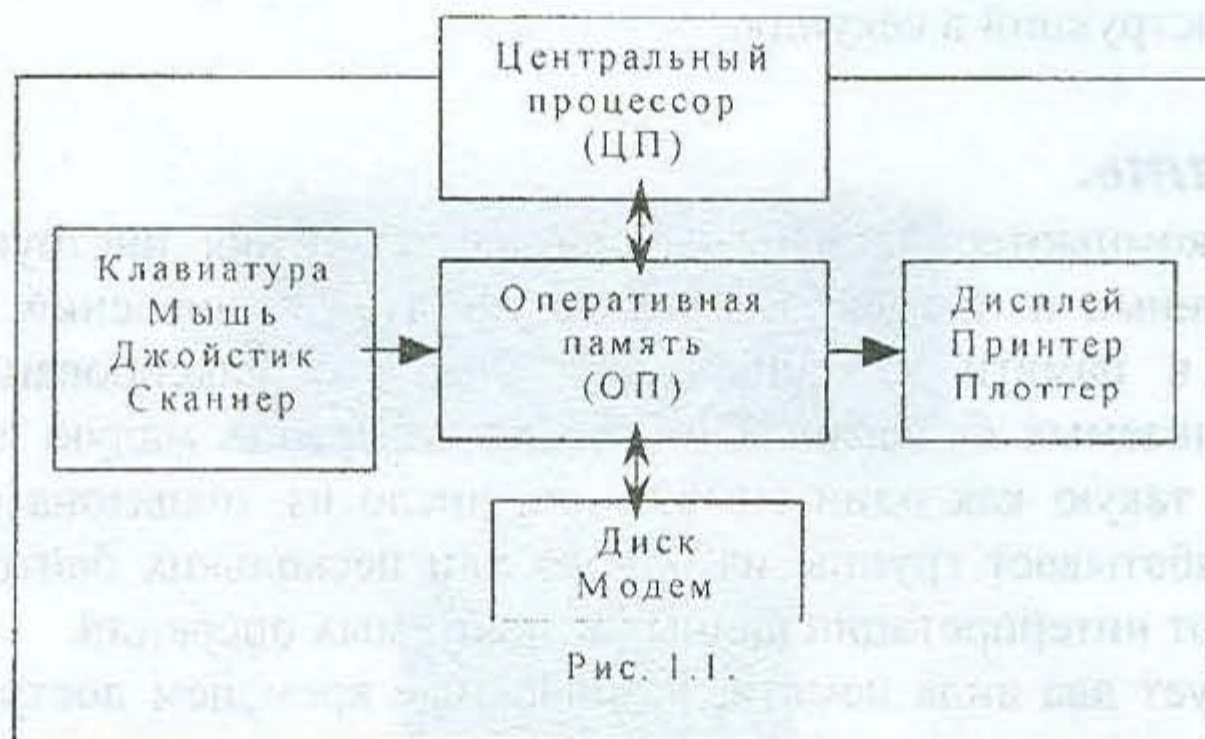


1. Из чего состоит компьютер.

Основные составные компоненты компьютера приведены на рис. 1.1. Двойные стрелочки указывают на то, что обмен информацией двусторонний, а связи с одинарными стрелочками односторонние, т.е. данные либо только поступают с этих устройств, либо только выводятся на них.



1.1. Центральный процессор.

Центральный процессор (ЦП) выполняет все операции по обработке данных на компьютере. Данные для обработки поступают в процессор через системную шину из главной (или оперативной) памяти. Результаты вычислений передаются обратно в оперативную память через ту же системную шину. Кроме вычислений, процессор контролирует и координирует функционирование элементов компьютера.

ЦП состоит из двух основных компонентов:

1. *Контрольный элемент* — осуществляет выборку инструкций из оперативной памяти и последующее выполнение этих инструкций. Кроме этого, ЦП контролирует устройства ввода/вывода и передачу данных арифметическо-логическому устройству для обработки.

2. *Арифметическо-логическое устройство (АЛУ)* — реализует арифметические операции над целыми и действительными (числа с

десятичной точкой) операндами. Кроме этого, он реализует простое логическое сравнение операнд — больше, меньше, равно.

Таким образом, ЦП реализует только простые арифметические операции и сравнение результатов вычислений с другими величинами, а также выбирает следующую инструкцию для обработки. Все остальные возможности, которые нам предоставляет компьютер, обеспечиваются средствами программирования.

Современные ЦП очень быстры. Они выполняют десятки миллионов инструкций в секунду.

1.2. Память.

Память компьютера предназначена для хранения инструкций программ, данных и промежуточных результатов вычислений. Вся информация в памяти закодирована в ячейках фиксированного размера, называемых *байтами*. Байт может содержать малую часть информации, такую как один символ или число из диапазона 0 — 255. ЦП обрабатывает группы из одного или нескольких байтов, в зависимости от интерпретации данных и требуемых операций.

