***Глава 2.***

***1. Опишите существующие единицы информации и их представление.***

***Ответ:*** Элемент памяти, содержащий один двоичный разряд, называется **битом**. Единица информации, состоящая из восьми битов, называется **байтом**. Для большего значения используются следующие единицы информации:

* **слово** — 2 байта или 16 бит, максимальное значение 65 535 (2^16 − 1);
* **двойное слово** — 4 байта или 32 бита, максимальное значение 2^32 − 1;
* **учетверенное слово** — 8 байт или 64 бита, максимальное значение 2^64 − 1;
* **параграф** — шестнадцать байт, максимальное значение 2^128 – 1.
* **двоичные (binary) числа** — каждая цифра означает значение одного бита (0 или 1), старший бит всегда пишется слева, после числа ставится буква «b». Для удобства восприятия тетрады могут быть разделены пробелами. Например, 1010 0101b;
* **шестнадцатеричные (hexadecimal) числа** — каждая тетрада представляется одним символом 0, . . ., 9, А, В, . . ., F. Обозначаться такое представление может по-разному. В данном пособии используется символ «h» после последней шестнадцатеричной цифры (например, A5h). В текстах программ это же число может обозначаться и как 0хА5, и как 0A5h, в зависимости от синтаксиса языка программирования. Незначащий ноль (0) добавляется слева от старшей шестнадцатеричной цифры, изображаемой буквой, чтобы различать числа и символические имена;
* **десятичные (decimal) числа** — каждый байт (слово, двойное слово) представляется обычным числом, а признак десятичного представления (букву «d») обычно опускают. Байт из предыдущих примеров имеет десятичное значение 165. В отличие от двоичной и шестнадцатеричной форм записи по десятичной трудно в уме определить значение каждого бита, что иногда приходится делать;
* **восьмеричные (octal) числа** — каждая тройка бит (разделение начинается с младшего) записывается в виде цифры 0–7, в конце ставится признак «о». То же самое число будет записано как 245о. Восьмеричная система неудобна тем, что байт нельзя разделить поровну, но зато все цифры — привычные.

В двухбайтном слове принят LH-порядок следования байт: адрес слова указывает на младший байт L (Low), а старший байт Н (High) размещается по адресу, на единицу большему. В двойном слове порядок будет аналогичным — адрес укажет на самый младший байт, после которого будут размещены следующие по старшинству. Этот порядок, естественный для процессоров Intel, применяется не во всех микропроцессорных семействах. Байт (8 бит) делится на пару тетрад (nible): старшую тетраду — биты (7:4) и младшую тетраду — биты (3:0).

***2. Дайте определения видам памяти и их месту в иерархии.***

***Ответ:* Память компьютера** предназначена для кратковременного и долговременного хранения информации — кодов команд и данных. В памяти информация хранится в массиве ячеек. Минимальной адресуемой единицей является байт — каждый байт памяти имеет свой уникальный адрес. Память можно рассматривать как иерархическую систему. Входящие в ее состав запоминающие устройства различаются емкостью, дбыстродействием и скоростью. **Самыми быстродействующими являются регистры центрального процессора** (ЦП), и они же имеют наименьшую емкость. Обращение к ним происходит при выполнении почти каждой машинной команды. **Кэш-память** процессора в целом отличается меньшим объемом, но более высоким быстродействием по сравнению с основной памятью. Она функционирует как буфер между процессором и основной памятью и содержит копии областей памяти, которые использовались последними, — их адреса и расположенные по этим адресам данные. Каждый раз, когда процессору требуется обратиться к памяти по заданному адресу, он сначала обращается к кэшу. Если происходит промах кэша, то есть этот адрес в кэше отсутствует, производится обращение к основной памяти. Данные считываются из памяти в указанный в команде регистр процессора, а их копия записывается в кэш. Для большей части программного обеспечения характерна временная локализация доступа, то есть значительная вероятность повторного обращения по одним тем же адресам в течение короткого промежутка времени. Если повторное обращение произойдет достаточно скоро, нужные данные все еще будут в кэше и их не придется считывать из основной памяти. Такая ситуация называется «попаданием в кэш». **Передача информации** между основной памятью и кэш-памятью осуществляется по шине памяти, которая состоит из шины данных и шины адреса. Со времени появления больших по размерам компьютеров сложилось деление памяти на внутреннюю и внешнюю. Под внутренней подразумевалась память, расположенная внутри процессорного «шкафа» (или плотно к нему примыкающая). Внешняя память представляла собой отдельные устройства с подвижными носителями — накопителями на магнитных дисках (а ранее, барабанах) и ленте. Со временем все устройства компьютера удалось поместить в один небольшой корпус, и прежнюю классификацию памяти применительно к персональным компьютерам можно переформулировать следующим образом. **Внутренняя память** — электронная (полупроводниковая) память, устанавливаемая на системной плате или на платах расширения. **Внешняя память** — память, реализованная в виде устройств с различными принципами хранения информации, чаще всего с подвижными носителями. В настоящее время сюда входят устройства магнитной (дисковой и ленточной) памяти, оптической и магнитооптической памяти; устройства внешней памяти могут размещаться как в системном блоке компьютера, так и в отдельных корпусах, достигающих иногда размеров небольшого шкафа.

