

# Вычислительные системы, сети и телекоммуникации

## Функциональная и структурная организация ЭВМ

Принципы функционирования ЭВМ, т.е. коды, система команд, алгоритмы выполнения машинных операций, технология выполнения различных процедур и взаимодействия аппаратной части и программного обеспечения, способы использования устройств при организации их совместной работы, образуют **функциональную организацию ЭВМ.**

# Вычислительные системы, сети и телекоммуникации

## Функциональная и структурная организация ЭВМ

Способы реализации функций ЭВМ составляют **структурную организацию ЭВМ**

Элементная база, функциональные узлы и устройства ЭВМ, программные модули различных видов (обработчики прерываний, драйверы, com-, exe-, tsr-программы, bat-файлы и др.) являются структурными компонентами ЭВМ.

Принципы функционирования ЭВМ  
могут быть реализованы:

- аппаратными,
- программно-аппаратными
- программными средствами.

# Вычислительные системы, сети и телекоммуникации

## Функциональная и структурная организация ЭВМ

При аппаратной и программно-аппаратной реализации могут быть применены регистры, дешифраторы, сумматоры; блоки жесткого аппаратного управления или микропрограммного с управлением подпрограммами (комплексами микроопераций); устройства или комплексы устройств, реализованные в виде автономных систем (программируемых или с жестким управлением) и др.

При программной реализации могут быть применены различные виды программ.

## Функциональная и структурная организация ЭВМ

ЭВМ представляет собой совокупность устройств, выполненных на больших интегральных схемах, каждая из которых имеет свое функциональное назначение.

Комплект интегральных схем, из которых состоит ЭВМ, называется **микропроцессорным комплектом.**

# Вычислительные системы, сети и телекоммуникации

## Функциональная и структурная организация ЭВМ

В основной состав микропроцессорных комплектов входят:

- системный таймер,
- микропроцессор,
- сопроцессоры,
- контроллер прерываний,
- контроллер прямого доступа к памяти,
- контроллеры устройств ввода-вывода.

Все устройства ЭВМ делятся на  
*центральные и периферийные.*

Центральные устройства — полностью электронные.

Периферийные устройства могут быть либо электронными, либо электромеханическими с электронным управлением.

## Функциональная и структурная организация ЭВМ

Системная магистраль обычно состоит из трех шин:

- шина данных (ШД),
- шина адреса (ША),
- шина управления (ШУ).

В состав системной магистрали входят :

- регистры-защелки, в которых запоминается передаваемая информация,
- шинные формирователи, - шинные арбитры, определяющие очередность доступа к системной магистрали,
- и др.

## Функциональная и структурная организация ЭВМ

Логика работы системной магистрали, количество разрядов (линий) в шинах данных, адреса и управления, порядок разрешения конфликтных ситуаций, возникающих при одновременном обращении различных устройств ЭВМ к системной магистрали, образуют **интерфейс системной шины.**

## Функциональная и структурная организация ЭВМ

В состав центральных устройств ЭВМ

входят:

- центральный процессор,
- основная память
- ряд дополнительных узлов, выполняющих служебные функции:
  - контроллер прерываний,
  - таймер и контроллер прямого доступа к памяти (ПДП).

# Вычислительные системы, сети и телекоммуникации

## Функциональная и структурная организация ЭВМ

**Периферийные устройства** делятся на два вида:

- внешние запоминающие устройства (ЗУ)
  - НМД
  - НГМД
  - НМЛ
- устройства ввода-вывода (УВВ):
  - клавиатура
  - дисплей
  - принтер
  - мышь
  - адаптер каналов связи (КС) и др.

## Алгоритм работы ЭВМ

- Загрузка выполняемой программы
- Передача адреса 1 выполняемой команды в счетчик команд
- Адрес из счетчика команд на шину адреса системной магистрали (СМ)
- На шину управления – «Выборка из оперативной памяти (ОП)»
- Содержимое соотв. ячейки ОП на шину данных
- Одновременно на шину управления сигнал о выполнении выборки из ОП
- Процессор считывает с шины данных и передает в регистр команд
- В регистре команд – разделение на кодовую и адресную части
- Код команды настраивает МП на выполнение заданной операции
- Адрес следующей команды заносится в счетчик команд
- Адресная часть команды выставляется на шину адреса СМ
- Одновременно на шину управления – «Выборка из ОП»
- Выбранная из ОП информация вводится в арифметическое устройство (АУ) и начинается ее выполнение в АУ.
- Результат на шину данных, на шину адреса – адрес ОП, на шину управления – «Запись в ОП»

## Алгоритм работы ЭВМ

В регистре команд полученная команда разделяется на кодовую и адресную части.

