ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»

Методические рекомендации для самостоятельной работы обучающихся по дисциплине

Б3.Б.2 Аппаратные средства вычислительной техники
(*код и наименование дисциплины в соответствии с РУП*)

Направление подготовки (специальность) 10.03.01 «Информационная безопасность»

Профиль образовательной программы Безопасность автоматизированных систем

Форма обучения *очная*

Содержание

1. ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1.1 Организационно-методические данные дисциплины

№ п. п.	Наименование темы	Общий объем часов по видам самостоятельной работы (*из табл. 5.1 РПД*)				
		подготовка курсового проекта (работы)	подготовка реферата/эссе	индивидуальные домашние задания (ИДЗ)	самостоятельное изучение вопросов (СИВ)	подготовка к занятиям (ПкЗ)
1	2	3	4	5	6	7
1	ифметические основы построения и логические основы построения ЭВМ.				2	
2	Минимизация логических функций. Выполнение операций в двоичном коде				2	2
3	строение логических схем. Комбинированные узлы. Узлы с памятью.					
4	Структуры запоминающих устройств ЭВМ. Структура ОЗУ.					
5	Устройства хранения данных. Структура основной памяти.				2	2
6	Устройства хранения данных.					
7	Аудиосистема ПК. Коммуникационные устройства.					
8	ринципы построения процессора. Структура машинных команд и способы адресации.				2	2
9	Современные микропроцессоры. Порядок выполнения машинных				2	

	команд.					
10	рганизация системы прерываний. Организация перехода к прерывающей программе. Принципы организации ввода-вывода.					
11	хитектура системной платы. Установка и конфигурирование компонентов.		2		2	2
12	Шины расширения. Шина USB.					2
13	Параллельный интерфейс. Последовательный интерфейс.					

2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО

САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ ВОПРОСОВ

2.1 Ячейки памяти, порты и регистры

Поясним разницу между ячейками памяти, портами и регистрами. Ячейки памяти служат лишь для хранения информации — сначала ее записывают в ячейку, а потом могут прочитать, а также записать иную информацию. Порты ввода-вывода, как правило, служат для преобразования двоичной информации в какие-либо физические сигналы и обратно. Например, порт данных параллельного интерфейса формирует электрические сигналы на разъеме, к которому обычно подключают принтер. Электрические сигналы, поступающие от принтера, порт состояния того же интерфейса отображает в виде набора битов, который может быть считан процессором. Регистр — довольно широкое понятие, которое зачастую используется как синоним порта. Регистры могут служить для управления устройствами (и их контроллерами) и для чтения их состояния. Регистры (как и порты) могут образовывать каналы:

Каналы ввода-вывода данных. Пример — регистр данных COM-порта: байты, записываемые друг за другом в этот регистр, в том же порядке будут передаваться по последовательному интерфейсу, то есть поступать в канал вывода.

Если этот интерфейс подключить к COM-порту другого компьютера и выполнять программные чтения его регистра данных, мы получим байт за байтом переданные данные. Таким образом, здесь регистр играет роль канала ввода.

Каналы управления. Если запись в регистр определенных данных (битовых комбинаций) изменяет состояние некоего устройства (сигнал светофора, положение какого-то механизма...), то регистр образует канал управления.

Каналы состояния. Пример — регистр игрового порта (game-порт), к которому подключен джойстик. Чтение регистра дает информацию о состоянии кнопок джойстика (нажаты или нет).

Канал отличается от ячейки памяти рядом свойств. Если в ячейку памяти записывать раз за разом информацию, то последующее считывание возвращает результат последней записи, а все предшествующие записи оказываются бесполезными. Если ячейку памяти считывать раз за разом, не выполняя запись в нее, то результат считывания каждый раз будет одним и тем же (при исправной памяти). «Лишнее» чтение ячейки памяти не приведет ни к каким побочным эффектам. На этих свойствах «настоящей» памяти основаны методы ускорения работы с ней: кэширование и спекулятивное чтение.