Существует два вида памяти, различаемые временем доступа к хранимой информации, количеством байтов, одновременно доступных для одной операции, и полным количеством байтов, которое предназначено для хранения информации.

Главная, т.н. оперативная память (ОП) компьютера — это рабочая память центрального процессора с быстрым доступом и ограниченным количеством передаваемых байтов. *Внешняя память (ВП)* служит для долгосрочного хранения информации. Данные передаются из внешней памяти в оперативную, где ЦП может их обработать. Скорость доступа к внешней памяти гораздо ниже, но ее объем измеряется тысячами байтов.

1.2.1. Оперативная память.

Оперативная память компьютера (RAM — Random Access Memory — память случайного доступа) konstruирована из интегрированных схем и нуждается в электрическом питании для обработки информации. Когда питание отключается, информация теряется автоматически. Оперативная память непосредственно

доступна только для центрального процессора. Время доступа для чтения или записи любого байта не зависит от его месторасположения в памяти и приблизительно равняется 50 НаноСекундам (10^{-9} секунд). Эта величина соответствует скорости доступа ЦП к данным. Оперативная память, относительно внешней памяти, более дорогая и, следовательно, имеет меньшую вместимость, хотя цена ее уменьшается с каждым годом. Многие персональные компьютеры обладают 32 МегаБайтами (1 МБ — 10^6 Байт) памяти, тогда как на коммерческих рабочих станциях она достигает 64 МБ.

ЦП передает данные из/в оперативную память обычно группами в два, четыре или восемь байтов, даже если для предпринимаемой операции требуется только 1 байт памяти.

ОП — это черновик центрального процессора, где происходят все операции, им выполняемые, а также рабочее место, где временно хранится обрабатываемая информация. Фактически ОП — это связывающее звено между центральным процессором (и компьютером в целом) и пользователем.

Кроме RAM, на плате оперативной памяти расположено т.н. *Постоянное Запоминающее Устройство*, ПЗУ (*ROM—Read Only Memory*). Этот тип памяти доступен только для чтения и на ней записаны основные системные программы (см. ниже).

1.2.2. Внешняя память.

Внешняя память, которую иногда называют *вспомогательной памятью*, позволяет постоянное хранение большого количества данных. Это обеспечивается определенными методами магнетической записи на *магнитных дисках (накопителях)* или *лентах*. Недавно стали использоваться оптические методы, основанные на принципе гравировки поверхности дисков (*CD-ROM*) лазерными лучами, хотя они более дорогостоящие, чем магнитные.

Вместимость внешней памяти высокая и в настоящее время измеряется сотнями Мега- и даже ГигаБайтами (1 ГБ — 10^9 Байтов).

Внешняя память обладает очень важным достоинством — информация сохраняется даже после отключения компьютера.

Самая распространенная форма внешней памяти — *жесткий диск*, или *винчестер*, который установлен в компьютере постоянно и обладает вместимостью несколько сотен МБ. Жесткий диск является плоской, покрытой окисью круглой пластинкой, которая вращается непрерывно. Вдобавок к этому многие компьютеры имеют *дисководы* для *съемных дискет*, на которых можно хранить копии важнейших файлов. *Гибкая дискета*, самая распространенная, имеет очень ограниченную вместимость (1.2 МБ, 1.44 МБ). Больше количество информации можно хранить на *магнитных лентах*. Недавно стали использоваться съемные *оптические диски* (230 МБ, 640 МБ).

Информация на дисках записывается с помощью намагниченных точек на окисном слое, каждая из которых может представлять два значения 1 или 0 (т.е. намагничено или нет). Расположение этих точек заранее не предопределено и для записи информации диск предварительно нужно *разметить, форматировать*. Все диски, как гибкие, так и жесткие, организованы одинаково. При форматировании поверхность диска разделяется на ряд концентрических колец, называемых *дорожками* или *треками*, а треки делятся радиально на *сектора*. Например, 5 1/4-дюймовая дискета (емкостью 1.2 МБ) имеет 80 треков. В системе MS DOS каждый трек разбит на 15 секторов. Размер сектора — 512 байтов, и

512 байтов — 9 секторов — 80 треков — 2 стороны
дает в итоге емкость дискеты 1200 КБ. Все типы дисков в MS DOS используют размер сектора 512 байтов.

Дисковые сектора определяются магнитной информацией, которую записывает утилита форматизации диска. Каждый сектор имеет идентификационный номер 1—8, 1—9 или 1—15 и т.д. в зависимости от емкости диска. Треки не маркируются, а определяются механически по смещению головки чтения/записи от внешнего края диска. Дорожки нумеруются от 0 до 39 или больше. Однако операционная система рассматривает все сектора диска как одну цепь, которая нумеруется подряд, начиная с нуля, поэтому каждый сектор имеет свой *логический номер*.