**Информация из внешней памяти во внутреннюю память попадает по шине ввода-вывода.** Для процессора непосредственно доступной является внутренняя память, доступ к которой осуществляется по адресу, заданному программой. Для внутренней памяти характерен одномерный (линейный) адрес, который представляет собой одно двоичное число определенной разрядности. Внутренняя память подразделяется на оперативную (ОЗУ — оперативно запоминающее устройство), информация в которой может изменяться процессором в любой момент времени, и постоянную (ПЗУ — постоянно запоминающее устройство), информацию, которую процессор может только считывать. Обращение к ячейкам оперативной памяти может происходить в любом порядке как по чтению, так и по записи, поэтому оперативную память называют памятью с произвольным доступом RAM (Random Access Memory) — в отличие от постоянной памяти ROM (Read Only Memory).

***3. Что собой представляет адресация и распределение памяти в реальном режиме работы микропроцессора Intel x86?***

***Ответ:*** **Адресация ячеек памяти в реальном режиме**. В реальном режиме адрес имеет размер 20 бит. Максимальный объем адресуемой памяти составляет 1 Мбайт (2^20 = 1 Мбайт). Для формирования 20-битового адреса в памяти используются два 16-битовых регистра: сегментный регистр и регистр смещения. Сегментный регистр определяет базовый адрес сегмента. В нем содержатся старшие 16 бит базового адреса сегмента, младшие четыре бита предполагаются равными 0. В регистре смещения находится смещение внутри сегмента (относительно базового адреса сегмента). Смещение может также задаваться непосредственной константой. Смещение добавляется к базовому адресу сегмента, и, таким образом, получается 20-битовый адрес в памяти. Более просто можно описать вычисление следующим образом — *16-битовое значение сегментного регистра сдвигается на 4 бита влево и к нему добавляется значение смещения*, в результате получается 20-битовое значение. Или это же действие можно представить математической формулой: **Физический адрес = Сегментный адрес ⋅ 10h + Смещение**. Такая схема применялась в 8086 и 8088 микропроцессорах, и такое же представление 20-битного адреса двумя 16-битными числами поддерживается в реальном режиме всех последующих микропроцессоров х86. С использованием реального режима функционировала OC MS DOS (Microsoft Disk Operating System) — дисковая операционная система от фирмы Microsoft.

