- отображение дерева каталогов и характеристик входящих в них файлов в форме, удобной для восприятия человека;
- создание, обновление и распаковку архивов (групп сжатых файлов);
- просмотр текстовых файлов;
- редактирование текстовых файлов;
- выполнение из её среды практически всех команд DOS;
- запуск программ;
- выдачу информации о ресурсах компьютера;
- создание и удаление каталогов;
- поддержку межкомпьютерной связи;
- поддержку электронной почты через модем.

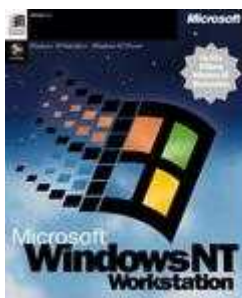
В начале 90-х годов во всем мире огромную популярность приобрела графическая оболочка **MS-Windows 3.x**, преимущество которой состоит в том, что она облегчает использование компьютера, и её графический интерфейс вместо набора сложных команд с клавиатуры позволяет выбирать их мышью из меню практически мгновенно. Операционная среда Windows, работающая совместно с операционной системой DOS, реализует все свойства, необходимые для производительной работы пользователя, в том числе — многозадачный режим.

Оболочка **Norton Navigator** — это набор мощных программ для управления файлами, расширяющий возможности Windows. Позволяет экономить время практически на всех операциях: поиск файлов, копирование и перемещение файлов, открытие каталогов.

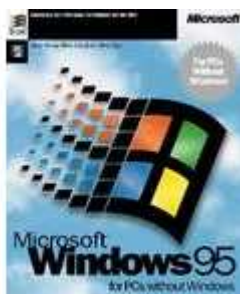
6.9. Что собой представляют операционные системы *Windows*, *Unix*, *Linux* ?

Операционные системы *Windows*

В настоящее время большинство компьютеров в мире работают под управлением той или иной версии операционной среды *Windows* фирмы *Microsoft*. Охарактеризуем наиболее распространенные версии.



Windows NT (*NT* — англ. *New Technology*) — это операционная система, а не просто графическая оболочка. Она использует все возможности новейших моделей персональных компьютеров и работает без *DOS*. *Windows NT* — 32-разрядная ОС со встроенной сетевой поддержкой и развитыми многопользовательскими средствами. Она предоставляет пользователям истинную многозадачность, многопроцессорную поддержку, секретность, защиту данных и многое другое. Эта операционная система очень удобна для пользователей, работающих в рамках локальной сети, для коллективных пользователей, особенно для групп, работающих над большими проектами и обменивающихся данными.



Windows 95 представляет собой универсальную высокопроизводительную многозадачную и многопоточную 32-разрядную ОС нового поколения с графическим интерфейсом и расширенными сетевыми возможностями. *Windows 95* — интегрированная среда, обеспечивающая эффективный обмен информацией между отдельными программами и предоставляющая пользователю широкие возможности работы с мультимедиа, обработки текстовой, графической, звуковой и видеoinформации. Интегрированность подразумевает также *совместное использование ресурсов компьютера всеми программами*.

Эта операционная система *обеспечивает работу пользователя в сети*, предоставляя встроенные средства поддержки для обмена файлами и меры по их защите, возможность совместного использования принтеров, факсов и других общих ресурсов. *Windows 95* позволяет отправлять сообщения электронной почтой, факсимильной связью, поддерживает удаленный доступ. Применяемый в *Windows 95* защищенный режим не позволяет прикладной программе в случае сбоя нарушить работоспособность системы,

надежно предохраняет приложения от случайного вмешательства одного процесса в другой, обеспечивает определённую устойчивость к вирусам.

Пользовательский интерфейс *Windows 95* прост и удобен. В отличие от оболочки *Windows 3* эта операционная система не нуждается в установке на компьютере операционной системы DOS. Она предназначена для установки на настольных ПК и компьютерах блокнотного типа с процессором *486* или *Pentium*. Рекомендуемый размер оперативной памяти 32—128 Мбайт. После включения компьютера и выполнения тестовых программ *BIOS* операционная система *Windows 95* автоматически загружается с жесткого диска. После загрузки и инициализации системы на экране появляется *рабочий стол*, на котором размещены различные *графические объекты*. Пользовательский интерфейс спроектирован так, чтобы максимально облегчить усвоение этой операционной системы новичками и создать комфортные условия для пользователя.

Windows 98 отличается от *Windows 95* тем, что в ней операционная система объединена с браузером ***Internet Explorer*** посредством интерфейса, выполненного в виде *Web*-браузера и оснащенного кнопками "Назад" и "Вперед" для перехода на предыдущую и последующую *Web*-страницы. Кроме этого, в ней улучшена совместимость с новыми аппаратными средствами компьютера, она одинаково удобна как для использования на настольных, так и на портативных компьютерах.

Windows 2000 Professional — операционная система нового поколения для делового использования на самых разнообразных компьютерах — от портативных до серверов. Эта ОС является наилучшей для ведения коммерческой деятельности в Интернете. Она объединяет присущую *Windows 98* простоту использования в Интернете, на работе, в пути с присущими *Windows NT* надежностью, экономичностью и безопасностью.

Windows CE 3.0 — операционная система для **мобильных вычислительных устройств**, таких, как карманные компьютеры, цифровые информационные пейджеры, сотовые телефоны, мультимедийные и развлекательные приставки, включая DVD проигрыватели и устройства целевого доступа в Интернет.