При доступе к информации *головка* чтения/записи устанавливается над подходящим треком и осуществляет чтение или запись с/на трек. Это означает, что перед доступом или модификацией

данных головка чтения/записи должна быть установлена над правильным треком. Время, требуемое для этого, называется временем поиска и измеряется в Милли Секундах. Некоторая задержка возникает при ожидании подходящего сектора трека, который вращается под головкой. Эта задержка гораздо меньше, чем время поиска. Как только подходящий сектор трека окажется под головкой, соответствующие байты информации могут быть переданы в оперативную память со скоростью несколько МБ в секунду. Это различие между скоростью доступа к первому байту и к последующим на том же треке указывает, что неэкономно передавать байты в малых количествах. Обычно передаются блоки из несколько сотен байтов или более. Отметим, что время доступа к информации, хранимой на внешней памяти, зависит от ее локации.

Фиксированные диски имеют некоторые специальные характеристики. Часто они состоят из двух или более параллельных пластин, у каждой из которых есть две головки, чтобы читать обе их стороны. Все треки, расположенные на одном расстоянии от центра, вместе называются *цилиндром*. Поскольку головки всех дисков двигаются тандемом, то экономия перемещений достигается, если заполнять все треки одного цилиндра, прежде чем переходить к следующему.

Несколько секторов на внешнем ободе дискеты зарезервированы для специальных нужд. Остальные доступны на основе правила «первый подошел — первого обслужат», т.е. по мере заполнения диска данными сектора постепенно заполняются по направлению от края к центру диска. При уничтожении *файла* сектора освобождаются и со временем свободные области становятся разбросанными по диску, разбивая новые файлы и замедляя доступ к ним для чтения и записи. Более подробно об организации информации файлами рассказывается ниже в этой же главе.

Жесткий диск содержит все программное обеспечение, требуемое для запуска компьютера, начиная от операционной системы до пакетов прикладных программ. Все данные и программы пользователя также хранятся на жестком диске. Фактически, он является библиотекой компьютера.

Важно отметить, что ЦП имеет доступ только к оперативной памяти. Чтобы обработать данные, находящиеся на внешней памяти, ЦП сначала должен перенести их в оперативную память.

1.3. Устройства ввода/вывода.

Как данные, так и текст инструкций для их обработки должны быть как-то переданы компьютеру. Кроме этого, результаты вычислений также должны быть как-то возвращены пользователю. Это взаимодействие обеспечивается т.н. периферийными устройствами ввода/вывода.

Самое распространенное устройство для ввода — *клавиатура и мышь*. Клавиатура позволяет вводить текстовую информацию, тогда как мышь позволяет выделить на экране компьютера нужную область с помощью перемещения курсора и нажатия соответствующей клавиши мыши. Кроме этого, с помощью мыши можно выбрать нужные пункты в меню на экране и т.д. На многих современных компьютерах такой способ является основным методом коммуникации. Альтернативными устройствами являются *шаровой манипулятор, световое перо и сенсорный экран*.

Самое распространенное устройство для вывода — *монитор, экран дисплея* — электронная трубка на катодных лучах, которая может отображать на экране текстовую и графическую информацию. Кроме того, с помощью дисплея можно контролировать вводимый с клавиатуры текст, проверять, корректировать, изменять и т.д. Для документальной (печатной) копии выводимых результатов используются т.н. *принтеры* различного вида.

Другими периферийными устройствами являются: *сканнер* (для ввода фотокопий печатной страницы), *модем* (для коммуникации с другими компьютерами в сети), *джойстик* (для перемещения и управления курсором на экране) и др.

Кроме того, данные в компьютер могут поступать с дисков, а также выводиться на них. Такую информацию можно хранить долгосрочно, а также корректировать ее, изменять, использовать многократно, и в любое время выводить на печать. Однако доступ к дискам осуществляется процессором не непосредственно, а через оперативную память.

1.4. Системная шина.

Все коммуникации между главными компонентами осуществляются с помощью *системной шины*. Шина — это просто кабель, который передает информационные сигналы, представляющие данные, с одного места локации в другое. Она может быть специфической для конкретного компьютера или же стандартной. Последний случай дает преимущество легко обновлять компьютер, добавляя только нужные детали.

Когда данные посылаются из памяти на принтер, они проходят через системную шину. Контрольные сигналы для активации принтера также посылаются центральным процессором через системную шину.



Рис. 1.2.

1.5. Операционные системы.

Структурной частью компьютера можно считать также его программное обеспечение, и, в первую очередь, *Операционную Систему (ОС)*. Программы существуют двух видов — *системные* и *прикладные*. Системные программы дают возможность и облегчают использование компьютера, а прикладные программы выполняют задачи пользователя.

Некоторые необходимые системные программы встроены в машину и они записаны ПЗУ (ROM). Эти программы, осуществляющие контроль, помощь и необходимые услуги прикладным

программам, называются *Базовой Системой Ввода/Вывода (BIOS—Basic Input Output System)*. Другие системные программы создаются на основе программ ROM—BIOS и обеспечивают более высокий уровень сервисной поддержки.

