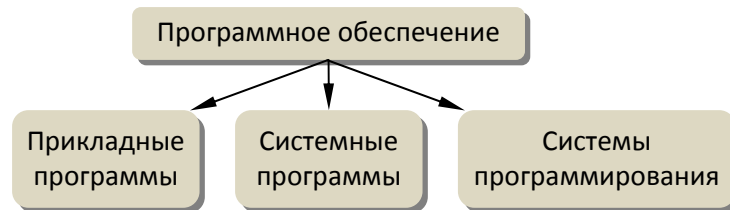


Глава 6. Программное обеспечение

6.1. Что такое программное обеспечение?

Компьютер – это электронное устройство, которое само по себе ничего не умеет. Чтобы использовать его для решения каких-то задач, необходимо **программное обеспечение** (англ. *software* – «мягкое оборудование») – программы, в которых заложены алгоритмы ввода, обработки и вывода данных.

Обычно выделяют три вида программного обеспечения (ПО): *прикладные программы, системные программы и системы программирования*.

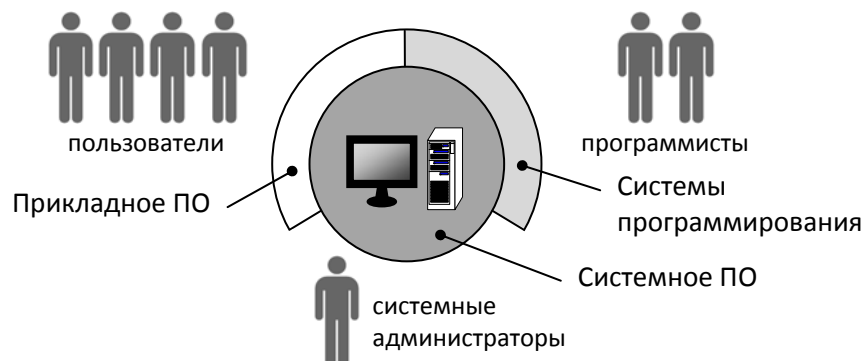


Всех, кто работает с компьютерами, можно разделить на *пользователей, системных администраторов и программистов*.

Пользователи решают свои задачи с помощью прикладных программ (к ним относятся текстовые и графические редакторы, электронные таблицы, базы данных, игры и т.п.).

Системные программы выполняют вспомогательную роль – они обеспечивают пользователю и прикладным программам удобный *интерфейс* (способ обмена данными) с аппаратными средствами. К этой группе относятся операционные системы, *драйверы* (программы для управления внешними устройствами) и *утилиты* (служебные программы). Задача системных администраторов – настроить системное и прикладное ПО, чтобы пользователи смогли нормально работать.

Программисты создают новые программы с помощью *систем программирования* (инструментальных средств).



Программное обеспечение освобождает человека от необходимости работать напрямую с компьютерным «железом» (аппаратными средствами, англ. *hardware* – «жесткое оборудование»).

Часто термин «программное обеспечение» понимают в широком смысле как целую отрасль, включающую все этапы разработки программ, в том числе тестирование (проверку программ, поиск ошибок) и разработку документации.



Контрольные вопросы

1. Назовите три типа программного обеспечения. Чем они отличаются?
2. Какие задачи решают пользователи, программисты, системные администраторы?
3. Что означает слово «интерфейс»?
4. Что такое драйверы, утилиты?
5. Что обозначают английские термины *hardware* и *software*?

6.2. Прикладные программы

6.2.1. Текстовые редакторы

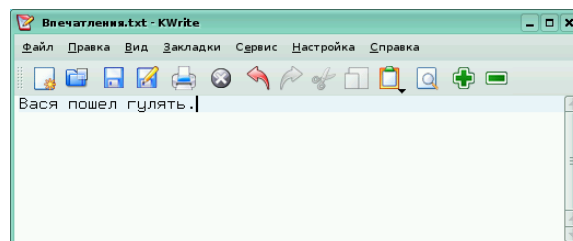
Многие пользователи используют компьютер, прежде всего, для работы с текстами. Обычно различают *редактирование текста* (изменение *содержания*; замена, вставка и удаление символов и слов; разбивка на абзацы) и *форматирование* (изменение *внешнего вида* текста – выбор шрифта, изменение размера, цвета и т.п.).

Простейшие программы этого класса – **текстовые редакторы** – умеют только редактировать текст. Они работают с файлами в формате «только текст» (англ. *plain text*), в которых хранятся коды символов без оформления. Современные редакторы умеют сохранять текст в разных кодировках, но чаще всего используются кодировки семейства UNICODE: UTF-16 (2 байта на символ) или UTF-8 (с переменным числом байт на символ).

Примеры текстовых редакторов:

- *Блокнот* в операционной системе *Windows*;
- *nano*, *gedit*, *KWrite* и *Kate* в системе *Linux*.

На рисунке справа показано окно текстового редактора **KWrite**.



Основные **возможности** современных текстовых редакторов:

- ввод и редактирование текста;
- создание, открытие, редактирование, сохранение и печать документов типа «только текст»;
- работа с буфером обмена (копирование, вырезание, вставка);
- отмена последних операций;
- поиск и замена фрагментов текста;
- подсветка ключевых слов языков программирования (*C*, *Паскаль* и др.) и разметки (*XML*, *HTML*, *LaTeX*);
- проверка орфографии.

Текстовые редакторы часто используют системные администраторы для редактирования файлов с настройками программ (файлов конфигурации). Тексты программ тоже хранятся в формате «только текст», поэтому программисты набирают их в текстовых редакторах.

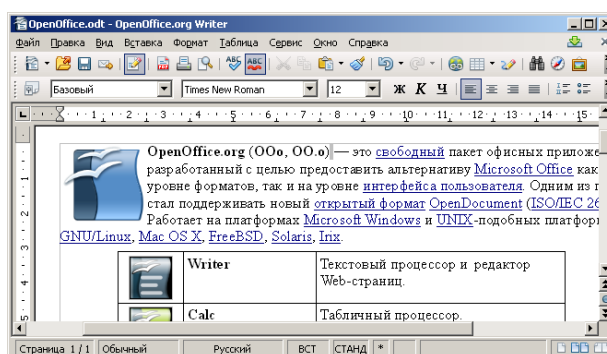
6.2.2. Офисные пакеты

Для подготовки офисных документов возможностей простейших редакторов недостаточно – нужно менять шрифт, выделять фрагменты жирным шрифтом, курсивом и подчеркиванием, добавлять таблицы, графики, рисунки и т.п. Кроме того, возникают и другие задачи: выполнение табличных расчетов, подготовка презентаций для выступлений и докладов, работа с базами данных и т.п. Набор программ для подготовки электронных документов называют «офисным пакетом». В офисный пакет обычно включается

- *текстовый процессор*, который позволяет не только редактировать текст, но и оформлять его по стандартам современного делопроизводства;
- *табличный процессор* – программа для выполнения расчетов с табличными данными;
- программа для подготовки *презентаций*;
- программа для работы с *базами данных*.

Самые известные офисные пакеты –  *Microsoft Office* (www.microsoft.com),  *OpenOffice.org* (openoffice.org) и *WordPerfect Office* (www.corel.com). Пакеты *Microsoft Office* и *WordPerfect Office* – коммерческие, а *OpenOffice.org* можно установить и использовать бесплатно. Кроме того, пакет *OpenOffice.org* – это *кроссплатформенное ПО*, то есть существуют его версии для разных операционных систем (*Windows, Linux*).

Текстовые процессоры – это следующий шаг в развитии редакторов текста. На рисунке показано окно текстового процессора  *OpenOffice.org Writer*:



В состав пакета *Microsoft Office* входит текстовый процессор  *Microsoft Word*, который фактически считается стандартным средством для оформления офисных документов.

С помощью текстовых процессоров можно не только редактировать, но и *форматировать* текст (изменять его оформление). Кроме того, они позволяют

- создавать составные документы, включающие списки, рисунки, таблицы, диаграммы;
- использовать стили оформления (например, заголовки разного уровня);
- использовать шаблоны (заранее оформленные заготовки) документов;
- выполнять несложные вычисления в таблицах;
- сохранять документ в разных форматах, в том числе в *HTML* (как веб-страницу) и *PDF* (универсальный формат документов).

Табличные процессоры (электронные таблицы, англ. *spreadsheet*) – это программы для обработки табличных данных. В отличие от текстовых процессоров, они не только хранят данные, но и позволяют выполнять с ними достаточно сложные вычисления, строить диаграммы, проводить анализ, делать прогнозы. Сейчас электронные таблицы – незаменимый рабочий инструмент экономистов, бухгалтеров, менеджеров. В состав пакета *Microsoft Office* включен табличный процессор  *Excel*, а в пакете *OpenOffice.org* есть близкая по возможностям программа  *OpenOffice.org Calc*.

Компьютерная презентация (лат. *praesentatio* – представление) – это набор изображений (*слайдов*), который предназначен для иллюстрации доклада или выступления. Задача презентации – улучшить восприятие информации. В современных презентациях применяют технологии *мультимедиа* (от лат. *multum* – множество, *medium* – средство), то есть в одном документе используют различные формы представления информации: текст, графика, звук, анимация, видео.

Для создания презентаций в пакете *Microsoft Office* применяется программа  *Microsoft PowerPoint*, а в пакете *OpenOffice.org* – программа  *OpenOffice.org Impress*.

Система управления базами данных (СУБД) – это ПО для поиска информации в базах данных, а также для создания и изменения баз данных. В пакет *Microsoft Office* входит СУБД  *Access*, а в пакет *OpenOffice.org* – СУБД  *OpenOffice.org Base*.

6.2.3. Онлайн-офис

Бурное развитие Интернета привело к появлению «онлайн-офисов» – специальных сайтов (Интернет-сервисов), которые предоставляют основные возможности офисных пакетов: текстовый редактор, электронные таблицы, средства для создания презентаций. Для использования такой службы необходим компьютер с доступом в Интернет, причем не имеет значения, какая операционная система на нем установлена. Документы пользователей хранятся на сервере, для доступа к ним нужно ввести пароль. Самый известный «онлайн-офис» – *Google Docs* (docs.google.com).