С регистрами, образующими каналы, такие вольности недопустимы. Здесь все обращения приводят к каким-либо изменениям. Кэширование и спекулятивное чтение недопустимы. Например, лишнее (спекулятивное) чтение регистра данных COM-порта «выдернет» байт из принимаемого потока. Операция чтения регистра состояния может быть неявным подтверждением сброса какого-либо признака (например, запроса прерывания), и она изменяет состояние устройства. Записи в канал данных (и управления) также нельзя опускать (для «ускорения»). Каждый байт (ячейка памяти, порт, регистр) имеет собственный уникальный физический адрес. Этот адрес устанавливается на системной шине процессором, когда он инициирует обращение к данным ячейке или порту. По этому же адресу к этой ячейке (порту, регистру) могут обращаться и другие активные компоненты системы — так называемые мастера шины.

В семействе x86 и PC-совместимых компьютерах пространства адресов ячеек памяти и портов ввода-вывода разделены. Это предусмотрено с обеих сторон: процессоры позволяют, а компьютеры используют данное разделение. Нынешние 32-битные процессоры имеют разрядность физического адреса памяти 32 и даже 36 бит, что

позволяет адресовать до 4 и 64 Гбайт соответственно. Пространство ввода-вывода использует только младшие 16 бит адреса, что позволяет адресовать до 65 384 однобайтных регистров. Адреса «исторических» системных устройств PC не изменились с самого рождения — это дань совместимости, которая без разделения пространств вряд ли бы обеспечивалась столько лет. Пространства памяти и портов ввода-вывода неравнозначны не только по объему, но и по способам обращения. Способов адресации к ячейке памяти в x86 великое множество, в то время как для адресации ввода-вывода их существует только два. К памяти возможна (и широко используется) виртуальная адресация (см. 7.3), при которой для программиста, программы и даже пользователя создается иллюзия оперативной памяти гигантского размера. К портам ввода-вывода обращаются только по реальным адресам; правда, и здесь возможна виртуализация, но уже чисто программными средствами операционной системы. И, наконец, самое существенное различие пространств памяти и портов ввода-вывода: процессор может считывать инструкции для исполнения только из пространства памяти. Конечно, через порт ввода можно считать фрагмент программного кода (что и происходит, например, при считывании данных с диска), но для того чтобы этот код исполнить, его необходимо записать в память.
Регистры различных устройств могут быть приписаны как к пространству портов ввода-вывода, так и к пространству памяти. Под портом устройства, как правило, подразумевают регистр, связанный с этим устройством и приписанный к пространству портов ввода-вывода. Точность приведенной терминологии, конечно же, относительна. Так, к примеру, ячейки видеопамяти (тоже память!) служат в основном не для хранения информации, а для управления свечением элементов экрана. Понятие Memory Mapped I/O означает регистры периферийных устройств, отображенные на пространство памяти (то есть занимающие адреса именно в этом пространстве, а не в пространстве ввода-вывода).
Разделение пространств памяти и ввода-вывода было вынужденной мерой в условиях дефицита адресуемого пространства 16-битных процессоров и сохранилось во всех процессорах x86. В процессорах ряда других семейств такого разделения нет, и для нужд ввода-вывода используется выделенная область единого адресного пространства. Тенденция изживания пространства ввода-вывода наблюдается в современных спецификациях устройств и интерфейсов для PC.