Операционная система компьютера — это большая программа, которая обеспечивает функционирование компьютерной системы. Она освобождает пользователя от непосредственного соприкосновения, обращения к техническим элементам компьютера, к аппаратуре, позволяет управлять оперативной памятью, дисками, запускать другие прикладные программы на выполнение, предоставляя программисту широкие возможности с помощью команд ОС.

ОС включает в себя следующие основные структурные компоненты:

- а. *Файловая система;*
- б. *Драйверы* внешних устройств;
- в. Командный язык.

На однопользовательном компьютере функции ОС заключаются в следующем:

1. Обеспечить интерфейс диалога компьютер — пользователь. Интерфейс может быть текстовориентированным, когда пользователь набирает команды в ответ на запросы компьютера, или графическим (например, операционная система Windows, Macintosh), управляемым мышью.

2. Контроль периферийными устройствами, например, клавиатурой, видео-монитором, принтером и др. с помощью специальных программ, которые называются *драйверами*.

3. Организует файлы пользователя, храня пути к их местоположениям на дисках, обновляя их после модификации и др. ОС обеспечивает пользователя *редактором*, который позволяет обрабатывать содержимое файлов.

4. Предоставляет системные средства, например, компиляторы для трансляции с языков высокого уровня на внутренний машинный язык компьютера.

Из-за несоответствия между скоростями устройств ввода/вывода (а также ввода данных пользователем) и центрального

процессора, много современных операционных систем позволяют работать в *многозадачном* режиме. Таким образом, пока компьютер останавливается, ожидая входные данные для одной программы, ОС активизирует другую программу до тех пор, пока она, в свою очередь, не приостановится в состоянии ожидания. Многозадачный режим может функционировать на автономном компьютере (например, операционная система Windows 95 на персональных компьютерах PC) и позволяет пользователю одновременно использовать несколько различных программ. Например, пока на фоне выполняется программа обработки большого количества данных или идет поиск информации по Интернету, можно набирать текст в редакторе Word.

В настоящее время общепринято объединять компьютеры в *сети*. Сети могут состоять из многих «немых» терминалов, персональных компьютеров и рабочих станций, соединенных вместе с несколькими более мощными компьютерами, которые предоставляют сети более мощные ресурсы для вычислений и памяти. Это обстоятельство дает возможность многим пользователям иметь одновременный доступ к общим ресурсам и *базам данных, электронной почте (electronic mail — e-mail)* и др.

Сети могут быть локальными в здании или ограниченными на малой местности (*Local Area Network — LAN*), или могут соединять несколько индивидуальных сетей страны или всего мира (*Wide Area Network — WAN*).

Частной формой ОС сети является операционная система с *разделением времени*. Много современных мощных компьютеров подключены вместе, чтобы обслуживать многих пользователей одновременно с помощью системы разделения времени. Каждый пользователь имеет прямой доступ к мощному центральному компьютеру, используя видео-монитор (*Visual Display Unit — VDU*) с клавиатурой и мышью для ввода информации и экраном — для обратной связи. На больших компьютерных системах одновременно могут работать несколько сотен пользователей. Между моментом ввода запроса пользователя и моментом получения ответа проходит буквально несколько секунд. ОС обслуживает все подключенные терминалы, по очереди выделяя каждому по надобности свою долю времени центрального процессора. Этот процесс продолжается

непрерывно, так что каждая программа получает столько времени ЦП, сколько нужно для вычислений, и по завершении исключается из списка ожидающих свою очередь программ.

В многопользовательных системах ОС может выполнять дополнительно следующие задачи:

1. Подтверждать законность прав пользователя на доступ к системе.
2. Распределять память и время обработки для отдельных программ.
3. Обеспечивать надежность сохранения файлов пользователя и выполнения их программ.

В системах с разделением времени мощность обработки обеспечивается центральной машиной. Пользователи обращаются к ней с *не-интеллектуальных терминалов*, которые не способны выполнять никаких задач самостоятельно. Появление мощных рабочих станций привело к распределению мощности компьютера по сети. *Распределенная* компьютерная система состоит из центрального процессора с большой дисковой памятью и мощных средств ввода/вывода, подключенных к сети машин, каждая из которых имеет собственный процессор и память.

Центральный процессор (или *сервер*) обеспечивает сохранение всех системных файлов и файлов пользователей. Каждый вычислительный узел в сети загружает любые файлы и требуемые системные средства, а затем сервер осуществляет все вычисления. Любые изменения в файлах или новых файлов сеть посылает серверу. На сервере память выделяется также для системных файлов. Все файлы организованы *каталогами*, облегчая операционной системе доступ к ним (см. ниже).

1.6. Файловая система.

Важнейшей проблемой в процессе работы с компьютером является чтение, запись и хранение данных на дисках. Чтобы идентифицировать диск, с которого в данный момент производится чтение информации или на который должна осуществиться запись данных, ОС использует т.н. *логические имена* — буквы латинского

алфавита. Как правило, первые две буквы алфавита, «А» и «В», предназначены для идентификации дискет. Логические имена, начиная с буквы «С», используются для обозначения жесткого диска, который, в свою очередь, может быть разбит на несколько частей, в том числе и на одну единственную часть. В последнем случае жесткому диску приписывается логическое имя «С». Если же жесткий диск разбит на несколько частей, тогда названия их логических имен, соответственно, будут: «С», «D», «Е» и т.д., следуя латинскому алфавиту.