Одно из достоинств «онлайн-офисов» – возможность совместной работы над документами через Интернет. Другим пользователям можно открыть доступ к отдельным документам для просмотра и/или изменения. Любой документ может быть *экспортирован* (сохранен) в файл на диске компьютера.

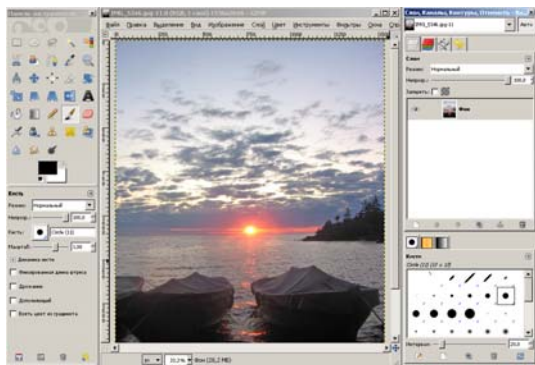
«Онлайн-офисы» используют технологию, известную под названием «облачные вычисления» (англ. *cloud computing*). Ее суть в том, что пользователь хранит свои данные на серверах Интернета и не должен заботиться о способе их хранения, операционной системе и программном обеспечении. Слово «облако» – это метафора, образ достаточно сложной системы, детали работы которой знать не обязательно. Несмотря на удобства «облачных» сервисов, существуют опасения, что пользователь может потерять контроль над своими данными, и это чревато серьезными проблемами. Например, иногда не удастся полностью удалить данные, которые человек сам же разместил. Кроме того, возможны утечки информации и потеря данных.

6.2.4. Графические редакторы

Графические редакторы – это программы для создания и редактирование изображений. Изображения, хранящиеся в компьютере, делятся на растровые и векторные (см. разд. 2.13). С ними нужно работать по-разному, поэтому различают растровые и векторные графические редакторы¹.

Растровые редакторы предназначены для

- обработки фотографий;
- подготовки изображений к печати;
- создания и редактирования изображений для веб-сайтов.



Лучшим профессиональным растровым редактором считается программа  *Adobe Photoshop* (www.adobe.com). Существуют ее версии для операционных систем *Windows* и *Mac OS X* (для компьютеров фирмы *Apple*). В состав *Windows* входит растровый редактор  *Paint*, но для сложной обработки (например, для коррекции фотографий) его возможности недостаточны.

Бесплатная программа  *Gimp* (gimp.org) – кроссплатформенная, она работает как в *Windows*, так и в *Linux*. На рисунке слева показано окно программы *Gimp*.

¹ Более точно называть их редакторами растровой и векторной графики.

Среди бесплатных растровых редакторов широкими возможностями обладает  **Paint.NET** (www.getpaint.net), но он пока устойчиво работает только в среде *Windows*.

В последнее время были созданы бесплатные «онлайновые» редакторы (например, www.pixlr.com), которые позволяют обрабатывать изображения на специальной веб-странице в Интернете, без установки дополнительного ПО на компьютер пользователя.

Векторные редакторы используются для подготовки

- художественных иллюстраций;
- технических иллюстраций (схем, графиков);
- логотипов, визиток, плакатов;
- изображений для веб-сайтов (иконок, кнопок).

Среди профессиональных векторных редакторов можно назвать двух лидеров – программы  **Adobe Illustrator** (www.adobe.com) и  **CorelDraw** (www.corel.com). В свободно распространяемый пакет *OpenOffice.org* входит векторный редактор  **OpenOffice.org Draw**. Бесплатный редактор  **Inkscape** (www.inkscape.org) – еще одна кроссплатформенная программа, работающая как в *Windows*, так и в *Linux*. На рисунке справа показано окно редактора **Inkscape**.

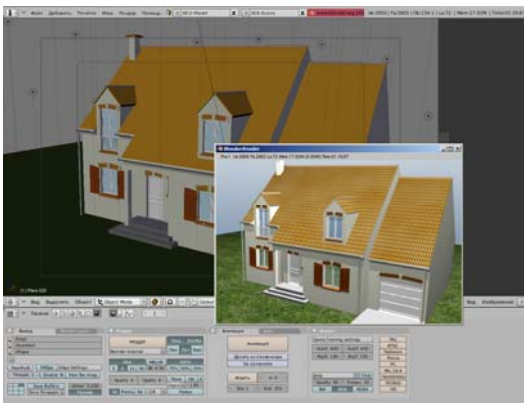


Во многих графических редакторах, например, в *Adobe Photoshop* и *Corel Draw*, можно создавать документы, содержащие как растровую, так и векторную графику. Текстовые процессоры (*Microsoft Word*, *OpenOffice.org Writer*) позволяют вставлять в документ растровые и векторные рисунки.

Во многих областях двумерных рисунков недостаточно, и необходимо представить объект в трехмерном пространстве. Такие задачи возникают, прежде всего, в архитектуре, кино, телевидении и компьютерных играх. С помощью трехмерной графики (англ. *3D = 3 dimensions*, «3 измерения») создаются многие современные мультфильмы.

Для работы с трехмерными объектами используют программы специального класса (программы 3D-моделирования), которые позволяют:

- определить форму (геометрию) объектов сцены;
- задать материалы для объектов;
- установить источники света;
- определить точки наблюдения (виртуальные камеры);
- создать анимацию с трехмерными объектами;
- выполнить *рендеринг*, то есть построить плоскую картинку или последовательность кадров анимации с учетом свойств объектов и источников света.



Среди программ 3D-моделирования, работающих в системе *Windows*, наиболее популярна  **3D Studio MAX** (usa.autodesk.com). В области кино и телевидения стандартом считается кроссплатформенная программа  **Maya** (www.autodesk.com/maya), версии которой существуют для ОС *Linux*, *Windows* и *Mac OS*. Среди свободно распространяемых программ наиболее известны кроссплатформенная программа  **Blender** (www.blender.org), а также *K-3D* (www.k-3d.org) и  **Wings-3D** (www.wings3d.com).

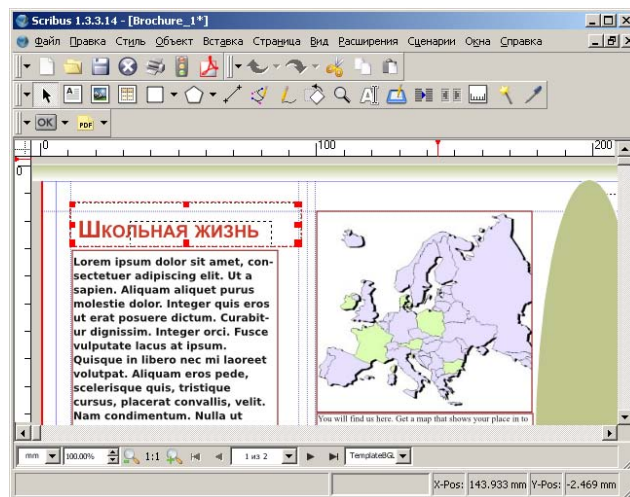
Алгоритмы работы с трехмерной графикой достаточно сложны и требуют большого объема вычислений. Поэтому для нормальной работы программ 3D-моделирования, особенно для выполнения рендеринга, требуется быстродействующий процессор и много оперативной памяти.

6.2.5. Настольно-издательские системы

В любой современной типографии для подготовки к печати книг и журналов используется специальное программное обеспечение – **настольно-издательские системы** (англ. *DTP = Desktop publishing*, «настольное издательство»).

Основное отличие этих программ от текстовых процессоров состоит в том, что в них можно выполнять *вёрстку* – точно задавать расположение текста, рисунков, таблиц и другого материала на странице в соответствии с правилами типографского дела. В настольно-издательских системах готовят *оригинал-макет* (изображение, точно совпадающее с будущим отпечатком) и отправляют его в типографию.

Долгое время промышленным стандартом была настольно-издательская система  **QuarkXPress** (www.quark.com), сейчас конкуренцию ей составляют  **Adobe InDesign** (www.adobe.com) и бесплатная программа  **Scribus** (www.scribus.net), окно которой показано на рисунке.



В состав пакета *Microsoft Office* входит программа верстки  **Microsoft Publisher** с несколько меньшими возможностями. Верстку несложных изданий (например, визиток, буклетов) можно выполнить в программе  **CorelDraw**, но профессионалы не рекомендуют ее использовать.

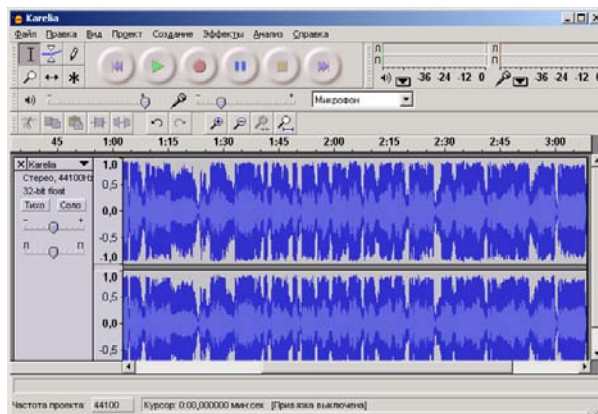
Лучшая система подготовки математических текстов называется *TeX* (ctan.org). Ее разработал известный американский математик и специалист по теоретической информатике Д. Кнут. Система *TeX* – бесплатная и кроссплатформенная, она принята в качестве стандарта во многих российских и зарубежных издательствах, выпускающих математическую литературу.

6.2.6. Редакторы звука и видео

Аудиоредакторы – это программы для редактирования звуковых файлов. С их помощью можно

- загружать, редактировать и сохранять звуковые файлы разных форматов;
- записывать звук с микрофона или другого источника;
- вырезать фрагменты из файла;
- соединять звуковые фрагменты в один файл;
- изменять громкость и темп звука;
- удалять шумы.