Операционная система *Windows CE* — 32-разрядная, многозадачная, многопоточная операционная система, имеющая открытую архитектуру, разрешающую использование множеств устройств. *Windows CE* позволяет устройствам различных категорий "говорить" и обмениваться информацией друг с другом, связываться с корпоративными сетями и с Интернет, пользоваться электронной почтой.

Windows CE компактна, но высоко производительна. Это мобильная система, функционирующая с микропроцессорами различных марок и изготовителей. Для нее есть программы *Word* и *Excel*, которые совместимы с их настольными аналогами. Имеет интегрированную систему управления питанием.

Операционная система *Unix*

Операционная система *Unix* была создана в *Bell Telephone Laboratories*. С краткой историей ее создания можно ознакомиться [здесь](#). *Unix* — **многозадачная операционная система, способная обеспечить одновременную работу очень большого количества пользователей**. Ядро ОС *Unix* написано на языке высокого уровня *C* и имеет только около 10 процентов кода на ассемблере. Это позволяет за считанные месяцы переносить ОС *Unix* на другие аппаратные платформы и достаточно легко вносить в нее серьезные изменения и дополнения. ***UNIX* является первой действительно переносимой операционной системой. В многочисленные существующие версии *UNIX* постоянно вносятся изменения.** С одной стороны, это расширяет возможности системы, делает ее

мощнее и надежнее, с другой — ведет к появлению различий между существующими версиями. В связи с этим **возникает необходимость стандартизации различных свойств системы**. Наличие стандартов облегчает переносимость приложений между различными версиями *UNIX* и защищает как пользователей, так и производителей программного обеспечения. Поэтому в 80-х годах разработан ряд стандартов, оказывающих влияние на развитие *UNIX*. **Сейчас существуют десятки операционных систем, которые можно объединить под общим названием *UNIX***. В основном, это коммерческие версии, выпущенные производителями аппаратных платформ для компьютеров своего производства. Причины популярности *UNIX*:

- **Код системы написан на языке высокого уровня *C*, что сделало ее простой для понимания, изменения и переноса на другие платформы.** Можно смело сказать, что *UNIX* является одной из наиболее открытых систем.
- ***UNIX* — многозадачная многопользовательская система.** Один мощный сервер может обслуживать запросы большого количества пользователей. При этом необходимо администрирование только одной системы. Кроме того, система способна выполнять большое количество различных функций, в частности, работать, как вычислительный сервер, как сервер базы данных, как сетевой сервер, поддерживающий важнейшие сервисы сети и т.д.
- **Наличие стандартов.** Несмотря на разнообразие версий *UNIX*, основой всего семейства являются принципиально одинаковая архитектура и ряд стандартных интерфейсов. Для администратора переход на другую версию системы не составит большого труда, а для пользователей он может и вовсе оказаться незаметным.
- **Простой, но мощный модульный пользовательский интерфейс.** Имея в своем распоряжении набор утилит, каждая из которых решает узкую специализированную задачу, можно конструировать из них сложные комплексы.
- **Использование единой, легко обслуживаемой иерархической файловой системы.** Файловая система *UNIX* — это не только доступ к данным, хранящимся на диске. Через унифицированный интерфейс файловой системы осуществляется доступ к терминалам, принтерам, сети и т.п.
- **Очень большое количество приложений, в том числе свободно распространяемых,** начиная от простейших текстовых редакторов и заканчивая мощными системами управления базами данных.

(По материалам пособия "ОС UNIX", <http://base1.spiiras.nw.ru/univer/unix>)

Операционная система *Linux*

Начало созданию системы *Linux* положено в 1991 г. финским студентом **Линусом Торвальдсом** (*Linus Torvalds*). В сентябре 1991 года он распространил по e-mail первый прототип своей операционной системы, и призвал откликнуться на его работу всех, кому она нравится или нет. С этого момента многие программисты стали поддерживать *Linux*, добавляя драйверы устройств, разрабатывая разные продвинутые приложения и др. Атмосфера работы энтузиастов над полезным проектом, а также свободное распространение и использование исходных текстов стали основой феномена *Linux*. В настоящее время *Linux* — **очень мощная система**, но самое замечательное то, что **она бесплатная** (free).



Линус Торвальдс с символом *Linux* — "пингином"

Линус Торвалдс разработал не саму операционную систему, а только ее ядро, подключив уже имеющиеся компоненты. Сторонние компании, увидев хорошие перспективы для развития своего бизнеса, довольно скоро стали насыщать ОС утилитами и прикладным ПО. **Недостаток такого подхода — отсутствие унифицированной и продуманной процедуры установки системы**, и это до сих пор является одним из главных сдерживающих факторов для более широкого распространения Linux.

Феномен *Linux* вызвал к жизни разговоры о том, что **родилась новая философия программирования**, принципиально отличающаяся от того, что было раньше. Традиционные стадии жизненного цикла программного продукта таковы: анализ требований, разработка спецификаций, проектирование, макетирование, написание исходного текста, отладка, документирование, тестирование и сопровождение. Главное, что отличает этот подход, — централизация управления разными стадиями и преимущественно "нисходящая" разработка (то есть постоянная детализация). Однако *Linux* создавалась по-иному. **Готовый работающий макет постоянно совершенствовался и развивался децентрализованной группой энтузиастов, действия которых лишь слегка координировались.** Налицо анархичный характер и "восходящая" разработка: сборка все более крупных блоков из ранее созданных мелких. Здесь можно отметить и другое. При традиционной разработке в основу кладется проектирование и написание текстов, при разработке по методу *Linux* — макетирование, отладка и тестирование. Первые два этапа распараллелить сложно, а с отладкой и тестированием дело обстоит полегче. Иными словами, **разработка по методу *Linux* — это метод проб и ошибок, построенный на интенсивном тестировании.** На любом этапе система должна работать, даже если это мини-версия того, к чему стремится разработчик. Естественный отбор оставляет только жизнеспособное. О том, что такое программирование — наука, искусство или ремесло, — спорят уже давно. И если в основе традиционной разработки ПО лежит прежде всего ремесло, то при разработке методом компьютерного дарвинизма — несомненно искусство.