Информация на внешней памяти организована в виде *файлов (File)*. Файл — это поименованный участок дискового пространства, дисковой памяти. Каждый файл может содержать, например, текст документа, компьютерную программу, набор экспериментальных данных, закодированное изображение и др. Так же, как и в оперативной памяти, информация представляется в виде совокупности байтов, каждый из которых обладает адресом локации и содержимым.

При размещении файлов ОС использует иерархическую, т.н. древовидную структуру. Такую организацию хранимой информации называют файловой системой. Древовидная структура заключается в следующем: все содержимое диска, т.е. записанную на ней информацию, можно просмотреть в оглавлении, которое называется *корневым каталогом (Root Directory)*. В последнее время для каталога используется другое название — *фолдер (folder)*, заимствованное от системы Macintosh.

Корневой каталог содержит информацию о файлах, а также может содержать другие каталоги, которые называются *подкаталогами (Sub Directory)*. Они находятся на более низком древовидном уровне. Подкаталоги, в свою очередь, могут содержать новые файлы, а также новые подкаталоги — второго уровня. Такое разветвление может продолжаться дальше, образуя подкаталоги все более низкого уровня. В итоге файлы располагаются на диске иерархически.

Каталоги высокого уровня относительно каталогов более низкого уровня иногда называют *родительскими (Up Directory, Parent Directory)*, каталоги низкого уровня относительно каталогов более высокого уровня — *дочерними (Child Directory)*.

Например, если лектор хочет составить курсы лекций по нескольким дисциплинам, то удобнее всего сохранить файлы в одном общем каталоге, а внутри него файлы, содержащие тексты лекций к отдельным предметам, хранить по отдельным подкаталогам. «Кабинет файлов» содержит всю информацию о курсах, каждый «ящик» (подкаталог) может хранить подразделения отдельных курсов, а каждая «папка» (файл) в ящике будет содержать текст отдельной лекции.

Отметим, что каждый файл, т.е. каждый поименованный участок диска, может фигурировать только в одном из каталогов.

Таким образом, в каждом каталоге зарегистрированы как файлы, так и подкаталоги на этом уровне. Содержимое каталога доступно пользователю с помощью интерфейса ОС, который позволяет просматривать каталог, редактировать, переименовывать, копировать, перемещать или удалять зарегистрированные в нем файлы.

В каталоге каждому файлу или подкаталогу соответствует название, которое может состоять из двух частей: непосредственно имени файла/каталога и т.н. *расширения*. Расширение разделяется от имени точкой, состоит максимум из трех букв и указывает на тип файла, т.е. идентифицирует, является ли файл запускаемой программой, текстом, документом какой-либо прикладной программы и т.д. В операционной системе DOS (Disk Operational System) имя файла может содержать максимум восемь символов.

Имя файла/каталога можно формировать с помощью букв латинского алфавита, цифр и знака подчеркивания «_». Не рекомендуется использовать другие символы, включая пробел « », а некоторые из них запрещены вовсе – «.», «\», «*», «?», «:», так как они имеют специальное функциональное назначение и могут создать проблемы во время обработки файлов в процессе выполнения различных прикладных программ.

Название файла, как и подкаталога, в пределах одного каталога уникально, хотя файлы с одинаковыми именами могут встречаться и в других каталогах. Чтобы иметь возможность зафиксировать именно нужный в данный момент файл, используется *полное имя* файла, или *путь (Path)*. Путь реализуется как строковое выражение, где указываются все каталоги, начиная от корневого вплоть до того

подкаталога, в котором расположен файл. При образовании пути используются символы-разделители: «:\» и «\». Например, C:\EDITORS\CW\cw.exe

Такая запись предполагает, что файл «cw.exe» (это текстовый редактор «времен динозавров») располагается в подкаталоге «CW», который в свою очередь расположен в подкаталоге «EDITORS», сам подкаталог «EDITORS» находится на диске «C» в корневом каталоге. На корневой каталог указывает разделитель «:\», следующий за логическим именем диска, а подкаталоги более низкого уровня разделяются друг от друга символом слэш — «\».

Понятие файла может быть обобщено на любой источник или потребитель информации в компьютере, например, коммуникационный канал (COM1, COM2, COM3, COM4), принтер (LST1, LST2, LST3), дисплей, клавиатура (CON[sole]) и др. Такая трактовка создает удобства для организации взаимодействия программ и обмена информацией с внешними устройствами.