Самые известные аудиоредакторы – это  **Adobe Audition** (www.adobe.com),  **Sound Forge** (www.sonycreativesoftware.com),  **Audacity** (audacity.sourceforge.net). В отличие от первых



двух программ, *Audacity* – бесплатный продукт, существующий для многих операционных систем (*Windows, Linux, Mac OS X*). На рисунке показано окно программы *Audacity*.

Видеоредакторы предназначены для создания и редактирования цифрового видео. Их основные возможности:

- ввод данных с видеокамеры;
- коррекция цвета;
- добавление, перестановка, удаление фрагментов фильма;
- добавление звука и титров;
- сохранение фильма в различных цифровых видеоформатах;
- создание DVD-дисков.



Среди коммерческих программ этого типа наиболее популярны *Adobe Premier* (www.adobe.com), *Pinnacle Studio* (www.pinnaclesys.com), *VideoStudio Pro* (www.corel.com).

На компьютерах фирмы *Apple* используется видеоредактор *iMovie* (www.apple.com).

Существуют и бесплатные видеоредакторы, например, программа *Kino* для операционной системы *Linux* (kinodv.org), окно которой показано на рисунке.

6.2.7. ПО для работы в Интернете

Для просмотра веб-страниц в Интернете нужна специальная программа, которую называют **браузер** (англ. *browser*). Пользователи Интернета чаще всего используют браузеры

- Internet Explorer* (входит в состав операционной системы *Windows*);
- Firefox* (www.mozilla-russia.org);
- Chrome* (www.google.com/chrome);
- Safari* (www.apple.com/safari);
- Opera* (www.opera.com).

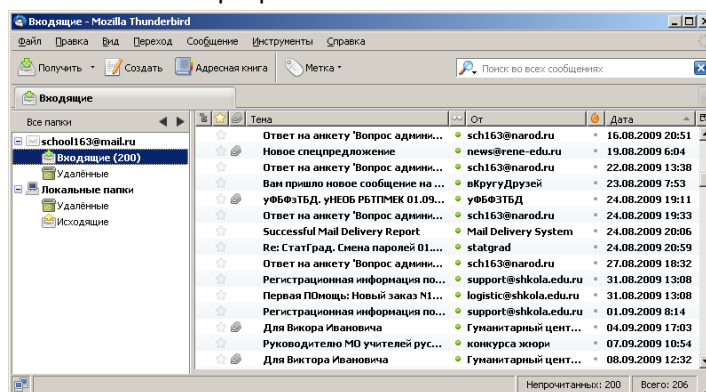
Большинство браузеров бесплатные, многие из них кроссплатформенные. Например, версии браузеров *Firefox* и *Opera* существуют для большинства современных операционных систем. В состав *Linux* входит браузер *Konqueror*, но пользователи чаще всего предпочитают *Firefox*.

Кроме браузера, большинство пользователей используют **почтовые программы** (*почтовые клиенты*), предназначенные для работы с электронной почтой. Их основные возможности:

- создание, отправка и прием сообщений;
- автоматическая проверка почты через заданный интервал;
- сортировка сообщений по папкам;
- ведение адресной книги (списка контактов).

В состав современных версий операционной системы *Windows* входит почтовая программа *Почта Windows*. Несколько большими возможностями обладают профессиональные программы *Microsoft Outlook* (входящая в пакет *Microsoft Office*) и *TheBat* (www.ritlabs.com). На компьютерах фирмы *Apple* устанавливается почтовый клиент *Apple Mail* (www.apple.com). В состав браузера *Opera* входит встроенный почтовый клиент *Opera Mail*, возможностей которого вполне достаточно для простых задач.

Самая известная из бесплатных программ –  *Mozilla Thunderbird* (www.mozilla-russia.org) – является кроссплатформенной: существуют ее версии для операционных систем *Windows*, *Linux*, *Mac OS X*. На рисунке показано окно программы *Mozilla Thunderbird*:



Для общения в *реальном времени* (это значит, что собеседники одновременно находятся за компьютерами) используют программы для обмена мгновенными сообщениями (**мессенджеры**). Самые известные мессенджеры –  *ICQ* (www.icq.com),  *Mail.ru Агент* (www.mail.ru),  *Kopete* (для *Linux*),  *iChat* (для компьютеров фирмы *Apple*). В последнее время все большую популярность приобретает система мгновенного обмена сообщениями  *Jabber* (www.jabber.org), также известная под названием  *XMPP*. Она бесплатная, обеспечивает надежную защиту передаваемых данных, позволяет устанавливать связь с пользователями других систем, например, *ICQ*.

С помощью программы  *Skype* (skype.com) можно установить через Интернет голосовую и видеосвязь между компьютерами. Для этого на каждом из компьютеров должен быть микрофон и веб-камера.



Контрольные вопросы

1. Какое ПО называют прикладным?
2. Чем отличаются текстовые редакторы и текстовые процессоры?
3. Что означает формат «только текст»? В каких случаях он используется?
4. Какие программы обычно входят в офисный пакет?
5. Что такое кроссплатформенное ПО?
6. Что такое СУБД?
7. Что такое «онлайн-офис»? В чем его достоинства и недостатки?
8. Что такое «облачные вычисления»?
9. Чем отличаются растровые и векторные графические редакторы?
10. Перечислите возможности растровых редакторов.
11. Для каких целей используются векторные редакторы?
12. Что такое настольно-издательская система? Чем она отличается от текстового процессора?
13. Что такое оригинал-макет?
14. Какая система лучше всего подходит для набора математических текстов?
15. Какими возможностями обладают аудиоредакторы?
16. Перечислите возможности редакторов видео.
17. Что такое браузер?
18. Перечислите возможности почтовых программ.
19. Что такое мессенджер?

6.3. Системное программное обеспечение

6.3.1. Что такое операционная система?

Команды, которые умеет выполнять процессор представляют собой числовые коды. Чтобы он выполнил программу, нужно эту программу загрузить в память и передать процессору адрес первой команды. В принципе, это можно делать вручную, с помощью переключателей (1/0) или перфокарт, так и было на первых компьютерах. Однако в этом случае ввод программы будет значительно занимать больше времени, чем ее выполнение, поэтому процессор будет простаивать. Кроме того, для ввода и вывода данных нужно программировать внешние устройства, каждое из которых имеет собственный набор команд. В таких условиях с компьютером могут работать только высококвалифицированные специалисты. Ситуация еще более усложняется, если требуется записать данные на жесткий диск или обеспечить одновременную работу нескольких программ.

Для решения всех этих проблем программисты разработали вспомогательные программы (точнее, программные системы, состоящие из многих программ), которые называются *операционными системами*.

Операционная система (ОС) – это комплекс программ, обеспечивающих пользователю и прикладным программам удобный **интерфейс** (способ обмена информацией) с аппаратными средствами компьютера.

Операционная система **обеспечивает**:

- взаимодействие пользователя и аппаратных средств;
- обмен данными между прикладными программами и устройствами компьютера;
- работу файловой системы (хранение данных в виде файлов и папок);
- запуск и выполнение прикладных программ;
- обработку ошибок, контроль за работой оборудования;
- распределение ресурсов компьютера между несколькими одновременно работающими программами (время работы процессора, память, внешние устройства).

Операционные системы бывают *однозадачные* (на компьютере в любой момент выполняется только одна программа) и *многозадачными* (пользователь может запустить несколько программ, которые будут выполняться одновременно).

Первые операционные системы появились на компьютерах второго поколения и были *однозадачными*. Нередко получалось так, что большую часть времени занимали не вычисления, а операции ввода и вывода данных, тогда как процессор в это время простаивал. Чтобы полностью использовать мощность компьютера, разработали *пакетный режим*: в разные области памяти загружали несколько программ. Когда одна программа выполняла операции ввода-вывода, процессор переходил к выполнению следующей, и таким образом мог быть загружен практически на полную мощность.

На компьютерах третьего поколения часто использовался *многопользовательский режим* (режим разделения времени), при котором с большим компьютером (*мэйнфреймом*) было связано несколько *терминалов* (так называли рабочие места с клавиатурой и монитором). С каждого терминала можно было отправить задание на выполнение, таким образом, с компьютером одновременно работало несколько программистов.

Операционные системы первых персональных компьютеров были однозадачными. Самая популярная ОС в 1980-х годах – *MS DOS* (*Microsoft Disk Operating System* – дисковая операционная система фирмы *Microsoft*). Сейчас иногда на дешевые ноутбуки устанавливается ее бесплатный аналог – *FreeDOS* (www.freedos.org).

Все современные ОС – *многозадачные*. ОС распределяет время работы процессора между запущенными программами, выделяя каждой *кванты* (небольшие интервалы) времени, так что создается впечатление, что программы работают одновременно, даже если на компьютере установлен один процессор.

В **состав операционной системы** обычно входят:

- *начальный загрузчик* – небольшая программа, расположенная в первом секторе загрузочного диска; его задача – организовать загрузку в память *ядра* (основной части) ОС и передать ему управление;
- *система управления памятью*;
- *система ввода и вывода*, которая управляет внешними устройствами и файлами; она использует программы для обмена данными с дисководом, клавиатурой, монитором и принтером, записанные в постоянном запоминающем устройстве (ПЗУ) микросхемы BIOS², расположенной на материнской плате;
- *командный процессор* – программа, которая выполняет команды пользователя, введенные в командной строке, и *командные файлы* – текстовые файлы, содержащие списки команд и даже программы на специальном языке программирования;
- *утилиты* (лат. *utilitas* – польза) – служебные программы для проверки и настройки компьютера.



Микросхема BIOS

Может ли компьютер работать без операционной системы? Да, в том случае, если он работает по одной единственной программе, которая хранится в ПЗУ или на диске, и автоматически запускается при включении питания. Например, микрокомпьютеры, встроенные в бытовые устройства, могут обходиться без операционной системы. Однако такой компьютер невозможно настраивать, поэтому во многих более сложных устройствах (игровых приставках, банковских терминалах и т.д.) используют операционные системы.