Нетрудно заметить, что "восходящая" разработка характеризует так называемое исследовательское программирование, когда система строится вокруг ключевых компонентов и программ, которые создаются на ранних стадиях проекта, а затем постоянно модифицируются. **Отсутствие четкого плана, минимальное управление проектом, большое число сторонних территориально удаленных разработчиков, свободный обмен идеями и кодами — все это атрибуты нового программирования.** Об особенностях исследовательского программирования написано немало статей. Так, швейцарские профессора А.Киральф, К.Чен и Й.Нивергельт выделили следующие важные моменты:

- разработчик ясно представляет направление поиска, но не знает заранее, как далеко он сможет продвинуться к цели;
- нет возможности предвидеть объем ресурсов для достижения того или иного результата;
- разработка не поддается детальному планированию, она ведется методом проб и ошибок;
- такие работы связаны с конкретными исполнителями и отражают их личностные качества.

6.10. Что такое транслятор, компилятор, интерпретатор?

Транслятор (англ. *translator* — переводчик) — это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются.

Компилятор (англ. *compiler* — составитель, собиратель) читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор (англ. *interpreter* — истолкователь, устный переводчик) переводит и выполняет программу *строка за строкой*.

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново *переводиться* на машинный язык при каждом очередном запуске программы.

Откомпилированные программы работают быстрее, но *интерпретируемые* проще исправлять и изменять.

Каждый конкретный язык ориентирован либо на компиляцию, либо на интерпретацию — в зависимости от того, для каких целей он создавался. Например, Паскаль обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью *компилятора*.

С другой стороны, Бейсик создавался как язык для начинающих программистов, для которых построчное выполнение программы имеет неоспоримые преимущества.

Иногда для одного языка имеется *и компилятор, и интерпретатор*. В этом случае для разработки и тестирования программы можно воспользоваться интерпретатором, а затем откомпилировать отлаженную программу, чтобы повысить скорость ее выполнения.

6.11. Что такое системы программирования?

Система программирования — это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям **мощные и удобные средства разработки программ**. В них входят:

- компилятор или интерпретатор;
- интегрированная среда разработки;

- средства создания и редактирования текстов программ;
- обширные [библиотеки стандартных программ](#) и функций;
- [отладочные программы](#), т.е. программы, помогающие находить и устранять ошибки в программе;
- "дружественная" к пользователю диалоговая среда;
- многооконный режим работы;
- мощные графические библиотеки; [утилиты](#) для работы с библиотеками
- встроенный [ассемблер](#);
- встроенная справочная служба;
- другие специфические особенности.

Популярные системы программирования — *Turbo Basic*, *Quick Basic*, *Turbo Pascal*, *Turbo C*.

В последнее время получили распространение системы программирования, ориентированные на создание *Windows-приложений*:



Borland Delphi 3.0

- пакет **Borland Delphi** (Дельфи) — блестящий наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.
- пакет **Microsoft Visual Basic** — удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Содержит инструментарий для создания *диаграмм* и *презентаций*.
- пакет **Borland C++** — одно из самых распространённых средств для разработки DOS и Windows приложений.

Ниже для иллюстрации приведены на языках Бейсик, Паскаль и Си программы решения одной и той же простой задачи — вычисления суммы S элементов одномерного массива $A=(a_1, a_2, \dots, a_n)$.

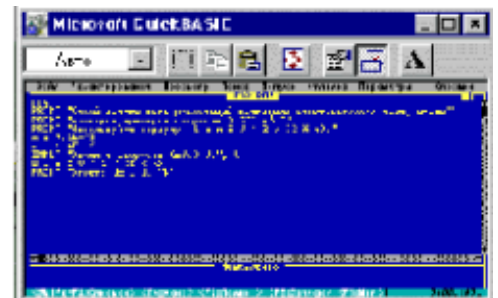
Язык Бейсик (BASIC — Beginners All-purpose Symbolic Instruction Code — универсальный символьный код для начинающих) был создан в 1965 г. **Дж. Кемени и Т.Курцем** как язык, облегчающий написание простых программ. Сейчас все чаще пишут **Basic** вместо **BASIC**, придавая другую трактовку названию: Basic — основной, базовый.

Существует много различных версий Бейсика — от очень простых до усовершенствованных, содержащих множество дополнительных языковых конструкций. Наибольшее распространение имеют следующие версии: **QuickBasic** (QBasic) 4.5 для DOS и **Visual Basic** 3.0-6.0 для Windows.

QuickBasic 4.5 фирмы Microsoft — это очень простой, но в то же время эффективный язык, унаследовавший от раннего Бейсика все его достоинства, но избавившийся от всех его недостатков и впитавший целый ряд передовых идей начала 90-х годов. Еще QuickBasic очень привлекателен своей средой программирования. Одна из удач — это использование своеобразного режима работы. Для быстрой работы в среде используется режим **интерпретатора**, а для окончательного перевода отлаженных программ на машинный язык используется **компилятор**. Из QBasic компилятор исключен. QBasic входит в минимальный комплект поставки программного обеспечения компьютера. Бейсик очень популярный язык программирования.