***4. Что собой представляет адресация и распределение памяти в защищенном режиме работы микропроцессора Intel x86?***

***Ответ:*** Управление памятью в архитектуре Intel делится на две части: сегментацию и трансляцию страниц. Сегментация предоставляет механизм для изолирования индивидуального кода, данных и стека (специальная область памяти, предназначенная для временного хранения данных и организованная по принципу «первым вошел, последним вышел»). Трансляция страниц предоставляет механизм для реализации виртуальной памяти с подкачкой страниц по запросу, где части программы отображаются, как необходимо, на физическую память. Трансляция страниц также может использоваться для изоляции между несколькими задачами. Все сегменты содержатся внутри линейного адресного пространства процессора. Для доступа к любому байту информации в каком-либо сегменте необходим логический адрес (иногда его называют дальним указателем). **Логический адрес состоит из селектора сегмента и смещения.** Селектор сегмента — это уникальный идентификатор сегмента. Селектор сегмента также является смещением в таблице дескрипторов. Каждый сегмент имеет дескриптор сегмента, который определяет размер, права доступа, уровень привилегий, тип сегмента и расположение его первого байта внутри линейного пространства. Смещение добавляется к базовому адресу сегмента для доступа к конкретному байту внутри сегмента. Базовый адрес плюс смещение формируют линейный адрес внутри линейного адресного пространства процессора. Если трансляция страниц не используется, то линейное адресное пространство непосредственно отображается на физическое адресное пространство процессора. Физическое адресное пространство определяется как диапазон адресов, которые процессор может генерировать на шине адреса. В многозадачных операционных системах линейное пространство гораздо больше, чем пространство физической памяти, поэтому необходим метод виртуализации линейного адресного пространства. Эта виртуализация линейного адресного пространства производится через механизм трансляции страниц. Трансляция страниц поддерживает среду с виртуальной памятью, где большое линейное пространство эмулируется за счет маленького количества физической памяти и некоторого количества дисковой памяти. При этом все сегменты делятся на 4-килобайтовые страницы, которые могут находиться либо в памяти, либо на диске. Когда задача пытается обратиться к странице, которой нет в памяти, происходит исключение и операционная система загружает недостающую страницу в память, после чего выполнение задачи возобновляется. Для более экономного использования памяти используется двухуровневая таблица схемы трансляции страниц. На верхнем уровне находится каталог страниц, который содержит ссылки на таблицы страниц. Таблицы страниц содержат ссылки на сами страницы. Трансляция страниц — это единственный способ, при котором возможно выполнение нескольких задач реального режима. В сегментном регистре в старших 14 битах находится селектор сегмента. Младший бит селектора (2 бит сегментного регистра при нумерации от нуля) указывает, какая таблица дескрипторов должна использоваться. Нуль соответствует глобальной таблице дескрипторов, единица — локальной таблице дескрипторов. Поле RPL (0 и 1 бит сегментного регистра) используется для контроля прав доступа программы к сегменту. Эффективный адрес формируется суммированием компонентов base (базы), index (индекса), displacement (смещения) с учетом scale (масштаба). Поскольку каждая задача может иметь до 16 Кбайт селекторов (2 14), а смещение, ограниченное размером сегмента, может достигать 4 Гбайт, логическое адресное пространство для каждой задачи может достигать 64 Тбайт. Все это пространство виртуальной памяти в принципе доступно программисту при условии поддержки со стороны операционной системы. Блок сегментации транслирует логическое адресное пространство в 32-битное пространство линейных адресов. 32-битный физический адрес памяти образуется после преобразования линейного адреса блоком страничной переадресации. Он выводится на внешнюю шину адреса процессора. В простейшем случае при отключенном блоке страничной переадресации физический адрес совпадает с линейным. Включенный блок страничной переадресации осуществляет трансляцию линейного адреса в физический страницами размером 4 Кбайт (для последних поколений процессоров возможны страницы размером 2–4 Мбайт). Блок обеспечивает расширение разрядности физического адреса процессоров шестого поколения до 36 бит. Блок переадресации может включаться только в защищенном режиме. Для обращения к памяти процессор совместно с внешними схемами формирует шинные сигналы для операций записи и чтения. Шина адреса разрядностью 32/36 бит позволяет адресовать 4/64 Гбайт физической памяти, но в реальном режиме доступен только 1 Мбайт, начинающийся с младших адресов.