Для управления дисковым пространством, занятым файлами, а также для хранения информации о свободных секторах, используется т.н. *таблица размещения файлов (FAT - File Allocation Table)*, физически расположенная сразу после т.н. *сектора начальной загрузки* (это небольшая программа, позволяющая компьютеру считать с диска остальные части ОС и расположенная в секторах с наименьшими логическими номерами). Из соображений безопасности на всех дисках хранятся две копии FAT. Число секторов, занимаемых FAT, определяется размером и типом диска. Как мы отметили выше, файлы распределены по секторам (512 байтные участки трека). Если в таблице размещения файлов хранить адреса всех секторов, его размер может стать слишком большим. Поэтому ОС рассматривает не отдельные сектора, а их смежные совокупности, называемые *кластерами*. Таким образом, в таблице хранится последовательность адресов всех кластеров, занятых файлами, которые физически могут быть разбросаны по всему диску.

При большом размере кластера напрасно расходуется дисковое пространство, но когда большие диски имеют малый размер кластера, то таблица размещения файлов становится слишком большой.

После таблицы размещения файлов располагается корневой каталог, который содержит список файлов и ссылок на подкаталоги, а также указывает, в каком месте диска они начинаются.

1.7. Внутреннее представление информации.

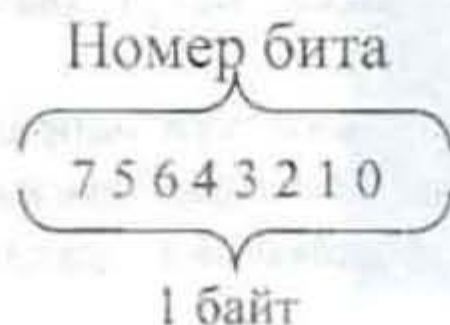
1.7.1. Единицы информации.

В основе работы компьютера лежит т.н. *двоичное представление* информации соответственно простому двухпозиционному принципу функционирования электрических устройств. Говоря грубо, в зависимости от того, проходит или нет ток в проводе, поверхность может быть намагничена или нет, что можно представить двумя цифрами 1 и 0.

Базовым квантом информации, наименьшим и основным, является *бит*, от слова *binary digit* (двоичная цифра). Он может принимать только значения 0 или 1, что может означать «выключено» или «включено», «нет» или «да».

Биты позволяют построить модель информации, с которой может работать компьютер. Сами по себе отдельные биты не представляют собой интерес, только когда рассматривается их определенная последовательность, можно говорить об информации, о данных и о способах их обработки. Поэтому обычно рассматриваются совокупности битов как единое целое.

Наименьшей единицей информации, с которой приходится общаться компьютеру, является *байт*, т.е. байт — это основная единица компьютерных данных — 8 битов, рассматриваемых как одно целое. Компьютер может работать как с совокупностями байтов, так и с отдельными битами внутри байта. В этом случае, для идентификации, их нумеруют справа налево (начиная с младшего бита) от 0 до 7:



Для одного байта (т.е. 8 различных битов) с учетом, что каждый бит может принимать значение 0 или 1, число всех возможных комбинаций будет $2^8 = 256$. С помощью одного байта можно представить либо целое число из диапазона $[0, 255]$, либо один символ. Большие числа, а также инструкции компьютера и строки символов занимают несколько смежных байтов.

Байты – это необработанные данные, которые могут использоваться самым разнообразным образом, в частности, для представления как чисел, так и текстов, в зависимости от той работы, которая на данном этапе выполняется на компьютере. Если мы работаем с числами, то байты воспринимаются компьютером как числа, и сочетания битов внутри байта получают числовую интерпретацию. Но когда мы работаем с текстовой информацией, байты интерпретируются как символы, причем каждый байт соответствует одному символу текста. Сочетания битов не уникальны и сами по себе не несут в себе того значения, какое мы в них вкладываем. Например, «65» может быть интерпретировано как число 65, или же буква «А».

Иногда рассматривают совокупность четырех битов, которые называются *ниблами (nibble)*.

Следующая единица информации – это *машинное слово*, представляет собой сочетание двух байтов, т.е. 16 битов.

Так как объем информации, объем дисковой памяти чаще всего значительно превышает размер в 1 байт, то для измерения информации памяти используются общепринятые меры – *Килобайт, Мегабайт, Гигабайт*:

Килобайт (К)	← →	1024 Б (2^{10} Б).
Мегабайт (МБ)	← →	1 048 576 Б (2^{20} Б).
Гигабайт (ГБ)	← →	1 073 741 824 Б (2^{30} Б).

1.7.2. Системы счисления.

Команды и данные передаются центральному процессору в двоичных кодах на т.н. машинном языке. Кроме того, программные сообщения выдаются компьютером также в двоичном

представлении. Следовательно, чтобы лучше понимать машинный язык, целесообразно иметь более подробное представление о системах счисления. В основном рассмотрим три системы: десятичную (*decimal*), двоичную (*binary*) и шестнадцатеричную (*hexadecimal*).

В системах счисления используется система т.н. позиционной нумерации, которая позволяет формировать все числа только с помощью ограниченного количества символов. Именно это количество определяет т.н. *базис* системы.