Программа на Бейсике

```
INPUT "N = "; N : DIM A(N)
FOR I = 1 TO N
    PRINT "A("; I; ") = ";
    INPUT A(I)
NEXT I
S = 0
FOR I = 1 TO N
    S = S + A(I)
NEXT I
PRINT "Сумма ="; S
END
```



Окно среды программирования Quick Basic

Программа на Паскале

```
Program Summa;  
Type Mas = Array [1 ..  
100] of Real;  
Var A : Mas;  
    i, n: Integer;  
    S : Real;  
BEGIN  
    Write('n = '); ReadLn(n);  
    For i := 1 to n do  
        begin  
            Write('A[', i, '] = ');  
            ReadLn(A[i]);  
        end;  
  
    S := 0;  
    For i := 1 to n do  
        S := S + A[i];  
    WriteLn('S = ', S:8:2);  
END.
```

Язык Паскаль был разработан в 1970 г. **Никласом Виртом** как язык обучения студентов программированию. Паскаль вырабатывает навыки соблюдения хорошего строгого стиля программирования, упрощающего разработку сложных программ.

Основные привлекательные черты Паскаля — **логичность, поддержка концепций структурного и процедурного программирования, работа с динамической памятью, возможность создания своих типов данных.** В Паскале программист должен всегда явно указывать, с какими конкретными переменными он желает работать и каковы типы этих переменных. Строгая типизация данных позволяет резко снизить количество ошибок, появляющихся в программе вследствие невнимательности или опечаток.

В своем первоначальном виде Паскаль имел довольно ограниченные возможности, но расширенный вариант этого языка — **Turbo Pascal**, является очень мощным языком программирования. Интегрированная оболочка Turbo Pascal, разработанная фирмой **Borland** (ныне Inprise), включающая в себя редактор, компилятор, компоновщик и отладчик, вместе с интерактивной справочной системой сделали разработку программ на Паскале делом простым и приятным.

Язык Си разработан **Деннисом Ритчи** в 1972 г. как язык, пригодный для программирования новой операционной системы UNIX.

Операционные системы ради повышения скорости работы традиционно писались на языке низкого уровня — ассемблере, но язык Си настолько хорошо зарекомендовал себя, что на нем было написано более 90% всего кода ОС UNIX. Язык СИ обрел популярность как так называемый язык среднего уровня, в котором удобство, краткость и мобильность языков высокого уровня сочетаются с возможностью непосредственного доступа к аппаратуре компьютера, что обычно достигаются только при программировании на языке Ассемблера.

Си не очень прост в изучении и требует тщательности в программировании, но позволяет создавать сложные и весьма эффективные программы.

Программа на СИ

```
# include <stdio.h>  
# include <conio.h>  
main()  
{  
    float a[100], s; int i, n;  
    clrscr(); printf("n=");  
    scanf("%i", &n);  
    for (i = 1; i <= n; i++) {  
        printf("a[%i]=", i);  
        scanf("%f", &a[i]);  
    }  
    s=0;  
    for (i = 1; i <= n; i++)  
        s = s + a[i];  
    printf("s = % f\n", s);  
    return 0;  
}
```

6.12. Для чего нужны инструментальные программы?

Инструментальные программные средства — это программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ.

По своему назначению они близки системам программирования. К инструментальным программам, например, относятся:

- редакторы;
- средства компоновки программ;
- отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- вспомогательные программы, реализующие часто используемые системные действия;
- графические пакеты программ и т.п.

Инструментальные программные средства могут оказать помощь на всех стадиях разработки ПО.

6.13. Что такое текстовый редактор?

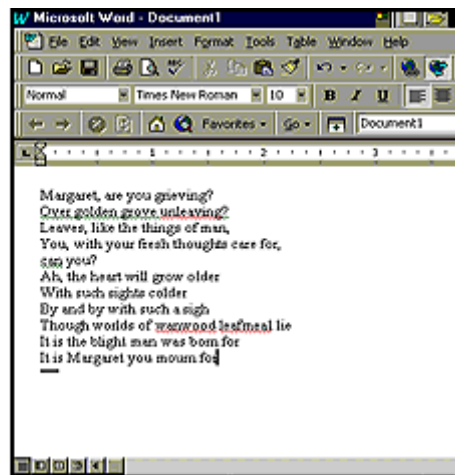
Текстовый редактор — это программа, используемая специально для ввода и редактирования текстовых данных.

Этими данными могут быть программа или какой-либо документ или же книга. Редактируемый текст выводится на экран, и пользователь может в диалоговом режиме вносить в него свои изменения.

Текстовые редакторы могут обеспечивать выполнение разнообразных функций, а именно:

- редактирование строк текста;
- возможность использования различных шрифтов символов;
- копирование и перенос части текста с одного места на другое или из одного документа в другой;
- контекстный поиск и замена частей текста;
- задание произвольных межстрочных промежутков;
- автоматический перенос слов на новую строку;
- автоматическая нумерацию страниц;
- обработка и нумерация сносок;
- выравнивание краев абзаца;
- создание таблиц и построение диаграмм;
- проверка правописания слов и подбор синонимов;
- построение оглавлений и предметных указателей;
- распечатка подготовленного текста на принтере в нужном числе экземпляров и т.п.