2.2 Подсистема памяти и хранения данных

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, простирающуюся от кэш-памяти процессора до ленточных архивов.
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). Сюда входили и электронная и магнитная память (на магнитных сердечниках). Внешняя память представляла собой отдельные устройства с подвижными носителями — накопители на магнитных дисках (а сначала — на барабанах) и ленте. Со временем все устройства компьютера удалось «поселить» в один небольшой корпус, и прежнюю классификацию памяти применительно к PC можно переформулировать так:
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.
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, информация в которой может изменяться процессором в любой момент времени, и постоянную,информацию в которой процессор может только считывать. Обращение к ячейкам оперативной памяти может происходить в любом порядке, причем как по чтению, так и по записи, поэтому оперативную память называют памятью с произвольным доступом (Random Access Memory, RAM) — в отличие от постоянной памяти (Read Only Memory, ROM).
Внешняя память адресуется более сложным образом — каждая ее ячейка имеет свой адрес внутри некоторогоблока, который, в свою очередь, имеет многомерный адрес. В ходе физических операций обмена данными блок может быть считан или записан только целиком. В случае одиночного дискового накопителя физический адрес блока является трехмерным — он состоит из номера поверхности (головки), номера цилиндра и номера сектора. В современных накопителях этот трехмерный адрес часто заменяют линейным номером — логическим адресом блока, а его преобразованием в физический адрес занимается внутренний контроллер накопителя. Поскольку дисковых накопителей в компьютере может быть множество, в адресации дисковой памяти участвуют также номер накопителя и номер канала интерфейса. С такой сложной системой адресации процессор справляется только с помощью программного драйвера, в задачу которого в общем случае входит копирование некоторого блока данных из оперативной памяти в дисковую и обратно. Название «дисковая память» широко применяется для внешней памяти с прямым доступом; словосочетание «прямой доступ» подразумевает возможность обращения к блокам (но не к его ячейкам!) с чередованием операций чтения и записи в произвольном порядке. Память с последовательным доступом накладывает ограничения на свободу: в ней невозможны произвольное чередование операций чтения/записи и произвольность адресов. Ряд устройств запись вообще не выполняют (например, CD-ROM).
Последовательный метод доступа используется в ленточных устройствах, а также в большинстве оптических дисков (CD, DVD). С такими неудобствами обращения мирятся только из-за того, что устройства последовательного доступа обеспечивают самое дешевое хранение больших объемов информации, к которой не требуется оперативного доступа:
Ниже перечислены наиболее важные параметры подсистемы памяти.
Объем хранимой информации. Нет необходимости объяснять, что чем он больше, тем лучше. Максимальный (в принципе — неограниченный) объем информации хранят ленточные и дисковые устройства со сменными носителями, за ними идут дисковые накопители, и завершает этот ряд оперативная память.
Время доступа — усредненная задержка начала обмена полезной информацией относительно появления запроса на данные. Минимальное время доступа имеет оперативная память, за ней идет дисковая, после нее — ленточная.
Скорость обмена при передаче потока данных (после задержки на время доступа). Максимальную скорость обмена имеет оперативная память, за ней идет дисковая, после нее — ленточная.
Удельная стоимость хранения единицы данных — цена накопителя (с носителями), отнесенная к единице хранения (байту или мегабайту). Минимальную стоимость хранения имеют ленточные устройства со сменными носителями, их догоняют дисковые накопители, а самая дорогая — оперативная память.

Помимо этих имеется и ряд других характеристик — энергонезависимость (способность сохранения информации при отключении внешнего питания), устойчивость к внешним воздействиям, время хранения, конструктивные особенности (размер, вес) и т. п. У каждого типа памяти есть различные реализации со своими достоинствами и недостатками.

2.3 Установка и обслуживание устройств

1. Техническое обслуживание, ремонт и профилактика ВТ и ПО

Работы по ремонту вычислительной техники и восстановлению разрушенного программного обеспечения (операционная система Windows, Microsoft Office, архиваторы, антивирусные программы, драйверы периферийных устройств), выполняются на основании заявок, подаваемых должностными лицами подразделений академии по телефону, электронной почте (тел. 3-41, к.312), либо по письменной заявке установленного образца, зарегистрированной в отделе ВиОТ (тел. 3-41, к.312).

Срок исполнения заявок:

а) Ремонт без замены комплектующих – в течение рабочего дня, следующего за днем подачи заявки, в порядке очередности.

б) Ремонт с заменой комплектующих – в течение двух рабочих дней, следующих за днем подачи заявки при условии наличия комплектующих, в порядке очередности.

в) В случае отсутствии комплектующих - по мере их приобретения в срок до 30 дней после подачи заявки.

Примечание 1: При выходе из строя печатающего устройства, время выполнения ремонта, в зависимости от сложности, может быть дополнительно увеличено на 1-5 рабочих дней.

Примечание 2: Работы по ремонту вычислительной и оргтехники могут выполняться специалистами сторонних организаций по договору, в этом случае сотрудниками отдела ВиОТ осуществляется контроль исполнения работ.

2. Сопровождение программного обеспечения

Заявки на консультирование по установленному на вычислительной технике стандартному программному обеспечению (операционная система Windows, Microsoft Office, архиваторы, антивирусные программы, драйверы периферийных устройств) принимаются в отделе ВиОТ по телефону 2-517-341 с 9-00 до 17-00 час и исполняются сотрудниками отдела на рабочем месте заявителя или, при необходимости, в отделе (к.312) с 13-30 до 17-00 час.