6.3.2. Современные операционные системы

Самые популярные современные операционные системы для персональных компьютеров – *Windows*, *Mac OS X*³ и *Linux*. Все они используют графический интерфейс с пользователем: окна программ, управление с помощью мыши, кнопки, переключатели и т.п.

Система  *Windows* разработана фирмой *Microsoft* (www.microsoft.com) и распространяется на коммерческой основе. По управлению *Windows* работает более 90% персональных компьютеров⁴, имеющих доступ в Интернет.

Примерно 5% пользователей используют  *Mac OS X*. Она устанавливается на компьютерах фирмы *Apple*, которые часто используют профессионалы в области дизайна, компьютерной графики, полиграфии, видеомонтажа.

Около 1% компьютеров работают под управлением ОС  *Linux*. Ее начал разрабатывать в 1991 году финский студент *Линус Торвальдс* в качестве хобби. Сейчас в развитии *Linux* принимают участие сотни разработчиков во всём мире. В современном ядре *Linux* насчитывается более 11 млн. строк кода⁵. Система *Linux* распространяется бесплатно вместе с исходными кодами, так что каждый (при желании и умении) может ее усовершенствовать.

² BIOS = Basic Input-Output System – базовая система ввода-вывода.

³ Последний символ в названии *Mac OS X* – это римская цифра X, что означает «версия 10».

⁴ Данные сайта www.netmarketshare.com (2010 г.)

⁵ Данные 2009 г.

На основе ядра *Linux* построено много различных *дистрибутивов* (распространяемых сборок), самые известные из них – *Ubuntu* (ubuntu.com), *Mandriva* (mandriva.ru), *OpenSUSE* (opensuse.org), *Slackware* (www.slackware.com), *Gentoo* (www.gentoo.org). В дистрибутивы входит не только сама операционная система, но и программное обеспечение, состав которого зависит от конкретной сборки. Существуют дистрибутивы с улучшенной поддержкой русского языка, например, *ALT Linux* (www.altlinux.org).

Достоинства *Linux*:

- бесплатное распространение ОС и многих программ для нее;
- высокий уровень безопасности и защиты от вирусов;
- невысокие требования к аппаратным средствам;
- возможность гибкой настройки.

Основные сферы применения *Linux*:

- личные компьютеры (не нужно платить за ПО);
- портативные компьютеры, которые покупаются организациями в большом количестве;
- серверы в локальных сетях и в Интернете (до 50%), важно быстродействие;
- суперкомпьютеры (до 80%⁶), важна возможность настройки для работы на нестандартном оборудовании;
- встроенные компьютеры в банкоматах, терминалах оплаты, стиральных машинах и даже беспилотных военных аппаратах; важна бесплатность и возможности настройки.

Среди *недостатков* этой ОС обычно отмечают:

- сложность настройки для неквалифицированного пользователя (для выполнения многих операций необходимо вводить команды в режиме командной строки);
- отсутствие драйверов для некоторых устройств и сложность их установки;
- отсутствие версий популярных профессиональных программ, например, *Adobe Photoshop*;
- отсутствие поддержки современных игр.

Появление карманных персональных компьютеров (КПК), смартфонов и коммуникаторов привело к развитию нового класса операционных систем, которые могут работать на маломощном оборудовании. Самая популярная **ОС для мобильных устройств** – *Symbian*, с ней конкурируют *Windows Mobile* (разработка фирмы *Microsoft*) и *BlackBerry*. Некоторые ОС этого класса строятся на основе ядра *Linux*: *Google Android*, *Palm webOS*, *Maemo* и другие. Портативные компьютеры фирмы *Apple* (*iPhone*, *iPod Touch*, *iPad*) используют операционную систему *iPhone OS*.

Новая операционная система компании *Google* для персональных компьютеров, названная **Chrome OS**, тоже строится на ядре *Linux*. Она нетребовательна к аппаратным ресурсам компьютера, основная роль отводится веб-браузеру и «облачным вычислениям». Данные пользователя хранятся на серверах Интернета, для их обработки используются веб-службы, при этом на компьютер не нужно устанавливать дополнительное программное обеспечение. Недостаток этой ОС – низкая безопасность. Также она не подойдет тем, кому нужно выполнять сложную обработку графики и видео.

Существует еще один класс операционных систем, от которых требуется не просто решать задачи, а делать это за определённый промежуток времени. Такие ОС называются операционными системами **реального времени**. Они применяются в тех случаях, когда задержка может привести к аварии, катастрофе или финансовым потерям: в системах аварийной защиты, системах управления роботами и самолетами, в военных приборах. Например, робот, снимающий деталь с конвейера, должен сделать это за маленький промежуток времени. Наиболее известные системы

⁶ Данные сайта www.top500.org (2010 г.)

реального времени – QNX (www.qnx.com), Windows CE (www.microsoft.com), VxWorks (www.windriver.com) и LynxOS (www.linuxworks.com/rtos).

Многие современные операционные системы, включая Linux, Mac OS X, QNX, VxWorks, LynxOS, относятся к классу UNIX-подобных ОС. Это значит, что они используют общие идеи и принципы, заложенные в 1970-х годах при разработке системы UNIX:

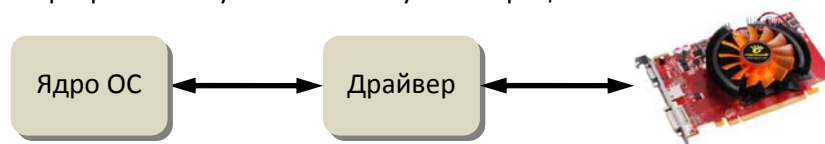
- для настройки и управления системой используются простые текстовые файлы;
- программы часто используют текстовый ввод данных и вывод результатов;
- широко применяются утилиты, запускаемые в командной строке;
- каждая утилита выполняет одну задачу; ее режимы работы можно задавать с помощью параметров командной строки;
- утилиты можно объединять в «конвейер», направляя результаты работы одной утилиты на вход следующей;
- все устройства (жесткие диски, флэш-диски, принтеры, сканеры) рассматриваются как файлы.

Сейчас система UNIX используется в основном для управления серверами. Все UNIX-подобные системы считаются очень надежными с точки зрения безопасности. Достаточно сказать, что для них практически неактуальна проблема компьютерных вирусов.

6.3.3. Драйверы устройств

Драйверы (англ. *driver* – водитель) – это программы специального типа, которые находятся в оперативной памяти и обеспечивают обмен данными между ядром ОС и внешними устройствами компьютера (звуковой картой, видеокартой, сетевой картой, принтером и т.п.). Драйверы обычно включают в подсистему ввода и вывода.

Драйвер представляет собой набор процедур, которые вызываются ядром ОС при необходимости передать данные устройству или принять от него данные. Задача драйвера – преобразовать команды ввода-вывода в команды конкретного устройства. Драйверы загружаются в память и фактически становятся частью ОС. Такая схема позволяет устанавливать и использовать устройства, которые были разработаны уже после выпуска операционной системы.



Если драйвер не установлен, устройство работать не будет, потому что неизвестно, как к нему обращаться. Драйверы наиболее популярных устройств обычно включаются в дистрибутив (установочный пакет) операционной системы. Когда ОС обнаруживает новое устройство, она пытается найти подходящий драйвер в своей базе данных. Если такого драйвера нет, его можно установить вручную с диска, который прилагается к устройству. Кроме того, любой драйвер можно бесплатно скачать из Интернета с сайта производителя.

6.3.4. Утилиты

Утилиты решают вспомогательные задачи, расширяя возможности ОС. К утилитам относятся

- программы для проверки дисков (**chkdsk** в Windows, **fsck** в Linux);
- программы для разбивки жестких дисков, с помощью которых можно сделать несколько разделов на одном диске (Управление дисками в Windows; GNU Parted в Linux);
- файловые менеджеры – программы для работы с файлами; самые известные файловые менеджеры для Windows – Проводник (входит в состав ОС),  Total Commander (www.ghisler.com),  Free Commander (www.freecommander.com),  Far Manager

(farmanager.com); в Mac OS X используется программа  Finder, а в операционной системе Linux – файловые менеджеры  Konqueror,  Midnight Commander и др.;

- **антивирусные программы:**  AVP (www.kaspersky.ru),  DrWeb (www.drweb.com),  Nod32 (www.eset.com),  McAfee (home.mcafee.com) и др.;
- **архиваторы и программы для сжатия данных;** в ОС Windows чаще всего используются  WinRAR (www.rarlab.com) и  WinZip (www.winzip.com); в Linux –  Ark (utils.kde.org) и  File Roller (fileroller.sf.net); архиватор  7ZIP (www.7-zip.org) распространяется бесплатно с исходными кодами для различных операционных систем;
- **программы для шифрования данных,** например, PGP и ее версии для разных операционных систем (www.pgpru.com);
- **редакторы,** позволяющие менять данные на диске и в оперативной памяти; например, программы HxD (mh-nexus.de/en/hxd) и WinHex (www.winhex.com) для ОС Windows или hexedit (rigaux.org/hexedit.html) для Linux;
- **сетевые утилиты** для проверки связи в локальной и глобальной сетях; например, утилиты **ping, tracert (traceroute), nslookup** в Windows и Linux.

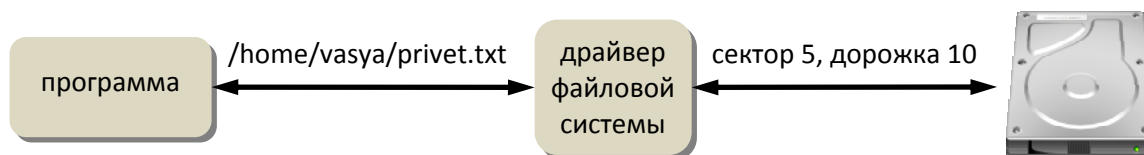
Часто к утилитам относят также

- **программы для записи CD и DVD-дисков;** в системе Windows наиболее популярны программы  Nero Burning ROM (www.nero.com),  CDBurnerXP (cdburnersp.se) и  DeepBurner (www.deepburner.com); в Linux для этой цели используют утилиту  K3b (k3b.org);
- **программы для сканирования и распознавания текста;** широко применяются коммерческая программа  ABBYY FineReader (www.abbyy.ru) и бесплатная CuneiForm (www.cuneiform.ru).