Возможности текстовых редакторов различны — от программ, предназначенных для подготовки небольших документов простой структуры, до программ для набора, оформления и полной подготовки к типографскому изданию книг и журналов (издательские системы).



Окно редактора Microsoft Word

Наиболее известный текстовый редактор — **Microsoft Word**.

Полнофункциональные издательские системы — **Microsoft Publisher**, **Corel Ventura** и **Adobe PageMaker**. Издательские системы незаменимы для компьютерной верстки и графики. Значительно облегчают работу с многостраничными документами, имеют возможности автоматической разбивки текста на страницы, расстановки номеров страниц, создания заголовков и т.д. Создание макетов любых изданий — от рекламных листовок до многостраничных книг и журналов — становится очень простым, даже для новичков.

6.14. Что такое графический редактор?

Графический редактор — это программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений. Предоставляет возможности рисования линий, кривых, раскраски областей экрана, создания надписей различными шрифтами и т.д.



Рисунок, полученный с помощью Corel Draw!

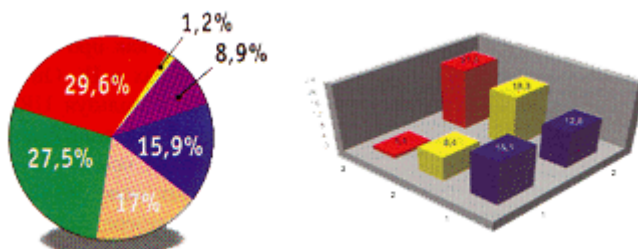
[Увеличить изображение](#)

Большинство редакторов позволяют обрабатывать изображения, полученные с помощью *сканеров*, а также выводить картинки в таком виде, чтобы они могли быть включены в документ, подготовленный с помощью текстового редактора.

Некоторые редакторы позволяют получать изображения трёхмерных объектов, их сечений, разворотов, каркасных моделей и т.п.

Пользуется известностью **Corel DRAW!** — мощный графический редактор с функциями создания публикаций, снабжённый инструментами для редактирования графики и трёхмерного моделирования.

6.15. Каковы возможности систем деловой и научной графики?

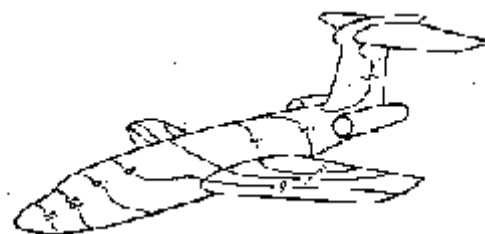


Круговые и столбиковые диаграммы

Системы деловой графики дают возможность выводить на экран различные виды графиков и диаграмм: >гистограммы; круговые и секторные диаграммы и т.д.

Эти системы позволяют наглядно представлять на экране различные данные и зависимости.

Системы научной и инженерной графики позволяют в цвете и в заданном масштабе отображать на экране следующие объекты:

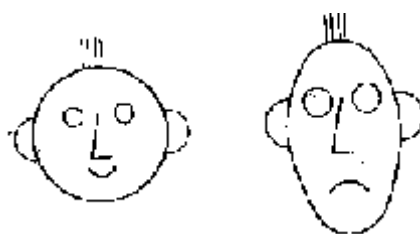


Чертеж с нанесенными изолиниями

- графики двумерных и трехмерных функций, заданных в табличном или аналитическом виде;
- системы изолиний, в том числе, и нанесённые на поверхность объекта;
- сечения, проекции, карты и т.д.

Для построения легко воспринимаемых реалистических изображений трёхмерных объектов системы инженерной графики позволяют удалять линии, не видимые наблюдателю.

Существуют **остроумные способы визуализации** наиболее простых **многомерных объектов** — множеств точек. Один из них носит название "лица Чернова" (Чернов — современный американский математик).



"Лица Чернова"

Этим способом можно отображать 10-20-мерные множества. Суть способа такова: каждому из измерений сопоставляется один из параметров схематически изображённого человеческого лица, например, первое измерение дает отношение высоты лица к ширине, второе — размер носа, третье — расстояние между глазами и т.д.

Таким образом, каждой точке исходного множества будет сопоставлено лицо. Рассматривая эти лица, можно отобрать похожие между собой или же выделить абсолютно непохожие и тем самым произвести некую классификацию исходного множества.

6.16. Что такое табличный процессор?

Табличный процессор — это комплекс взаимосвязанных программ, предназначенный для обработки электронных таблиц.

Электронная таблица — это компьютерный эквивалент обычной таблицы, состоящей из строк и граф, на пересечении которых располагаются клетки, в которых содержится числовая информация, формулы или текст.

Значение в числовой клетке таблицы может быть либо записано, либо рассчитано по соответствующей формуле; в формуле могут присутствовать обращения к другим клеткам.

	A	B	C	D	E	F
1	Vertrod #1					
2	Feil Pouch 3630					Feil Pouch 3630
3	Dwell #2	Dwell #3	Dwell #4	Dwell #5	Dwell #6	Dwell #7
4	1/30/95	1/30/95	1/30/95	1/30/95	1/31/95	1/31/95
5	17.85	22.9	23.65	24.8	19.15	23.1
6	19	26.4	22.6	27.5	18.6	21.1
7	13.15	25.45	26.9	25.9	19.01	24.1
8	19.05	24.55	27.65	28.2	19.25	24.1
9	14.1	25.4	29.1	21.45	17.65	22.1
10	27.6	21.45	27.4	23.65	22.02	21.1
11	36.5	21.2	23.3	20.65	21.9	26.1
12	24.25	26.3	24.8	20.5	19.75	23.1
13	15.05	23.55	23.6	23.9	14.65	22.1
14	24.9	25	24.45	25.1	17.9	22.1
15	26.55	27.35	20.25	24.1	15.85	21.1
16	30.7	22.45	24.55	22.95	15.7	21.1
17	31.3	26.35	25.6	22.3	13.25	24.1

Электронная таблица
Microsoft Excel

Каждый раз при изменении значения в клетке таблицы в результате записи в нее нового значения с клавиатуры *пересчитываются* также значения во всех тех клетках, в которых стоят величины, зависящие от данной клетки.