Арифметика человека использует десятичную систему счисления, т.е. систему, где базовыми являются 10 упорядоченных цифр 0—9, причем, после «9», последней наибольшей цифры, следующие по порядку числа составляются с помощью комбинации наименьших литералов, т.е. «0», «1», затем «2», «3» и т.д. по закону возрастания порядка. Последовательные цифры, начиная справа налево, представляют число с соответствующим весом (разрядом позиции), которая выражается степенью от 10, т.е. основы. Таким образом, 415 в десятичной системе:

$$415 = 5 \cdot 10^0 + 1 \cdot 10^1 + 4 \cdot 10^2$$

это сумма четырех сотен, одной десятки и пяти единиц. Аналогично, двоичная система образует все числа из «0» и «1», и степеней от двух. 110 в двоичной системе представляет величину:

$$110 \leftrightarrow 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2,$$

что в десятичной системе соответствует числу 6. Чтобы указать, в каком представлении записано число, применяют соответствующую индексацию, например,

$$110_2 = 6_{10}.$$

Вся информация внутри компьютера и на внешней памяти представляется в двоичной системе счисления. Оперативную память можно представить себе в виде последовательности пронумерованных байтов, начиная от 0, 1, 2, ... и выше. Каждый

байт, содержащий некоторую комбинацию из восьми битов и снабженный номером, или *адресом*, может быть доступным для центрального процессора для обработки. Например, рассмотрим следующий фрагмент памяти:

Адрес	Содержимое
3168	10110111
3167	01000111
3166	01010101

Байт с адресом 3167 содержит двоичную комбинацию 01000111. В зависимости от обстоятельств, ЦП может интерпретировать его как инструкцию, как число (или часть числа), или же как символ. Данная комбинация может представлять символ «G» из ASCII-таблицы, или же десятичное число 71 (*ASCII = American Standard Code for Information Interchange* — стандартный американский код для обмена информацией, которая используется почти универсально по всему миру для представления текстовых данных).

Таким образом, очень важно четко разграничить адрес локации памяти и содержимое той локации.

Чаше для представления машинной информации, чтобы избежать громоздких записей, что неизбежно увеличивает возможность появления ошибок, вместо двоичной используется шестнадцатеричная система счисления. В этой системе основными являются 16 символов: цифры 0—9 и шесть букв латинского алфавита A, B, C, D, E и F, причем соблюдается соответствие:

Десятичное	Шестнадцатеричное
0	0
1	1
2	2
3	3
4	4
5	5
6	6

7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

И, наконец, рассмотрим способы перевода чисел из одной системы счисления в другую.

Выше уже был приведен пример перевода двоичного числа 110_2 в десятичное. Сформулируем общее правило для перевода числа из любой системы счисления в десятичную:

Чтобы перевести число в десятичное представление, надо вычислить сумму цифр, каждая из которых умножается на основу системы в степени (разряд цифры – 1).

Так, если *Basis* — это основа системы счисления, а *abcd* — литеральная запись числа в этой системе, то получаем:

$$abcd_{\text{Basis}} = d \times \text{Basis}^0 + c \times \text{Basis}^1 + b \times \text{Basis}^2 + a \times \text{Basis}^3.$$

Приведем примеры:

$$11011_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 1 + 2 + 0 + 8 + 16 = 27_{10}.$$

$$FF_{16} = 15 \cdot 16^0 + 15 \cdot 16^1 = 255_{10}.$$

Чтобы перевести десятичное число в другую систему счисления, это число надо повторно делить на основу той системы до тех пор, пока в частном делении не получим ноль, запоминать остатки, а затем выписать их в обратном порядке.

Рассмотрим примеры:

$$\begin{array}{r|l}
 137 & 2 \\
 \hline
 136 & 68 \\
 \hline
 1 & 68 \\
 & 0 \\
 & \hline
 & 34 \\
 & \hline
 & 34 \\
 & \hline
 & 17 \\
 & \hline
 & 16 \\
 & \hline
 & 8 \\
 & \hline
 & 8 \\
 & \hline
 & 4 \\
 & \hline
 & 4 \\
 & \hline
 & 2 \\
 & \hline
 & 2 \\
 & \hline
 & 1 \\
 & \hline
 & 0 \\
 & \hline
 & 0 \\
 & \hline
 & 0 \\
 & \hline
 & 1
 \end{array}$$

т.е. $137_{10} = 10010001_2$.

$$\begin{array}{r|l}
 17350 & 16 \\
 \hline
 16 & 1084 \\
 \hline
 1 & 96 \\
 & \hline
 & 124 \\
 & \hline
 & 112 \\
 & \hline
 & 12 \\
 & \hline
 & 67 \\
 & \hline
 & 64 \\
 & \hline
 & 3 \\
 & \hline
 & 4 \\
 & \hline
 & 4 \\
 & \hline
 & 0 \\
 & \hline
 & 0
 \end{array}$$