6.3.5. Файловые системы

Вы знаете, что данные можно хранить в виде файлов на жестких дисках, CD и DVD-дисках, флэш-дисках. Однако винчестер и флэш-диск «ничего не знают» о том, что записанные на них данные в самом деле объединены в файлы и папки.

С другой стороны, когда программа сохраняет файл на диск, она «ничего не знает» о том, в какое место диска эта информация будет записана, указывается только имя файла и каталог. Поэтому между программой и носителем информации необходим «посредник», который определяет, в какое именно место диска будут записаны биты переданных данных. Эту роль выполняет *драйвер файловой системы*:



Файловая система – это порядок размещения, хранения и именования данных на носителе информации.

Файловые системы решают несколько **задач**:

- определяют правила построения имен файлов и каталогов;
- определяют, как именно размещаются файлы на диске;
- предоставляют программам функции для работы с файлами;
- обеспечивают защиту данных в случае сбоев и ошибок;
- обеспечивают установку прав доступа к данным для каждого пользователя;

- обеспечивают совместную работу с файлами (когда один пользователь открыл файл, для остальных устанавливается режим «только чтение»).

С точки зрения файловой системы диск делится на *кластеры* (блоки) размером от 512 байт до 64 Кбайт. Каждому файлу выделяется целое число кластеров. Файлу размером 1 байт выделяется целый кластер, остальное место считается занятым, но фактически не используется. Поэтому при большом размере кластера хранить мелкие файлы невыгодно, значительная часть места пропадает впустую. С другой стороны, при увеличении размера кластера скорость чтения и записи больших файлов повышается, кроме того, увеличивается и максимальный объем диска, который поддерживает файловая система.

В операционной системе *Linux* применяются файловые системы **ext3** и **ext4**. Они поддерживают журналирование, помогающее сохранить данные в случае сбоев. Его суть в том, что *перед* выполнением операции с файлами ОС записывает «план действий» в специальный журнал. Когда операция полностью закончена, эта запись из журнала удаляется. Если во время операции произошел сбой (например, отключение питания), по записям в журнале можно сразу определить, какие файлы могли быть затронуты. Таким образом, журналируемая файловая система устойчива к сбоям.

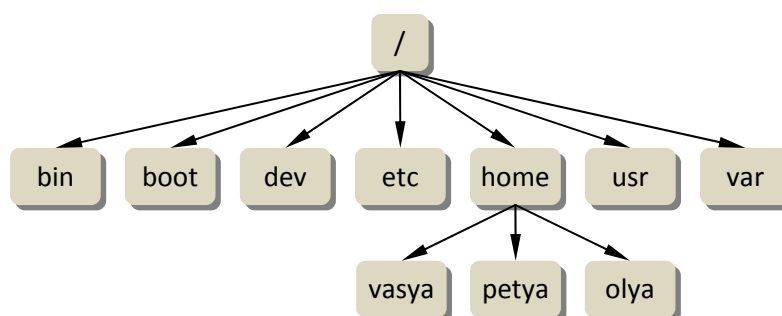
В системе *Windows* применяют файловые системы **NTFS** и **FAT32**. Хотя **FAT32** в некоторых случаях работает быстрее и требует меньше памяти, она считается устаревшей. В отличие от **FAT32**, файловая система **NTFS**

- обеспечивает защиту от сбоев с помощью журналирования (в **FAT32** журналирование не используется);
- позволяет назначить права доступа к файлам и папкам (в **FAT32** каждый пользователь может просматривать и изменять данные всех остальных);
- позволяет задать *квоту* (ограничение) на использование диска для каждого пользователя;
- позволяет использовать сжатие файлов и папок без специальных программ.

В операционной системе *Mac OS X* применяется файловая система **HFS** (англ. *Hierarchical File System*, иерархическая файловая система).

Первые файловые системы были *одноуровневыми*, то есть, все файлы хранились в одном каталоге (на дискете). С увеличением емкости дисков (и количества файлов!) это стало неудобно, поэтому разработали *иерархические* (многоуровневые) файловые системы, где файлы группируются в каталоги, а сами каталоги вложены друг в друга. Такая структура называется *деревом каталогов*.

В операционной системе *Linux* существует один корневой каталог (обозначаемый как «/»), остальные файлы и каталоги вложены в него. Любое устройство (включая жесткие диски, принтеры, сканеры и т.п.) в *Linux* рассматривается как файл, то есть входит в состав единой иерархической файловой системы.



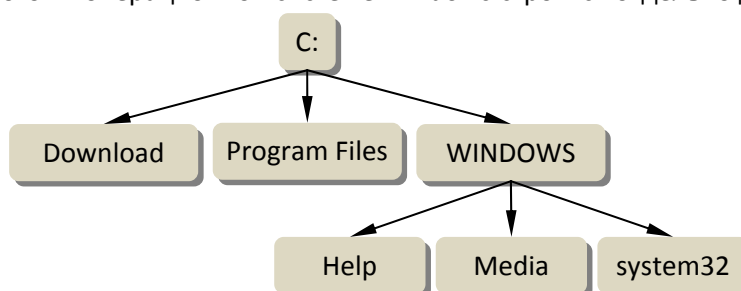
Вот что хранится в показанных каталогах:

- **bin** – команды операционной системы;

- **boot** – ядро ОС и данные для загрузки;
- **dev** – файлы устройств, подключенных к ОС;
- **etc** – файлы с настройками ОС и некоторых программ;
- **home** – домашние каталоги пользователей;
- **usr** – установленные пакеты программ;
- **var** – часто меняющиеся данные, например, журналы ОС.

Чтобы указать путь к файлу или каталогу, перечисляют (начиная от корня) все каталоги, в которых он находится, разделяя их символом «/» (он называется «слэш»). Например, адрес домашнего каталога пользователя **petya** запишется как **/home/petya**, а адрес файла **qq.txt** в этом каталоге – как **/home/petya/qq.txt**.

Дерево каталогов в операционной системе *Windows* строится отдельно для каждого диска:



В качестве разделителя при записи адреса файла или каталога (папки) используют обратный слэш «\», например, **C:\WINDOWS\System32\shell32.dll**.

Для работы с группами файлов используются *маски* или *шаблоны* (англ. *wildcards*). Кроме символов, которые допустимы в именах файлов, маска может включать два специальных символа: знак «*» заменяет любое количество любых символов, а знак «?» – один любой символ. Приведем несколько примеров:

- *.* все файлы;
- *.bmp все файлы с расширением **.bmp**;
- a*.* файлы, имя которых начинается с буквы «а», а расширение состоит из 1 символа
- *x*.*.* файлы, в имени которых есть буква «х», а расширение содержит не менее 2-х символов;
- *z.a* файлы, имя которых заканчивается на букву «z», а расширение начинается с буквы «а» и состоит из 2-х символов.

Маски чаще всего применяют для поиска файла по известной части имени или по расширению. Например, для того, чтобы найти все документы *Word*, у которых имя содержит слово «отчет», можно использовать маску ***отчет*.doc***. При этом будут найдены, например, такие файлы:

отчет2010.doc
Самый_важный_отчет.docx
Новый_отчет_март_2011.docx

В операционной системе *Windows* заглавные и строчные буквы в названиях файлов и каталогов не различаются, то есть к файлу с именем **Вася.txt** можно обращаться как **вася.txt**, **вАся.txt**, **ВаСя.txt** или **ВАСЯ.txt**. В *Unix*-подобных ОС (*Linux*, *Mac OS X*) это не так, все перечисленные имена файлов – разные, и такие файлы могут быть созданы в одном каталоге.

Нужно отметить, что файловая система не обязательно напрямую связана с жестким диском или другим физическим носителем информации. Существуют, например, *сетевые файловые системы*, которые являются просто способом обращения к файлам на удалённых компьютерах.



Контрольные вопросы

1. Зачем нужны операционные системы? Можно ли обойтись без них?
2. Что такое операционная система?
3. Какие задачи выполняет ОС?
4. Чем отличаются многозадачные ОС от однозадачных?
5. Как обеспечивается многозадачность на компьютерах с одним процессором?
6. Что такое пакетный режим работы?
7. Что такое многопользовательский режим?
8. Перечислите составные части ОС.
9. Что такое начальный загрузчик?
10. Что такое драйвер?
11. Что такое утилита?
12. Назовите наиболее популярные современные ОС для персональных компьютеров.
13. Какими достоинствами и недостатками обладает система *Linux*?
14. Какие ОС используются для мобильных устройств?
15. В чем состоят особенности операционной системы *Chrome OS*?
16. Что такое «операционная система реального времени»? Где применяются такие системы?
17. Что такое *UNIX*-подобные ОС? В чем их особенности?
18. Как ОС обменивается данными с внешними устройствами? В чем достоинства такой схемы?
19. Что происходит, когда ОС обнаружила новое устройство?
20. Какие программы относятся к утилитам?
21. Что такое файловая система? Зачем она нужна?
22. Какие задачи решает файловая система?
23. Что такое *кластер*?
24. Почему невыгодно хранить мелкие файлы в файловой системе с большим размером кластера?
25. Что улучшается при увеличении размера кластера?
26. Какие файловые системы используются в ОС *Linux*, *Windows* и *Mac OS X*?
27. Что такое журналируемая файловая система? Какие журналируемые системы вы знаете?
28. Почему файловая система **FAT32** считается устаревшей?
29. Что такое одноуровневая и многоуровневая файловые системы?
30. Что такое корневой каталог?
31. В чем отличие файловых систем в ОС *Linux* и *Windows*?
32. Где расположены каталоги пользователей в ОС *Linux*?
33. Какие символы используются как разделители при записи адреса файла в ОС *Linux* и *Windows*?
34. Что такое сетевая файловая система?