Работы по восстановлению программного обеспечения (операционная система Windows, Microsoft Office, архиваторы, антивирусные программы, драйверы периферийных устройств) выполняются в течение 1-2-х рабочих дней, в зависимости от сложности, в порядке очередности.

По согласованию с начальником отдела ВиОТ в перечень сопровождаемого программного обеспечения могут быть включены дополнительные программы, необходимые для использования установленного периферийного оборудования.

3. Установка нового программного обеспечения

Замена или модернизация программных продуктов (операционная система Windows, Microsoft Office, архиваторы, антивирусные программы, драйверы периферийных устройств) производится по письменному заявлению руководителя соответствующего

подразделения на имя начальника отдела ВиОТ при наличии в отделе соответствующей документации и дистрибутивов.

Работы по установке нового программного обеспечения выполняются в течение 1-2-х рабочих дней с момента подачи заявки.

4. Модернизация вычислительной техники

Модернизация вычислительной техники в подразделениях института производится по годовому плану, утверждаемому директором института.
В исключительных случаях модернизация может быть произведена после предварительной консультации с сотрудниками отдела ВиОТ на основании служебной записки руководителя соответствующего подразделения на имя директора, а также при наличии в отделе требующихся для этого комплектующих. Работы по модернизации, в зависимости от их сложности, выполняются в срок от 2 до 5 рабочих дней с момента регистрации заявки.
В случае отсутствия комплектующих, отделом ВиОТ осуществляется их приобретение и учет, после чего осуществляется модернизация вычислительной техники.

5. Ввод в эксплуатацию новой вычислительной техники

Ввод в эксплуатацию новой ВиОТ выполняется в течение 3-х рабочих дней со дня ее поступления в отдел, но не более 3-х компьютеров в неделю.

Новая техника устанавливается в подразделения института на основании служебной записки соответствующего руководителя, составленной на имя директора в соответствии с вынесенной резолюцией.

Новая вычислительная техника выдается материально ответственным лицом руководителю структурного подразделения под расписку. На руководителя структурного подразделения налагается ответственность за сохранность и правильную техническую эксплуатацию техники в подразделении.

6. Замена и получение расходуемых материалов

Замена картриджей выполняется сотрудниками отдела ВиОТ на основании письменной заявки руководителя соответствующего подразделения в соответствии с нормативами, утвержденными директором института.

Новые носители информации получают должностные лица, ответственные за эксплуатацию ВиОТ в подразделении. Носители информации выдаются материально-ответственным лицом отдела ВиОТ на основании письменной заявки руководителя подразделения.

После выполнения любого вида ремонтных работ в обязанности сотрудника отдела ВиОТ входит проверка на рабочем месте пользователя работоспособности компьютера и периферийного оборудования, а также работоспособности основных программ.

2.4 Исполнение программного кода

Задача центрального процессора — выполнять программы (программный код), находящиеся в основной памяти (кэш-памяти). Он вызывает команды из памяти (кэш-памяти), определяет их тип, а затем выполняет их одну за другой. Компоненты соединены шиной, представляющей собой набор параллельно связанных проводов, по которым передаются адреса, данные и сигналы управления.

Когда команды извлекаются из кэша (или основной памяти), их необходимо декодировать и отправить на исполнение.

Как видно, весь процесс обработки команды состоит из четырех шагов, что и определяет так называемый 4 – ступенчатый процесс (конвейер).

1. Извлечение из кэша (оперативной памяти).
2. Декодирование (разборка команды).
3. Исполнение команды (применение действий).
4. Запись в кэш (оперативную память).

Каждую из этих ступеней команда должна проходить ровно за один такт. Поэтому чем быстрее каждая из ступеней выполняет свои функции, тем быстрее работает весь процессор и тем выше его тактовая частота. Выполнение всех этих четырех команд определяет цикл. Большинство процессоров действительно исполняют команды за один цикл, но существуют сложные команды, для которых требуется несколько циклов. При исполнении сложных команд различные устройства задействуют собственные исполнительные конвейеры, тем самым, добавляя еще несколько ступеней к основному конвейеру процессора. Количество ступеней определяет глубину конвейера.