Графам и строкам можно присваивать наименования. Экран монитора трактуется как окно, через которое можно рассматривать таблицу целиком или по частям.

Табличные процессоры представляют собой удобное средство для проведения бухгалтерских и статистических расчетов. В каждом пакете имеются сотни встроенных математических функций и алгоритмов статистической обработки данных. Кроме того, имеются мощные средства для связи таблиц между собой, создания и редактирования электронных баз данных.

Специальные средства позволяют автоматически получать и распечатывать настраиваемые отчеты с использованием десятков различных типов таблиц, графиков, диаграмм, снабжать их комментариями и графическими иллюстрациями.

Табличные процессоры имеют встроенную справочную систему, предоставляющую пользователю информацию по конкретным командам меню и другие справочные данные. Многомерные таблицы позволяют быстро делать выборки в базе данных по любому критерию.

Самые популярные табличные процессоры — *Microsoft Excel (Эксель)* и *Lotus 1—2—3*.

В *Microsoft Excel* автоматизированы многие рутинные операции, специальные шаблоны помогают создавать отчёты, импортировать данные и многое другое.

Lotus 1—2—3 — профессиональный процессор электронных таблиц. Широкие графические возможности и удобный интерфейс пакета позволяют быстро ориентироваться в нём. С его помощью можно создать любой финансовый документ, отчёт для бухгалтерии, составить бюджет, а затем разместить все эти документы в базах данных.

6.17. Что такое системы управления базами данных?

База данных — это один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

В базе данных предприятия, например, может храниться:

- вся информация о штатном расписании, о рабочих и служащих предприятия;
- сведения о материальных ценностях;
- данные о поступлении сырья и комплектующих;
- сведения о запасах на складах;
- данные о выпуске готовой продукции;
- приказы и распоряжения дирекции и т.п.

Даже небольшие изменения какой-либо информации могут приводить к значительным изменениям в разных других местах.

Пример. Издание приказа о повышении в должности одного работника приводит к изменениям не только в личном деле работника, но и к изменениям в списках подразделения, в котором он работает, в ведомостях на зарплату, в графике отпусков и т.п.

Базы данных используются под управлением *систем управления базами данных (СУБД)*.

Система управления базами данных (СУБД) — это система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.



Окно СУБД Access

Системы управления базами данных позволяют *объединять* большие объемы информации и *обрабатывать* их, *сортировать*, *делать выборки* по определённым критериям и т.п.

Современные СУБД дают возможность включать в них не только *текстовую* и *графическую* информацию, но и звуковые фрагменты и даже видеоклипы.

Простота использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями.

СУБД обеспечивают *правильность*, *полноту* и *непротиворечивость* данных, а также *удобный доступ* к ним.

Популярные СУБД — ***FoxPro, Access for Windows, Paradox.***

Для менее сложных применений вместо СУБД используются **информационно-поисковые системы (ИПС)**, которые выполняют следующие функции:

- *хранение* большого объема информации;
- быстрый *поиск* требуемой информации;
- *добавление, удаление* и *изменение* хранимой информации;
- *вывод* ее в удобном для человека виде.

6.18. Что такое библиотеки стандартных подпрограмм?

Библиотека стандартных подпрограмм — это совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих определенным единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм и т.п.

Стандартные подпрограммы имеют единую форму обращения, что обеспечивает простоту и удобство настройки параметров подпрограммы на решение конкретной задачи.

В качестве примера можно привести библиотеку стандартных подпрограмм по численным математическим методам решения уравнений, вычисления интегралов, нахождения экстремумов и т.п.

6.19. Что такое пакеты прикладных программ?

Пакеты прикладных программ (ППП) — это специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

В зависимости от характера решаемых задач различают следующие разновидности ППП:

- пакеты для решения типовых инженерных, планово-экономических, общенаучных задач;
- пакеты системных программ;
- пакеты для обеспечения систем автоматизированного проектирования и систем автоматизации научных исследований;
- пакеты педагогических программных средств и другие.

Чтобы пользователь мог применить ППП для решения конкретной задачи, пакет должен обладать средствами настройки (иногда путём введения некоторых дополнений).

Каждый ППП обладает обычно рядом возможностей по методам обработки данных и формам их представления, полноте диагностики, что дает возможность пользователю выбрать подходящий для конкретных условий вариант.

ППП обеспечивают значительное снижение требований к уровню профессиональной подготовки пользователей в области программирования, вплоть до возможности эксплуатации пакета без программиста.

Часто пакеты прикладных программ располагают базами данных для хранения данных и передачи их прикладным программам.

6.20. Что такое интегрированные пакеты программ?

Интегрированные пакеты представляют собой набор нескольких программных продуктов, объединенных в единый удобный инструмент. Наиболее развитые из них включают в себя текстовый редактор, органайзер, электронную таблицу, СУБД, средства поддержки электронной почты, программу создания презентационной графики.