т. е. $17350_{10} = 43C1_{16}$

$$\begin{array}{r|l}
 43981 & 16 \\
 \hline
 32 & 2748 \\
 \hline
 119 & 16 \\
 \hline
 112 & 114 \\
 \hline
 78 & 112 \\
 \hline
 64 & 28 \\
 \hline
 141 & 16 \\
 \hline
 128 & 2 \\
 \hline
 13 & \\
 \hline
 & \\
 & \hline
 & 171 \\
 & \hline
 & 160 \\
 & \hline
 & 10 \\
 & \hline
 & 0 \\
 & \hline
 & 10 \\
 & \hline
 & 16 \\
 & \hline
 & 0
 \end{array}$$

т.е. $43981_{10} = ABCD_{16}$

Приведем в таблице соответствия числа от 0_{10} до 16_{10} в десятичном, шестнадцатеричном и двоичном представлении. Как видим, для выражения одной десятичной или шестнадцатеричной цифры используется группа из четырех двоичных цифр, которая называется *тетрадой*.

Decimal	Hexadecimal	Тетрада	Значение
0	0	0000	0
1	1	0001	1
2	2	0010	10
3	3	0011	11
4	4	0100	100
5	5	0101	101
6	6	0110	110
7	7	0111	111
8	8	1000	1000
9	9	1001	1001
10	A	1010	1010
11	B	1011	1011
12	C	1100	1100
13	D	1101	1101
14	E	1110	1110
15	F	1111	1111

Теперь рассмотрим перевод двоичных чисел в шестнадцатеричные, и наоборот, минуя десятичное представление.

Для этого двоичное число надо разбить на тетрады справа налево, дополнив спереди последнюю неполную тетраду (если такая имеется) нулями, а затем заменить каждую тетраду соответствующей шестнадцатеричной цифрой.

Например,

$$10110110111110_2 \rightarrow \underbrace{0010}_2 \underbrace{1101}_D \underbrace{1011}_B \underbrace{1110}_E \rightarrow 2DBE_{16}.$$

Обратный перевод числа из шестнадцатеричной системы в двоичную осуществляется путем замены каждой шестнадцатеричной цифры значением соответствующей тетрады:

$$864A_{16} \rightarrow \underbrace{1000}_8 \underbrace{0110}_6 \underbrace{0100}_4 \underbrace{1010}_A$$

Заметим некоторые закономерности.

❖ Основа в любой системе счисления литерально выражается как «10».

❖ Умножение числа на основу литерально выражается в добавлении символа «0» в конце справа, и, наоборот, при делении на основу от записи числа убирается последний правый символ «0» (при условии, что число делится на основу без остатка):

$$25_{10} \cdot 10_{10} = 250_{10}.$$

$$10_2 (=2) \cdot 10_2 (=2) = 100_2 (=4).$$

$$F_{16} (=15) \cdot F_{16} (=15) = F0_{16} (=225).$$

$$250_{10} : 10_{10} = 25_{10}.$$

$$100_2 (=4) : 10_2 (=2) = 10_2 (=2).$$

$$F0_{16} (=225) : F_{16} (=15) = F_{16} (=15).$$

❖ Заметим, что если числа 10, 100, 1000, 10^6 и т.д. в десятичной системе счисления являются «круглыми» числами, то в двоичной системе такие числа, как 2, $2^2 = 4$, $2^4 = 16$, $2^{10} = 1\,048\,576$ и т.д. являются вполне «круглыми». Именно они и используются для выражения единиц измерения информации в байтах:

$$1 \text{ байт} = 2^3 \text{ битов}$$

$$1 \text{ слово (Word)} = 2 \text{ байта}$$

$$1 \text{ КилоБайт (Кб)} = 1024 \text{ байта } (2^{10} \text{ Б})$$

$$1 \text{ МегаБайт (Мб)} = 1\,048\,576 \text{ байтов } (2^{20} \text{ Б})$$

$$1 \text{ ГигаБайт (Гб)} = 1\,073\,741\,824 \text{ байтов } (2^{30} \text{ Б})$$

1.8. Цикл выполнения.

Когда компьютер следует инструкциям компьютерной программы, говорят, что *выполняется программа*. Чтобы компьютер мог запустить программу, она сначала должна стать резидентной в памяти, т.е. должна быть переведена на внутренний машинный язык компьютера и занимать определенную последовательность байтов в оперативной памяти. Каждый ЦП понимает свой собственный машинный язык, а это означает, что перед тем, как запустить программу на другой компьютер, она должна быть переписана на соответствующий машинный язык ЦП.

Тот фактор, что запускаемая программа хранится в памяти, является очень важным моментом. Именно этот принцип делает компьютер решателем задач такого общего назначения – изменив только программу в памяти компьютера, можно выполнить совершенно другую задачу.

Когда программа загружена в память, выполняется следующий цикл:

Присвоить адрес первой инструкции переменной
Адреса Инструкции (АИ)

Пока программа не закончена, повторять:

```
{  
  Вызвать инструкцию с текущего АИ  
  Обновить АИ  
  Выполнить инструкцию  
}
```

Эта последовательность называется циклом подбораки – выполнения (etch-execute cycle).