Программный код — это последовательность команд, или инструкций, каждая из которых определенным образом закодирована и расположена в целом числе смежных байтов памяти. Каждая инструкция обязательно имеет операционную часть, несущую процессору информацию о требуемых действиях. Операндная часть, указывающая процессору, где находится его «предмет труда» — операнды, — может присутствовать в явном или неявном виде и даже отсутствовать. Операндная часть может описывать от нуля до двух операндов, участвующих в выполнении данной инструкции (есть инструкции, в которых, помимо двух операндов, задается еще и параметр инструкции). Здесь могут быть сами значения операндов (непосредственные операнды); явные или неявные указания на регистры процессора, в которых находятся операнды; адрес (или его составная часть) ячейки памяти или порта ввода-вывода; регистры процессора, участвующие в формировании адреса, и разные комбинации этих компонентов. Длина инструкции 32-битного процессора семейства x86 может быть от 1 до 12 байтов (а с префиксами — и до 17 байтов) и определяется типом инструкции. Исторически сложившийся формат инструкций x86 довольно сложен, и «понять», сколько байтов занимает конкретная инструкция, процессор может, лишь декодировав ее первые 1-3 байта. Инструкции могут предшествовать префиксы (к счастью, всегда однобайтные, но их может быть несколько), указывающие на изменение способа адресации, размера операнда или/и необходимость многократного (по счетчику и условию) повторения для данной инструкции. Адрес (логический) текущей исполняемой инструкции хранится в специальном регистре — указателе инструкций (InstructionPointer, IP), который соответствует счетчику команд фон-неймановской машины. После исполнения так называемой линейной инструкции этот указатель увеличивает свое значение на ее длину, то есть указывает на начало следующей инструкции. Линейная инструкция не нарушает порядок выполнения инструкций, определяемый последовательностью их расположения в памяти (по нарастанию адреса). Помимо линейных инструкций существуют инструкции передачи управления, среди которых различают инструкции переходов и вызовов процедур. Эти инструкции в явном или неявном виде содержат информацию об адресе следующей выполняемой инструкции, который может указывать на относительно произвольную ячейку памяти.

2.5 Программная модель современных процессоров

Центральный процессор (ЦП) - устройство, непосредственно предназначенное для выполнения вычислительных операций. Процессор работает под управлением программы, выполняя вычисления или принимая логические решения, необходимые для обработки информации.