Задачи

1. Определите, какое из указанных имен файлов удовлетворяет маске: **?hel*lo.c?***
а) **hello.c** б) **hello.cpp** в) **hhelolo.cpp** г) **hhelolo.c**
д) **hello.cc** е) **ahello.cpp** ж) **ahelolo.c** з) **ahelolo.cp**
(Ответ: в, е, з)

2. Пользователь **vasya** работает в ОС *Linux*. Перемещаясь из одного каталога в другой, он последовательно посетил каталоги **math**, **lections**, **professor** и оказался в своем домашнем каталоге. Каково полное имя каталога, из которого начал перемещение пользователь?
(Ответ: **/home/vasya/professor/lections/math**)
3. Пользователь ОС *Windows*, перемещаясь из одного каталога в другой, последовательно посетил каталоги **Математика**, **Задания**, **С:**, **Классы**, **10-А**. Каково полное имя каталога, из которого начал перемещение пользователь?
(Ответ: **С:\Задания\Математика**)

6.4. Системы программирования

6.4.1. Зачем нужны системы программирования?

Процессор, выполняющий всю обработку данных, понимает только машинные команды (числовые коды). Чаще всего их записывают в шестнадцатеричном виде, например, так:

B82301052500

Чтобы понять, что делает этот код, нужно взять таблицу команд процессора и посмотреть, что означает каждая пара шестнадцатеричных цифр (байт).

Программы для первых компьютеров составляли именно в машинных кодах. Программирование было доступно только специалистам, отладка программы занимала очень много времени. Человек плохо воспринимает коды, поэтому для каждой машинной команды придумали символические обозначения. Например, приведенная выше программа – это две машинные команды для процессоров фирмы *Intel*, их можно записать так:

MOV AX, 0123h

ADD AX, 25h

Здесь **AX** – это имя *регистра* (ячейки памяти) процессора, команда **MOV** записывает в регистр новое значение, а команда **ADD** – добавляет число к содержимому ячейки. Буква «h» после числа означает, что оно записано в шестнадцатеричной системе счисления.

Чтобы переводить программу с такого языка в машинные коды, используют программы-*ассемблеры* (англ. *assembler* — рабочий-сборщик), а сам язык называется *языком ассемблера*. Этот язык – **машинно-ориентированный**, потому что он определяется набором команд конкретного процессора (ориентирован на машину).

Очевидно, что программировать на языке ассемблера тоже не очень удобно – нужно хорошо знать команды процессора, организацию памяти и т.п. Кроме того, каждый процессор имеет свою систему команд и свой язык ассемблера. Это значит, что программы на языке ассемблера *непереносимы* – программа, написанная для одного процессора, не будет работать на другом.

Людам хочется (в идеале) разговаривать с компьютером на естественном языке, не думая о том, какой процессор он использует. К сожалению, пока это невозможно. Сейчас для программирования чаще всего используют компромиссный вариант – *языки программирования высокого уровня* или *алгоритмические языки*. Это *формальные языки*, созданные специально для разработки программ. Команды строятся из слов естественного (чаще всего, английского) языка, каждая команда воспринимается однозначно в соответствии с установленными правилами.

Вспомним, что процессор может выполнить только программу, написанную в машинных кодах. Поэтому возникает задача: перевести программу, написанную на языке высокого уровня, в машинные коды. Для этого используют специальные программы-*трансляторы* (англ. *translator* — переводчик). Кроме трансляторов, в системы программирования входят и другие программы, о которых будет рассказано далее.

Системы программирования – это программные средства для создания и отладки новых программ.

6.4.2. Языки программирования

К 2010 году в мире было разработано более 8500 языков программирования⁷. Первой программисткой в мире считается Ада Лавлейс, которая в 1843 году написала программу для аналитической машины Чарльза Бэббиджа. В 1979 году в США был разработан язык программирования Ада, названный в её честь. Один из первых алгоритмических языков – *Фортран* – создан в 1957 году, однако он и сейчас применяется для научных вычислений.

Как вы уже знаете, языки программирования можно разделить на *языки низкого уровня* (машинно-ориентированные, языки ассемблера) и *языки высокого уровня* (алгоритмические языки). По области применения можно выделить несколько групп современных языков программирования:

- профессиональные языки общего назначения⁸: *Java, C, C++, C#, Visual Basic, Delphi*;
- языки для программирования Интернет-сайтов: *PHP, Javascript, Perl, ASP*;
- языки для решения задач искусственного интеллекта: *Лисп, Пролог*;
- языки для обучения программированию: *Бейсик, Паскаль, Лого, Рапира*.

6.4.3. Трансляторы

Основа любой системы программирования – транслятор.

Транслятор – это программа, которая переводит в машинные коды тексты программ, написанных на языке высокого уровня.

Существует два типа трансляторов – *интерпретаторы* и *компиляторы*.

Интерпретатор анализирует текст программы по частям. Разобрав очередной фрагмент, он немедленно выполняет описанные в нем действия и переходит к обработке следующего фрагмента. *Достоинства* интерпретаторов:

- программы *переносимы* (программа будет работать в любой системе, где установлена программа-интерпретатор);
- удобно отлаживать программу.

Есть и существенные *недостатки*:

- программу невозможно выполнить, если не установлен интерпретатор;
- программы выполняются медленно (в цикле из 100 шагов каждая строчка 100 раз «разбирается» интерпретатором);
- в тех частях программы, которые не выполнялись во время отладки, могут оставаться синтаксические ошибки.

Второй тип трансляторов – **компиляторы**. Они, в отличие от интерпретаторов, сразу переводят всю программу в машинный код и строят исполняемый файл, готовый к запуску. *Достоинства* компиляторов:

- чтобы запустить программу, не нужно устанавливать транслятор;
- поскольку программа уже переведена в машинные коды, она выполняется значительно быстрее, чем при использовании интерпретатора.

Недостатки тоже есть:

⁷ <http://hopl.murdoch.edu.au>

⁸ <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

- при любом изменении нужно ждать окончания компиляции (перевода в коды); это несколько затрудняет отладку;
- готовая программа будет выполняться только в той операционной системе, для которой она была создана⁹.

Чтобы как-то совместить достоинства интерпретаторов и компиляторов, была предложена идея компиляции программы в некоторый промежуточный исполняемый код (*псевдокод*, *P-код*), а не сразу в команды конкретного процессора. Для выполнения такого псевдокода нужна специальная среда – *виртуальная машина*, которую в принципе можно разработать для любого процессора и любой операционной системы.

Программа сначала обрабатывается *компилятором*, который строит псевдокод, а потом этот псевдокод выполняется *интерпретатором*. Таким образом,

- при компиляции в псевдокод проверяются все синтаксические ошибки, поэтому при выполнении такую проверку делать не нужно; это значительно ускоряет работу программ в сравнении с интерпретацией;
- обеспечивается *переносимость* программ – можно выполнять программу (псевдокод) на любом компьютере, где есть виртуальная машина.

Байт-код – это разновидность псевдокода, в котором команда занимает 1 байт, а далее следуют ее аргументы (или их адреса). Современные версии интерпретируемых языков *Perl*, *PHP*, *Python* используют компиляцию в байт-код для ускорения выполнения программы.

Готовые программы на *Java* распространяются в виде байт-кода, поэтому для их выполнения необходимо установить виртуальную *Java*-машину. При этом для ускорения работы часто используется *JIT*-компиляция (англ. *JIT = just-in-time* – «в это самое время»), при которой байт-код «на лету» преобразуется в команды конкретного процессора. Тогда при повторном выполнении команды трансляция уже не нужна.

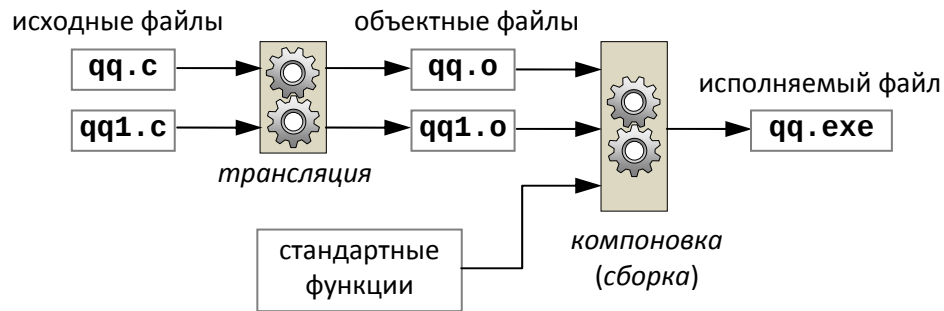
Аналогичный подход применяется в среде *.NET*, которую разработала фирма *Microsoft*. Одна из основных идей среды *.NET* – объединение программ, написанных на разных языках. В частности, разные части программы могут быть написаны на *C#*, *J#*, *VB.NET*, *Delphi.NET*, все они в конечном счете транслируются в байт-код на промежуточном языке *IL* (англ. *Intermediate Language*), который потом выполняется виртуальной машиной.

6.4.4. Состав системы программирования

Кроме трансляторов, в **состав системы программирования** обычно входят

- *компоновщик* (редактор связей, сборщик, англ. *linker*) – программа, которая собирает разные части (модули) создаваемой программы и функции из стандартных библиотек в единый исполняемый файл; вот, например, как собирается программа на языке Си, состоящая из двух модулей (исходные файлы **qq.c** и **qq1.c**):

⁹ Многие программы, разработанные для ОС *Windows*, могут быть запущены в *Linux* с помощью программы-оболочки *Wine*.



- *отладчик* (англ. *debugger*¹⁰) – программа для поиска ошибок в других программах; отладчик позволяет:
 - выполнять программу в пошаговом режиме (по одной строчке);
 - выполнять программу до строчки, где установлен курсор;
 - устанавливать точки останова (англ. *breakpoints*);
 - просматривать и изменять значения переменных в памяти.
- *профилировщик* (англ. *profiler*) – программа, позволяющая оценить время работы каждой процедуры и функции («профиль» времени выполнения программы); используется для того, чтобы выяснить, какую именно процедуру нужно оптимизировать в первую очередь.