***5. Что собой представляет адресация и распределение памяти в архитектуре AMD64?***

***Ответ:*** Переход к 64-битной архитектуре был обусловлен необходимостью выделения программе больше 2 Гбайт памяти, то есть преодолеть ограничение 32-битных систем. В 32-битных системах на платформе Windows шина адреса позволяет адресовать 4 Гбайта памяти (2^32). Верхние 2 Гбайта адресов зарезервированы для ядра и API-функций. Реально Windows-приложениям доступно всего 2 Гбайта адресов (3 Гбайта в некоторых версиях Windows Advanced Server при указании специальной опции). Потребности же многих приложений (баз данных, программ обработки растровой графики и видео, систем автоматизированного проектирования), могут превышать этот лимит, так как оперируют большими наборами данных. Процессоры Athlon 64 и Opteron не полностью 64-битные. Для указателей в программах используются новые 64-битные регистры, но объем адресуемой памяти в 16 эксабайт для 64-битной адресации в AMD посчитали излишним. Поэтому процессоры с архитектурой AMD64 используют 48-битную адресацию, и позволяют адресовать до 256 Тбайт виртуальной памяти. Старшие 16 бит 64-битного адреса не используются. Для задания физического адреса в адресной шине отведено 40 бит, что позволяет иметь до 1 Тбайт физической оперативной памяти. Скорость компьютера сильно зависит от нагрузки, приходящейся на каналы памяти. В типичном коде около половины объема приходится на адреса операндов и связанные с ними константы, следовательно, двукратное увеличение разрядности адреса должно привести к увеличению объема кода примерно в полтора раза. Что, в свою очередь, означало бы пропорциональное увеличение нагрузки на шину данных и снижение эффективности кэша. Иначе говоря, переход Athlon’а в 64-битный режим мог бы сопровождаться заметным падением производительности. Для предотвращения данной проблемы инженеры AMD добавили новый режим адресации относительно 64-битного указателя инструкций — RIP-relative addressing. Эффективный адрес в этом случае получается суммированием 32-битного адреса операнда, который играет роль смещения, и регистра RIP2. Адреса, используемые в командах перехода, имеют закон распределения, близкий к нормальному, с вершиной в команде перехода. То есть чем дальше адрес отстоит от текущего значения указателя инструкций, тем меньше вероятность его применения. Это позволяет снизить разрядность адресов в командах. Когда 32-х бит все же недостаточно, компилятор формирует полный 64-битный адрес, после чего, по возможности, снова продолжает генерировать 32-битные адреса. Такой подход практически ликвидировал главный недостаток 64-битного кода — большой размер.Большинство 64-битных операционных систем поддерживают режим совместимости Compatibility Mode. Но вызов системной функции, сделанный в 64-битной среде 32-битным приложением, нуждается в преобразовании некоторых аргументов, в основном, указателей адреса. ОС Windows реализует эти преобразования через подсистему «Windows on Windows». Специальная библиотека, wow.dll, динамически подключается ко всем 32-битным приложениям и выполняет следующие функции:

* приводит аргументы к 64-битному виду;
* передает управление (вызов) 64-битному ядру;
* приводит результаты вызова обратно к 32-битному виду;
* возвращает их приложению.