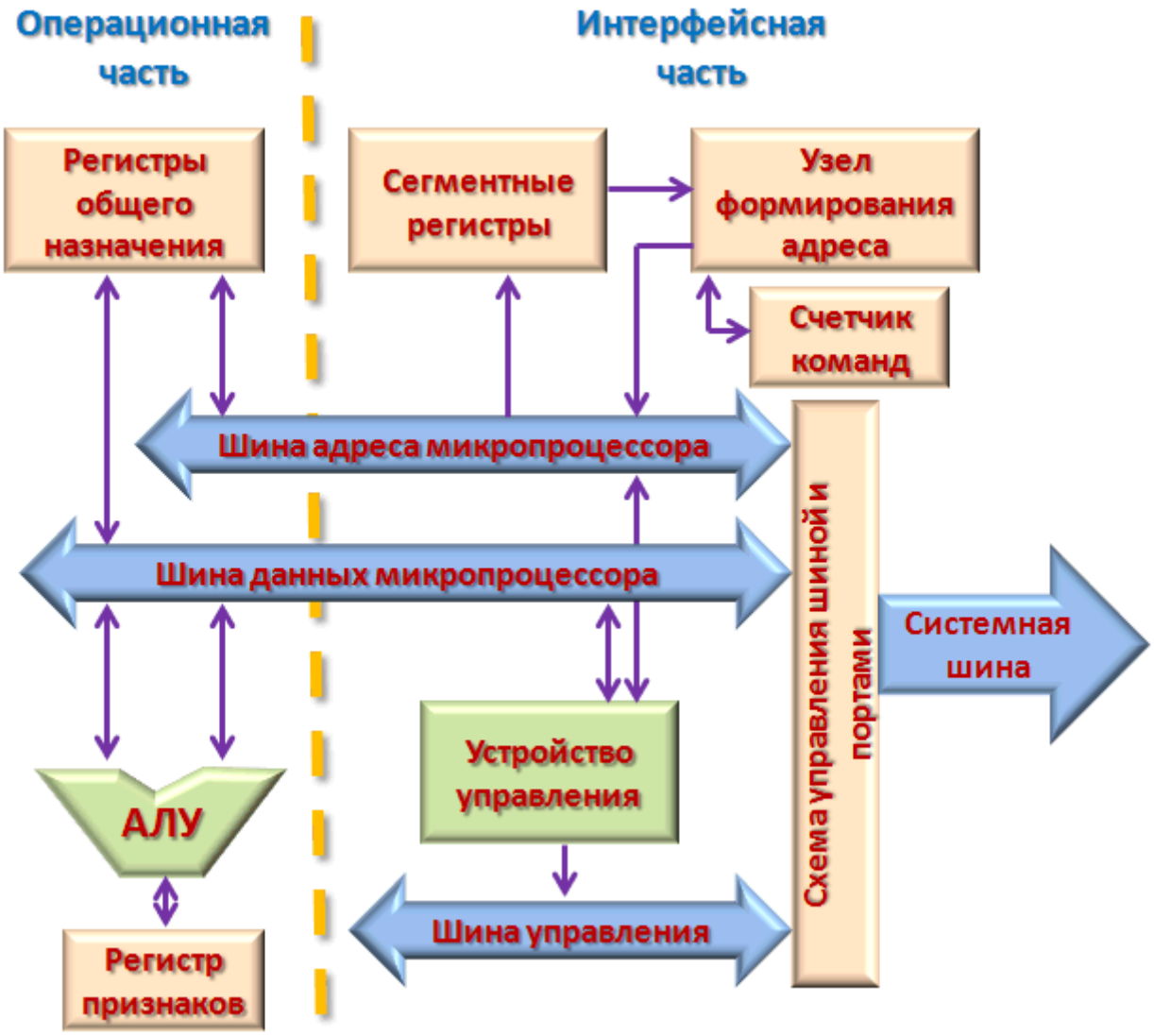
Большинство современных центральных процессоров строятся на базе 32-битной архитектуры Intel-совместимых процессоров IA-32 (Intel Architecture), которая является третьим поколением базовой архитектуры x86.

Структура центрального процессора

Функционально центральный процессор можно разделить на две части:

операционную, содержащую арифметико-логическое устройство (АЛУ) и микропроцессорную память (МПП) - регистры общего назначения;

интерфейсную, содержащую адресные регистры, устройство управления, регистры памяти для хранения кодов команд, выполняемых в ближайшие такты; схемы управления шиной и портами.



Обе части ЦП работают параллельно, причем интерфейсная часть опережает операционную, так что выборка очередной команды из памяти (ее запись в блок регистров команд и предварительный анализ) происходит во время выполнения операционной частью предыдущей команды. Такая организация ЦП позволяет существенно повысить его эффективное быстродействие.

Устройство управления (УУ) вырабатывает управляющие сигналы, поступающие по

кодовым шинам инструкций в другие блоки вычислительной машины. УУ формирует управляющие сигналы для выполнения команд центрального процессора.
Арифметико-логическое устройство (АЛУ) предназначено для выполнения арифметических и логических операций преобразования информации.
Системная шина – набор проводников, по которым передаются сигналы, соединяющая процессор с другими компонентами на системной плате. Системная шина состоит из шины данных, шины адреса, шины управления.
Шина данных – служит для пересылки данных между процессором и оперативным запоминающим устройством (ОЗУ).
Шина адреса – используется для передачи сигналов, с помощью которых определяется местоположение ячейки памяти для выполняемых процессором операций чтения/записи и ввода-вывода.
Шина управления – служит для пересылки управляющих сигналов. Каждая линия этой шины имеет своё особое назначение, поэтому они могут быть как однонаправленными, так и двунаправленными.