Результаты, полученные отдельными подпрограммами, могут быть объединены в окончательный документ, содержащий табличный, графический и текстовый материал.

Интегрированные пакеты, как правило, содержат некоторое ядро, обеспечивающее возможность тесного взаимодействия между составляющими.

Пример: **интегрированный пакет для написания книг, содержащих иллюстрации.** Он содержит:

- текстовый редактор;
- орфографический корректор на 80000 слов (программу обнаружения орфографических ошибок);
- программу слияния текстов;
- программу формирования оглавлений и составления указателей;
- автоматический поиск и замену слов и фраз;
- средства телекоммуникации;
- электронную таблицу;
- систему управления базами данных;
- модули графического оформления;
- графический редактор;
- возможность печати сотнями разных шрифтов и т.д.



Наиболее известные интегрированные пакеты:

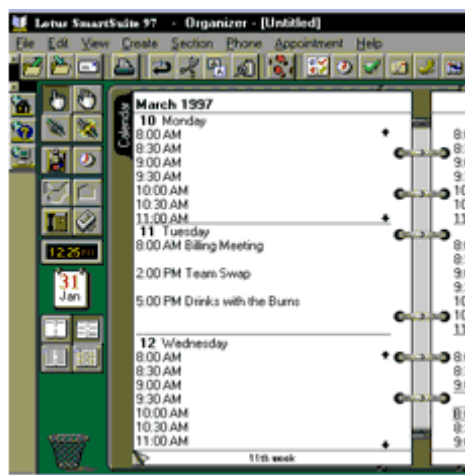
Microsoft Office. В этот мощный профессиональный пакет вошли такие необходимые программы, как текстовый редактор *WinWord*, электронная таблица *Excel*, программа создания презентаций *PowerPoint*, СУБД *Access*, средство поддержки электронной почты *Mail*. Мало того, все части этого пакета составляют единое целое, и даже внешне все программы выглядят единообразно, что облегчает как их освоение, так и ежедневное использование.

Microsoft Works — это очень простой и удобный пакет, объединяющий в себе текстовый редактор, электронные таблицы и базы данных, а также телекоммуникационные средства для соединения с другими компьютерами по

телефонным линиям. Пакет ориентирован на людей, не имеющих времени осваивать сложные продукты, на начинающих пользователей, а также на домашних пользователей.

6.21. Что такое органайзеры?

Органайзеры — это программы — электронные секретари.



Lotus Organizer

Они позволяют эффективно распорядиться рабочим временем, финансовыми средствами и т.п. Обладают возможностью автоматизации регулярных действий, составления персональных и групповых расписаний, планирования встреч, ведения записной книжки. В их состав традиционно входят календарь, часы, калькулятор и т.п.

Lotus Organizer — блокнот, разбитый по секциям: календарь, список дел, адресная и телефонная книга, планировщик, записная книжка, список памятных дат.

Microsoft Project позволяет спланировать проведение проектов и представить расписание в графическом виде, что очень удобно для сложных проектов.

6.22. Что такое сетевое программное обеспечение?

Сетевое программное обеспечение предназначено для организации совместной работы группы пользователей на разных компьютерах. Позволяет организовать общую файловую структуру, общие базы данных, доступные каждому члену группы. Обеспечивает возможность передачи сообщений и работы над общими проектами, возможность разделения ресурсов.

Функции и характеристики сетевых операционных систем (ОС)

(по материалам [www-сайта Омского государственного технического университета](http://www.omgtu.omsktelecom.ru)
<http://edu.omgtu.omsktelecom.ru>)

К основным функциям сетевых ОС относят:

- управление каталогами и файлами;

- управление ресурсами;
- коммуникационные функции;
- защиту от несанкционированного доступа;
- обеспечение отказоустойчивости;
- управление сетью.

Управление каталогами и файлами в сетях заключается в обеспечении доступа к данным, физически расположенным в других узлах сети. Управление осуществляется с помощью специальной **сетевой файловой системы**. Файловая система позволяет обращаться к файлам путем применения привычных для локальной работы языковых средств. При обмене файлами должен быть обеспечен необходимый уровень конфиденциальности обмена (секретности данных).

Управление ресурсами включает обслуживание запросов на предоставление ресурсов, доступных по сети.

Коммуникационные функции обеспечивают адресацию, буферизацию, выбор направления для движения данных в разветвленной сети (маршрутизацию), управление потоками данных и др.

Защита от несанкционированного доступа — важная функция, способствующая поддержанию целостности данных и их конфиденциальности. Средства защиты могут разрешать доступ к определенным данным только с некоторых терминалов, в оговоренное время, определенное число раз и т.п. У каждого пользователя в корпоративной сети могут быть свои права доступа с ограничением совокупности доступных директорий или списка возможных действий, например, может быть запрещено изменение содержимого некоторых файлов.

Отказоустойчивость характеризуется сохранением работоспособности системы при воздействии дестабилизирующих факторов. Отказоустойчивость обеспечивается применением для серверов автономных источников питания, отображением или дублированием информации в дисковых накопителях. Под отображением обычно понимают наличие в системе двух копий данных с их расположением на разных дисках, но подключенных к одному контроллеру. Дублирование отличается тем, что для каждого из дисков с копиями используются разные контроллеры. Очевидно, что дублирование более надежно. Дальнейшее повышение отказоустойчивости связано с дублированием серверов, что однако требует дополнительных затрат на приобретение оборудования.