Любая система программирования включает *библиотеки стандартных подпрограмм*. Это набор готовых процедур и функций, которые можно вызывать из своей программы. Например, в большинстве языков программирования есть стандартные функции для вычисления синуса и косинуса. Они подключаются к программе на этапе сборки, это делает компоновщик.

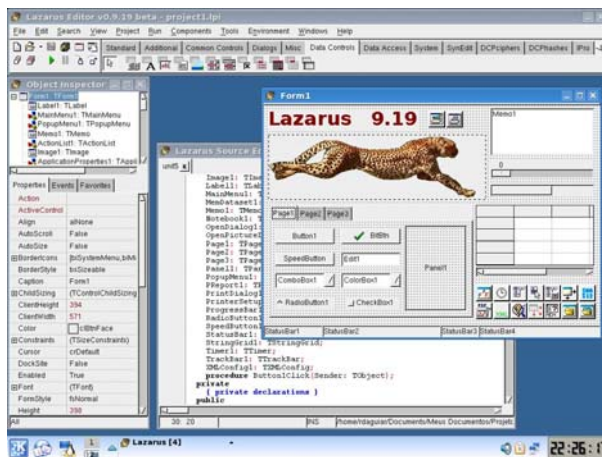
Многие программы используют одни и те же достаточно сложные системные функции (например, операции с окнами в графической среде). Если включать эти функции в код каждой программы, размеры исполняемых файлов намного увеличатся, из-за этого жесткий диск и память будут расходоваться неэффективно. Поэтому библиотеки таких функций хранятся на диске в виде отдельных файлов – *динамически подключаемых библиотек*, в системе *Linux* они имеют расширение **.so** (от англ. *shared objects* – разделяемые объекты), а в *Windows* – расширение **.dll**. Когда программа вызывает функцию из такой библиотеки, библиотека загружается в память и управление передается вызванной функции. Несколько программ могут обращаться к одной и той же копии библиотеки в памяти.

Набор стандартных структур данных и функций операционной системы, которые программисты могут использовать в прикладных программах, называется *интерфейсом прикладного программирования* (англ. *API = Application Programming Interface*). В ОС *Windows* применяется *Windows API*, а в *Unix*-подобных операционных системах – стандарт *POSIX* (англ. *Portable Operating System Interface for Unix* – переносимый интерфейс операционных систем *Unix*).

Сейчас для разработки программ чаще всего используются **интегрированные среды** (англ. *IDE = Integrated Development Environment*). В такую оболочку обычно входит текстовый редактор для набора текста программ, транслятор, компоновщик, отладчик и профилировщик.

Многие современные интегрированные среды позволяют строить интерфейс программы (расположение элементов в окне) с помощью мыши. Они называются *средами быстрой разработки приложений* (англ. *RAD = Rapid Application Development*) или средами визуального программирования. На рисунке показано окно RAD-среды *Lazarus* для программирования на объектном Паскале:

¹⁰ Согласно одной из версий, это название связано с жучком, который попал между контактов реле компьютера Mark II в 1947 году. Дословно «*debug*» – «удаление жучков».



Среди профессиональных RAD-сред нужно, в первую очередь, назвать  Microsoft Visual Studio (msdn.microsoft.com/vstudio). Её профессиональная версия – коммерческая, но все желающие могут бесплатно скачать и использовать ограниченную версию (Express) для учебных целей.

Большой популярностью пользуется также среды  Dev-C++ (wxdsn.sourceforge.net) и  Delphi (embarcadero.com). Кроссплатформенная среда  Code::Blocks (www.codeblocks.org) распространяется бесплатно, существуют версии для Windows, Mac OS X и Linux.

Контрольные вопросы

1. Что такое машинный код?
2. Зачем нужны системы программирования? Можно ли обходиться без них?
3. Что такое язык ассемблера? Почему он называется машинно-ориентированным?
4. Что такое язык программирования высокого уровня?
5. Как можно разделить языки программирования по области применения?
6. Зачем нужен транслятор?
7. Какие два типа трансляторов вы знаете? В чем их достоинства и недостатки?
8. Какие программы входят в системы программирования?
9. Зачем нужен компоновщик?
10. Что такое отладчик? Перечислите возможности отладчиков.
11. Что такое профилировщик? Зачем он нужен?
12. Что такое интегрированная среда разработки?
13. Что такое среда быстрой разработки (RAD-среда)?

6.5. Установка программ

Большинство современных программ требуется устанавливать (*инсталлировать*, от англ. *install* – установить). Это связано с тем, что

- необходимо проверить, соответствует ли компьютер требованиям (к процессору, оперативной памяти, операционной системе и т.д.), которые обязательны для работы программы;
- программы содержат множество файлов, которые должны быть записаны на диск определенным образом;
- у пользователя должна быть возможность выбора нужных ему компонентов программы (остальные не устанавливаются);
- необходимо записать некоторые файлы в каталоги операционной системы (например, при установке драйверов устройств);
- необходимо настроить режимы работы программы с учетом особенностей компьютера;

- при установке коммерческих программ необходимо вводить *ключ* (серийный номер копии программы).

Инсталляция – это установка и настройка программы на компьютере пользователя.

Пользователь получает программу в виде *дистрибутива* (установочного пакета, от англ. *distribute* – распространять). Дистрибутив – это несколько файлов на CD- или DVD-диске, или один файл (который часто можно загрузить из Интернета). В таком виде программу невозможно использовать по прямому назначению. Данные в дистрибутиве обычно сжаты, распаковка происходит во время установки. Чтобы установить или удалить ПО, пользователь должен иметь права администратора компьютера.

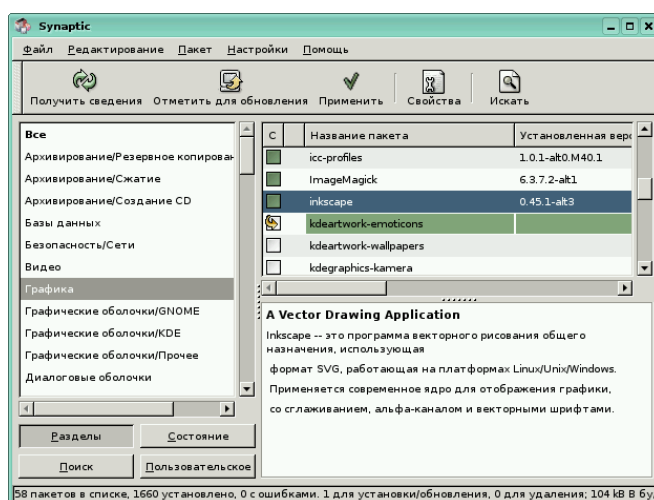
Иногда программное обеспечение может поставляться в виде исходного кода. Чтобы преобразовать его в готовую для выполнения программу, нужно использовать одну из систем программирования.

Обычно установка включает несколько этапов (некоторые из них могут отсутствовать):

- выбор компонентов программы, которые пользователь хочет установить;
- просмотр лицензионного соглашения (договора о возможности использования программы);
- ввод ключа (серийного номера) программы;
- определение каталога, в котором нужно разместить файлы программы;
- распаковка и копирование файлов на жесткий диск компьютера;
- настройка программы с помощью файлов конфигурации (или запись настроек в системный реестр в ОС *Windows*);
- создание ярлыков для запуска программы в меню и/или на *Рабочем столе*.

В операционной системе *Linux* программы чаще всего распространяются в виде *пакетов* (файлы с расширениями **.rpm** или **.deb**, в зависимости от сборки) или в исходных кодах. Для установки пакетов используются утилиты **apt-rpm** и **apt-get**. Они позволяют проверить *зависимости* пакетов: устанавливается не только указанный пакет, но и другие пакеты, необходимые ему для работы.

Начинающим пользователям *Linux* проще всего использовать графические программы для работы с пакетами (*менеджеры пакетов*), например, *Aptitude* или  *Synaptic*. В них можно мышкой отметить пакеты, которые требуется установить, обновить или удалить.



В ОС *Windows* для установки программ используется служба *Windows Installer*, которая работает с установочными пакетами – файлами в формате **.msi**. Дистрибутив также может пред-

ставлять собой программу (файл с расширением **.exe**), которая содержит все необходимые данные и при запуске «ведёт» пользователя через все этапы установки.

Программы для операционной системы *Mac OS X* распространяются в виде пакетов (файлов с расширением **.pkg**). Для работы с ними в ОС включена программа  *Installer*.

Системные администраторы, которым приходится устанавливать ПО на большое количество компьютеров, нередко используют автоматическую установку (без участия человека) или удаленную установку (через сеть).

Установка ПО – это дополнительное неудобство для пользователей. Поэтому особой популярностью пользуются *переносимые программы* (англ. *portable applications*). Их не нужно устанавливать, они могут быть просто скопированы на жесткий диск компьютера или запущены прямо с CD-, DVD- или флэш-диска. К этой группе относятся многие небольшие бесплатные программы для ОС *Windows* и *Mac OS X*.

Часто делают специальные переносимые версии «обычных» программ (которые необходимо устанавливать); их можно найти, например, на сайте *portableapps.com*. Они очень полезны для тех, кто часто работает на разных компьютерах и хочет использовать привычный набор программ. Пользователь может запустить программу со своего флэш-диска, причем все настройки (например, закладки в браузере *Opera Portable*) сохраняются в папке программы.

Существуют ознакомительные версии операционных систем, которые загружаются прямо с CD- или DVD-диска (англ. *live disc* – «живой диск») или флэш-диска. Они не меняют данные на жестком диске, а все файлы, необходимые для работы, размещаются в оперативной памяти компьютера. Многие «живые диски» позволяют работать с файлами на винчестере, поэтому с их помощью можно сохранить данные в случае отказа установленной на компьютере операционной системы. Большинство *live*-систем строится на базе *Linux*; часто с «живого диска» можно сразу же выполнить полную установку ОС на компьютер.