2.6 Организация памяти

Главная задача компьютерной системы – выполнять программы. Программы вместе с данными, к которым они имеют доступ, в процессе выполнения должны (по крайней мере частично) находиться в оперативной памяти. Операционной системе приходится решать задачу распределения памяти между пользовательскими процессами и компонентами ОС. Эта деятельность называется управлением памятью. Таким образом, память (storage, memory) является важнейшим ресурсом, требующим тщательного управления. В недавнем прошлом память была самым дорогим ресурсом. Часть ОС, которая отвечает за управление памятью, называется менеджером памяти.
Физическая организация памяти компьютера
Запоминающие устройства компьютера разделяют, как минимум, на два уровня: основную (главную, оперативную, физическую) и вторичную (внешнюю) память. Основная память представляет собой упорядоченный массив однобайтовых ячеек, каждая из которых имеет свой уникальный адрес (номер). Процессор извлекает команду из основной памяти, декодирует и выполняет ее. Для выполнения команды могут потребоваться обращения еще к нескольким ячейкам основной памяти. Обычно основная память изготавливается с применением полупроводниковых технологий и теряет свое содержимое при отключении питания.
Вторичную память (это главным образом диски) также можно рассматривать как одномерное линейное адресное пространство, состоящее из последовательности байтов. В отличие от оперативной памяти, она является энергонезависимой, имеет существенно большую емкость и используется в качестве расширения основной памяти.
Эту схему можно дополнить еще несколькими промежуточными уровнями, как показано на рис. 8.1. Разновидности памяти могут быть объединены в иерархию по убыванию времени доступа, возрастанию цены и увеличению емкости.

Рис. 8.1. Иерархия памяти

Многоуровневую схему используют следующим образом. Информация, которая находится в памяти верхнего уровня, обычно хранится также на уровнях с большими номерами. Если процессор не обнаруживает нужную информацию на i-м уровне, он начинает искать ее на следующих уровнях. Когда нужная информация найдена, она переносится в более быстрые уровни.

Локальность

Оказывается, при таком способе организации по мере снижения скорости доступа к уровню памяти снижается также и частота обращений к нему.

Ключевую роль здесь играет свойство реальных программ, в течение ограниченного отрезка времени способных работать с небольшим набором адресов памяти. Это эмпирически наблюдаемое свойство известно как принцип локальности или локализации обращений.

Свойство локальности (соседние в пространстве и времени объекты характеризуются похожими свойствами) присуще не только функционированию ОС, но и природе вообще. В случае ОС свойство локальности объяснимо, если учесть, как пишутся программы и как хранятся данные, то есть обычно в течение какого-то отрезка времени ограниченный фрагмент кода работает с ограниченным набором данных. Эту часть кода и данных удается разместить в памяти с быстрым доступом. В результате реальное время доступа к памяти определяется временем доступа к верхним уровням, что и обусловливает эффективность использования иерархической схемы. Надо сказать, что описываемая организация вычислительной системы во многом имитирует деятельность человеческого мозга при переработке информации. Действительно, решая конкретную проблему, человек работает с небольшим объемом информации, храня не относящиеся к делу сведения в своей памяти или во внешней памяти (например, в книгах).

Кэш процессора обычно является частью аппаратуры, поэтому менеджер памяти ОС занимается распределением информации главным образом в основной и внешней памяти компьютера. В некоторых схемах потоки между оперативной и внешней памятью регулируются программистом (см. например, далее оверлейные структуры), однако это связано с затратами времени программиста, так что подобную деятельность стараются возложить на ОС.

Адреса в основной памяти, характеризующие реальное расположение данных в физической памяти, называются физическими адресами. Набор физических адресов, с которым работает программа, называют физическим адресным пространством.

Логическая память

Аппаратная организация памяти в виде линейного набора ячеек не соответствует представлениям программиста о том, как организовано хранение программ и данных. Большинство программ представляет собой набор модулей, созданных независимо друг от