Управление сетью связано с применением соответствующих протоколов управления. Программное обеспечение управления сетью обычно состоит из менеджеров и агентов. **Менеджером** называется программа, вырабатывающая сетевые команды. **Агенты** представляют собой программы, расположенные в различных узлах сети. Они выполняют команды менеджеров, следят за состоянием узлов, собирают информацию о параметрах их функционирования, сигнализируют о происходящих событиях, фиксируют аномалии, следят за трафиком, осуществляют защиту от вирусов. Агенты с достаточной степенью интеллектуальности могут участвовать в восстановлении информации после сбоев, в корректировке параметров управления и т.п.

Программное обеспечение сетевых ОС распределено по узлам сети. Имеется ядро ОС, выполняющее большинство из охарактеризованных выше функций, и дополнительные программы (службы), ориентированные на реализацию протоколов верхних уровней, выполнение специфических функций для коммутационных серверов,

организацию распределенных вычислений и т.п. К сетевому программному обеспечению относят также **драйверы сетевых плат**. Для каждого типа ЛВС разработаны разные типы плат и драйверов. Внутри каждого типа ЛВС может быть много разновидностей плат с разными характеристиками интеллектуальности, скорости, объема буферной памяти.

В настоящее время наибольшее распространение получили три основные сетевые ОС — **UNIX, Windows NT и Novell Netware**.

ОС UNIX применяют преимущественно в крупных корпоративных сетях, поскольку эта система характеризуется высокой надежностью, возможностью легкого масштабирования сети. В Unix имеется ряд команд и поддерживающих их программ для работы в сети. Во-первых, это команды **ftp, telnet**, реализующие файловый обмен и эмуляцию удаленного узла на базе протоколов **TCP/IP**. Во-вторых, протокол, команды и программы **UUCP**, разработанные с ориентацией на асинхронную модемную связь по телефонным линиям между удаленными Unix-узлами в корпоративных и территориальных сетях.

ОС Windows NT включает серверную (**Windows NT Server**) и клиентскую (**Windows NT Workstation**) части и, тем самым, обеспечивает работу в сетях "клиент/сервер". Windows NT обычно применяют в средних по масштабам сетях.

ОС Novell Netware состоит из серверной части и оболочек **Shell**, размещаемых в клиентских узлах. Предоставляет пользователям возможность совместно использовать файлы, принтеры и другое оборудование. Содержит службу каталогов, общую распределённую базу данных пользователей и ресурсов сети. Эту ОС чаще применяют в небольших сетях.

6.23. Вопросы для самоконтроля.

- 6.1. Что такое программа?
- 6.2. Что включает в себя понятие "программное обеспечение"?
- 6.3. Назовите и характеризуйте основные категории программного обеспечения.
- 6.4. В чем отличие прикладных программ от системных и инструментальных?
- 6.5. Что входит в системное программное обеспечение?
- 6.6. В чем состоит назначение операционной системы?
- 6.7. Характеризуйте основные классы операционных систем.
- 6.8. Опишите процесс начальной загрузки операционной системы в оперативную память компьютера.
- 6.9. Что такое файл?
- 6.10. Как организована файловая система?
- 6.11. Какой модуль операционной системы осуществляет обслуживание файлов?

- 6.12.** Приведите пример иерархической файловой структуры.
- 6.13.** Что такое базовая система ввода-вывода (BIOS), и в каком разделе памяти она размещается?
- 6.14.** Из каких основных модулей состоит операционная система MS-DOS?
- 6.15.** Назовите основные разновидности программ-утилит и дайте им краткую характеристику.
- 6.16.** К каким категориям программного обеспечения относятся программные пакеты:
- Norton Commander;
 - MS-DOS;
 - Windows 3.x;
 - Windows-NT, Windows 95;
 - Microsoft Word;
 - Adobe PageMaker;
 - Turbo Pascal, Turbo Basic;
 - Microsoft Excel, Lotus;
 - FoxPro, Access for Windows;
 - Microsoft Office, Microsoft Works?
- 6.17.** Для чего предназначен пакет программ Norton Commander?
- 6.18.** Какой вид интерфейса удобнее для пользователя — командный или графический?
- 6.19.** Чем объясняется широкая популярность пакета Norton Commander?
- 6.20.** Что такое компьютерные вирусы, в чем состоят их вредные действия?
- 6.21.** Какие существуют средства борьбы с компьютерными вирусами?
- 6.22.** В чем суть процесса сжатия информации?
- 6.23.** Характеризуйте основные особенности операционных систем Windows-NT и Windows 95.
- 6.24.** Какие языки и системы программирования вы знаете и в чем их особенности?
- 6.25.** В чем отличие процесса интерпретации от процесса компиляции?
- 6.26.** Назовите основные функции текстовых редакторов.
- 6.27.** Какие дополнительные возможности редактирования текстов обеспечивают полнофункциональные издательские системы по сравнению с текстовыми редакторами?
- 6.28.** Назовите функциональные возможности табличного процессора.
- 6.29.** Какие виды входных данных могут быть введены в клетки электронных таблиц?

6.30. Дайте определение и опишите назначение базы данных.

6.31. Приведите пример возможного наполнения базы данных вашего учебного заведения.

6.32. Каковы основные функциональные возможности систем управления базами данных?

6.33. Что такое информационно-поисковые системы?

6.34. Дайте определение пакета прикладных программ.

6.35. Каково назначение библиотек стандартных программ?

6.36. Дайте определения интегрированного пакета программ.

6.37. Каково назначение сетевого программного обеспечения?