***Виртуальная память***. Технология расширения доступной физической памяти компьютера. В системе виртуальной памяти операционная система создает файл подкачки и делит память на единицы, называемые страницами. Страницы, к которым недавно обращались, хранятся в физической памяти (ОЗУ). Если какое-то время к странице памяти не обращаются, она записывается в файл подкачки (процесс, называемый «подкачкой памяти» или «откачкой страницы»). Если позже к этому участку памяти обращается программа, операционная система считывает страницу памяти из файла подкачки и помещает ее в физическую память (процесс, называемый «подкачкой памяти» или «подкачкой страницы»). Общий объем памяти, доступной программам, равен сумме физической памяти компьютера и размера файла подкачки. ***Файл подкачки*** — файл, хранящийся на диске, который используется компьютером для увеличения объема физического хранилища виртуальной памяти. **Гиперпространство** — специальная область, используемая для управления страницами процессов с применением виртуальной памяти. **Выгружаемый пул** — область виртуальной памяти в системном пространстве, которая может выгружаться из рабочего множества системных процессов и загружаться в него. **Невыгружаемый пул** — область памяти, состоящая из ячеек виртуальных системных адресов, которые всегда находятся в физической памяти, и поэтому доступ к ним возможен из любого адресного пространства без загрузки/выгрузки при подкачке страниц. **Системный кэш** — страницы памяти, используемые для распределения открытых файлов в системном кэше. **PTE (Page Table Entry) системы** — пул системных элементов таблицы страниц, используемый для сопоставления системных страниц, таких как загружаемое/выгружаемое пространство, стеки ядра и списки дескрипторов памяти.

***6. Как в ОС Windows на платформе NT можно узнать такие параметры, как общий размер физической памяти и размер памяти ядра ОС?***

***Ответ:*** В операционной системе Windows через меню Пуск/Все программы/Стандартные/Служебные можно вызвать приложение «Сведения о системе», в котором можно получить сведения об основных характеристиках организации памяти в компьютере (рис. 2.10). В этом приложении можно узнать полный объем физической, виртуальной и свободной в данный момент памяти, размещение и объем файла подкачки и т. п.

***7. Что такое файл подкачки и как им можно управлять в ОС Windows на платформе NT?***

***Ответ:* Файл подкачки** — это область жесткого диска, используемая Windows для хранения данных оперативной памяти. Он создает иллюзию, что система располагает большим объемом оперативной памяти, чем это есть на самом деле. Единой стратегии работы с файлом подкачки не существует. Многое определяется назначением и настройкой компьютера. По умолчанию Windows удаляет файл подкачки после каждого сеанса работы и создает его в процессе загрузки операционной системы. Размер файла постоянно меняется по мере выполнения приложений и контролируется операционной системой. Обычно используется единственный файл подкачки, расположенный на том же диске, что и операционная система. Наличие двух жестких дисков может дать значительное преимущество при настройке файла подкачки. Для максимально эффективного использования файла подкачки нужно так его настроить, чтобы он располагался на жестком диске в виде достаточно протяженных фрагментов (это уменьшает количество перемещений считывающей головки, радикально влияющих на производительность). Кроме того, файл подкачки необходимо периодически удалять, чтобы избежать его фрагментации. Определение размера файла подкачки до сих пор вызывает многочисленные дискуссии. Основное правило заключается в том, что при небольшом объеме оперативной памяти файл подкачки должен быть достаточно большим. При большом объеме оперативной памяти файл подкачки можно уменьшить. Существует возможность вообще ликвидировать файл подкачки, выполнив определенную настройку реестра. Существует несколько приемов, позволяющих оптимизировать использование файла подкачки для повышения производительности. Файл подкачки следует по возможности размещать на отдельном жестком диске. При наличии нескольких жестких дисков файл подкачки следует разделить, т. к. это повышает скорость работы с ним. Если при наличии двух жестких дисков разделить файл подкачки, то поскольку доступ к данным на обоих жестких дисках осуществляется одновременно, это значительно повысит производительность. Однако если имеются два жестких диска, из которых один быстрее другого, возможно, более эффективным решением будет размещение файла подкачки только на более быстром жестком диске. Определить наиболее производительную конфигурацию для данной системы можно экспериментальным путем.

Для установки размера файла подкачки нужно выполнить следующую последовательность действий. Щелкнуть правой клавишей мыши по значку «Мой компьютер» и выбрать в контекстном меню строку «Свойства». На экране появится

окно «Свойства системы». Перейти на вкладку «Дополнительно» и нажать кнопку «Параметры» в рамке «Быстродействие». В появившемся окне «Параметры быстродействия» на вкладке «Дополнительно» нажать кнопку «Изменить».