друга. Иногда все модули, входящие в состав процесса, располагаются в памяти один за другим, образуя линейное пространство адресов. Однако чаще модули помещаются в разные области памяти и используются по-разному.
Схема управления памятью, поддерживающая этот взгляд пользователя на то, как хранятся программы и данные, называется сегментацией. Сегмент – область памяти определенного назначения, внутри которой поддерживается линейная адресация. Сегменты содержат процедуры, массивы, стек или скалярные величины, но обычно не содержат информацию смешанного типа.
По-видимому, вначале сегменты памяти появились в связи с необходимостью обобществления процессами фрагментов программного кода (текстовый редактор, тригонометрические библиотеки и т. д.), без чего каждый процесс должен был хранить в своем адресном пространстве дублирующую информацию. Эти отдельные участки памяти, хранящие информацию, которую система отображает в память нескольких процессов, получили название сегментов. Память, таким образом, перестала быть линейной и превратилась в двумерную. Адрес состоит из двух компонентов: номер сегмента, смещение внутри сегмента. Далее оказалось удобным размещать в разных сегментах различные компоненты процесса (код программы, данные, стек и т. д.). Попутно выяснилось, что можно контролировать характер работы с конкретным сегментом, приписав ему атрибуты, например права доступа или типы операций, которые разрешается производить с данными, хранящимися в сегменте.

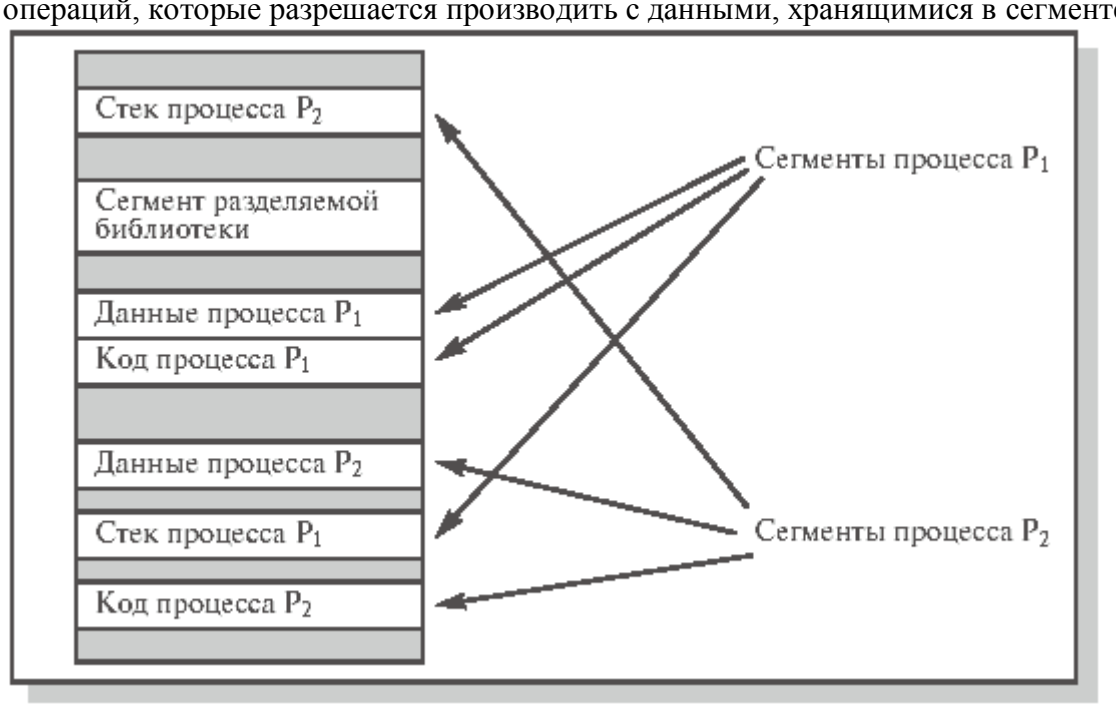


Рис. 8.2. Расположение сегментов процессов в памяти компьютера
Некоторые сегменты, описывающие адресное пространство процесса, показаны на рис. 8.2. Более подробная информация о типах сегментов имеется в лекции 10.
Большинство современных ОС поддерживают сегментную организацию памяти. В некоторых архитектурах (Intel, например) сегментация поддерживается оборудованием. Адреса, к которым обращается процесс, таким образом, отличаются от адресов, реально существующих в оперативной памяти. В каждом конкретном случае используемые программой адреса могут быть представлены различными способами. Например, адреса в исходных текстах обычно символические. Компилятор связывает эти символические адреса с перемещаемыми адресами (такими, как n байт от начала модуля). Подобный адрес, сгенерированный программой, обычно называют логическим (в системах с виртуальной памятью он часто называется виртуальным) адресом. Совокупность всех логических адресов называется логическим (виртуальным) адресным пространством.