Контрольные вопросы

1. Что такое инсталляция? Почему она необходима для многих современных программ?
2. Что происходит во время инсталляции?
3. Что такое дистрибутив? В чем его отличие от установленной программы?
4. Что такое менеджер пакетов?
5. Как устанавливаются программы в разных операционных системах?
6. Какие методы используются для массовой установки ПО?
7. Что такое переносимая программа?
8. Как можно познакомиться с операционной системой, не устанавливая ее на жесткий диск?

6.6. Правовая охрана программ и данных

6.6.1. Авторские права

По законам большинства стран компьютерные программы и данные охраняются *авторским правом*. Это значит, что автор (или правообладатель, например, фирма, в которой работает автор) могут ограничивать распространение и использование программы.

В Конституции Российской Федерации записано, что «интеллектуальная собственность охраняется законом» (ст. 41 ч. 1). Интеллектуальная собственность – это права на результаты творческой деятельности человека. Эти права детально определены в Гражданском кодексе РФ (часть IV, «Права на результаты интеллектуальной деятельности и средства индивидуализации»).

Авторские права **распространяются** на

- программы для компьютеров (включая подготовительные материалы, а также звук, графику и видео, которые получаются с помощью программы);
- базы данных (массивы данных, специально организованные для поиска и обработки с помощью компьютеров).

Не охраняются авторским правом

- алгоритмы и языки программирования;
- идеи и принципы, лежащие в основе программ, баз данных, интерфейса;
- официальные документы.

Важно, что охраняется форма, а не содержание. Это значит, авторские права получает не тот, кто придумал *метод* решения задачи, а тот, кто написал *программу*, которая решает задачу на основе предложенного алгоритма.

Согласно российским законам об авторском праве, автор – это физическое лицо (не организация). Авторское право

- возникает «в силу создания» продукта и не требует формальной регистрации, хотя при желании автор может зарегистрировать программу в государственных органах;
- обозначается знаком ©, после которого записывается фамилия автора и год первого выпуска программы, например, © Иванов, 2008;
- действует в течение жизни и 70 лет после смерти автора;
- передается по наследству.

Автор получает **личные права**:

- право авторства (право считаться автором);
- право на имя (право выпускать программу под своим именем, псевдонимом или анонимно);
- право на неприкосновенность программы и ее названия;

и **имущественные права**: осуществлять или разрешать

- выпуск программы в свет;
- копирование в любой форме;
- распространение;
- изменение (в т.ч. перевод на другой язык).

Серьезные нарушения авторских прав могут попасть под действие Уголовного кодекса РФ (ст. 146, «Нарушение авторских и смежных прав»). Уголовная ответственность наступает при крупно ущербе (более 50 000 руб.). Присвоение авторства (*плагиат*) наказывается лишением свободы на срок до 6 месяцев. В случаях незаконного использования, а также приобретения и хранения объектов авторского права (например, дисков с нелегальными программами) в целях сбыта срок лишения свободы может достигать 5 лет (при особо крупном ущербе).

6.6.2. Типы лицензий на использование ПО

Право на использование программы дает документ (договор), который называют **лицензией** (лат. *litentia*) или **лицензионным соглашением**. Это соглашение между правообладателем и пользователем, где четко определены права и обязанности сторон. Как правило, в соответствии с лицензией пользователь без дополнительного разрешения автора может

- установить программу на 1 компьютер (или так, как указано в договоре);
- вносить изменения, необходимые для работы программы на компьютере пользователя; исправлять явные ошибки;
- изготовить копию, чтобы можно было восстановить программу в случае сбоя;
- передать программу другому лицу вместе с лицензией.

Программы, которые получены и используются в соответствии с законом, называют *лицензионными*. Если же при создании копии были нарушены авторские права, ее называют *контрафактной* или *пиратской*.

По **типу лицензий** можно разделить ПО на 4 типа:

- коммерческое;
- условно-бесплатное (англ. *shareware*);
- бесплатное (англ. *freeware*);
- свободное ПО (англ. *open source* – ПО с открытым кодом).

Значительная часть ПО является **коммерческим**, поскольку создается с целью получения прибыли путем продажи экземпляров. За каждую копию коммерческого ПО нужно платить (покупать лицензию); исходный код, как правило, не распространяется. Обычно фирмы предусматривают скидки при закупке большого количества лицензий (лицензии на организацию) и скидки для образовательных учреждений. Зарегистрированные пользователи программ имеют право на бесплатную техническую поддержку – консультации по телефону или электронной почте. Типичный пример коммерческого ПО – операционная система *Windows*.

Часто разработчики дают возможность бесплатно скачать пробную (англ. *trial*) версию программы из Интернета и попробовать, как она работает (англ. «*try before you buy*» – «попробуй, прежде чем купить»). Такие программы называют **условно-бесплатными** (англ. *shareware*). Пробные версии всегда имеют какие-то ограничения, например:

- ограниченный срок работы (обычно 30 дней);
- ограниченное количество запусков;
- ограничение функций (например, невозможно сохранить результаты на диск);
- встроенный рекламный блок;
- всплывающие сообщения с призывом заплатить автору деньги за программу.

Обычно в лицензионном соглашении указывается, что пробная версия не может быть использована для коммерческих целей и профессиональной работы.

К условно-бесплатным можно отнести многие популярные программы, у которых есть пробные версии. Например, на ноутбуки часто устанавливается пробная версия пакета *Microsoft Office*, которую можно бесплатно использовать 60 дней. В Интернете можно скачать пробные версии векторного редактора *Corel Draw*, почтовой программы *TheBat*, всех программ фирмы *Adobe*.

Для решения большинства задач можно найти **бесплатные** программы (англ. *freeware*). Это значит, что лицензионное соглашение не требует никаких выплат автору. Бесплатные программы можно свободно скачать из Интернета и использовать, однако чаще всего коммерческое использование и изменение запрещается, исходные коды не распространяются. Иногда фирмы бесплатно распространяют ограниченные версии коммерческих программ. К бесплатным программам относятся браузеры *Opera*, *Chrome* и *Safari*, программа для записи CD и DVD-дисков *CDBurber XP*, множество небольших утилит.

Наибольшие возможности предоставляет **свободное ПО** (англ. *open source* – открытый исходный код). Авторы свободных программ передают пользователю не только исполняемую программу, но и ее исходный код, и предоставляют:

- право использовать программу в любых целях;
- право изучать исходный код и изменять его для своих целей;
- право свободно распространять программу;
- право улучшать программу и распространять измененные версии на тех же условиях.

Программное обеспечение, которое не удовлетворяет этим критериям, называется **собственническим** или **проприетарным** (англ. *proprietary* – частное).

В первые годы на свободное ПО никакие документы не оформлялись, но возникла необходимость защищать права авторов юридически. Поэтому сейчас свободное ПО чаще всего распространяется под лицензией *GPL* (англ. *General Public Licence* – «генеральная общественная лицензия»). В ней перечислены приведенные выше права и установлено одно ограничение: разрешается распространять измененную версию только как свободное ПО (запрещается делать его несвободным).

К свободному программному обеспечению относится операционная система *Linux*, браузер *Mozilla Firefox*, почтовая программа *Mozilla Thunderbird*, графические редакторы *Gimp* и *Inkscape*, архиватор **7ZIP** (www.7-zip.org), среда для быстрой разработки программ *Code::Blocks* и многие другие программы.

Как ни странно, свободное ПО может приносить прибыль. Например, некоторые фирмы оказывают платную техническую поддержку по развертыванию и настройке системы *Linux*. Второй вариант – предоставление коммерческой лицензии в том случае, если открытый исходный код используется в коммерческих программах.



Контрольные вопросы

1. Что обозначает термин «авторские права»?
2. Что такое интеллектуальная собственность?
3. Какие законы РФ регулируют вопросы, связанные с авторскими правами?
4. На какие творческие результаты распространяются авторские права?
5. Что не охраняется авторским правом?
6. Что значит положение «охраняется форма, а не содержание»?
7. Нужно ли регистрировать авторское право?
8. Кто может быть правообладателем согласно российским законам?
9. Как обозначается авторское право в документе?
10. Как действует авторское право после смерти автора?
11. Какие личные и имущественные права имеет автор?
12. Какие наказания предусмотрены за нарушение авторских прав?
13. Василий Пупкин разработал программу *SuperPuper* в 2010 году. Как он должен правильно обозначить в тексте программы своё авторское право?
14. Что такое лицензионное соглашение?
15. Что обычно может делать пользователь программы, не спрашивая дополнительного разрешения автора?
16. Что такое лицензионная программа?
17. Что такое контрафактная программа?
18. Какие типы лицензий на ПО сейчас существуют?
19. Что такое условно-бесплатная программа? Какие ограничения она может иметь?
20. Какие программы относятся к бесплатным (*freeware*)?
21. Что такое свободное ПО? Почему оно распространяется по лицензии?
22. Какие свободы предоставляет пользователю лицензия *GPL*?
23. Какие ограничения предусматривает лицензия *GPL*?
24. Что такое собственническое (проприетарное) ПО? Чем оно отличается от коммерческого?
25. Как можно сделать разработку свободного ПО коммерчески выгодным?
26. Какие типы ПО можно законно загружать из Интернета?
27. Можно ли, не спрашивая автора (правообладателя)
 - а) скопировать картинку с веб-страницы на свой компьютер?

- б) послать скопированную картинку другу?
- в) разместить на своем сайте отсканированную книгу?
- г) привести на сайте цитату из книги с указанием источника?
- д) разместить на своем сайта картинку с другого сайта?

(Ответ: можно а, б, г)

28. Можно ли размещать в Интернете, не спрашивая авторов:

- а) произведения А.С. Пушкина?
- б) записи популярных исполнителей?
- в) документы, принятые Государственной Думой?
- г) описание алгоритма решения квадратного уравнения?
- д) базу данных сотовых телефонов?

(Ответ: можно а, в, г)