Код команды поступает в блок управления для выработки сигналов, настраивающих МП на выполнение заданной операции и для определения адреса следующей команды (который сразу заносится в счетчик команд).

Адресная часть команды выставляется на шину адреса системной магистрали (СМ) и сопровождается сигналом “Выборка из ОП” на шине управления.

## Алгоритм работы ЭВМ

Выбранная из ОП информация через шину данных поступает на внутреннюю магистраль МП, с которой вводится в арифметическое устройство (АУ) и начинается ее выполнение в АУ.

Результат выполнения операции выставляется микропроцессором на шину данных, на шину адреса выставляется адрес ОП, по которому этот результат необходимо записать, а на шину управления выставляется команда “Запись в ОП”.

## Алгоритм работы ЭВМ

ОП считывает адрес и данные с системной магистрали, организует запись данных по указанному адресу и после выполнения команды выставляет на шину управления сигнал, обозначающий, что число записано. Процессор, получив этот сигнал, начинает выборку очередной команды: выставляет адрес из счетчика команд на шину адреса, формирует команду “Выборка из ОП” на шине управления и т.д.

## Алгоритм работы ЭВМ

В каждом цикле, получив команду в регистр команд и выделив код операции, процессор определяет, к какому устройству она относится.

Если команда должна выполняться процессором, организуется ее выполнение по описанному циклу. Если же команда предназначена для выполнения в другом устройстве ЭВМ, ЦП передает ее соответствующему устройству.

## Алгоритм работы ЭВМ

Процесс передачи команды другому устройству предусматривает следующие действия:

- ЦП выставляет на шину адреса СМ адрес интересующего его устройства;
- по шинам управления передается сигнал “Поиск устройства”;
- все устройства, подключенные к системной магистрали, получив этот сигнал, читают номер устройства с шины адреса и сравнивают его со своим номером. Устройства, для которых эти номера не совпадают, на эту команду не реагируют. Устройство с совпавшим номером, вырабатывает сигнал отклика по шине управления;
- ЦП, получив сигнал отклика, в простейшем случае выставляет имеющуюся у него команду на шину данных и сопровождает ее по шине управления сигналом “Передаю команду”;
- получив сигнал о приеме команды, ЦП переходит к выполнению очередной своей команды, выставляя на шину адреса содержимое счетчика команд.

## Алгоритм работы ЭВМ

### Интерфейс ввода-вывода

логическая последовательность действий  
вместе с устройствами, реализующими ее.

## Интерфейс ввода-вывода

Передача информации из периферийного устройства (ПУ) в центральные устройства (ЦУ) ЭВМ - *операция ввода.*

Передача из ЦУ ЭВМ в ПУ — *операцией вывода.*

## Интерфейс ввода-вывода

При разработке систем ввода-вывода ЭВМ особое внимание обращается на решение следующих проблем:

- ❖ обеспечение возможности реализации машин с переменным составом оборудования, в первую очередь с различным набором периферийных устройств, с тем чтобы пользователь мог выбирать состав оборудования (конфигурацию) машины в соответствии с ее назначением, легко дополнять машину новыми устройствами;
- ❖ одновременная работа процессора над программой и выполнение периферийными устройствами процедур ввода-вывода;
- ❖ простота для пользователя и стандартизованность программирования операций ввода-вывода, независимость программирования ввода-вывода от особенностей того или иного периферийного устройства;
- ❖ автоматическое распознавание и реакция ядра ЭВМ на многообразие ситуаций, возникающих в ПУ (готовность устройства, отсутствие носителя, различные нарушения нормальной работы и др.).

## Интерфейс ввода-вывода

В общем случае для организации и проведения обмена данными между двумя устройствами требуются специальные средства:

- специальные управляющие сигналы и их последовательности;
- устройства сопряжения;
- линии связи;
- программы, реализующие обмен.

Весь этот комплекс линий и шин, сигналов, электронных схем, алгоритмов и программ, предназначенный для осуществления обмена информацией, называется ***интерфейсом***.

## Интерфейс ввода-вывода

### Стандартный интерфейс

- совокупность унифицированных технических программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных элементов в автоматических системах обработки информации при условиях, предписанных стандартом и направленных на обеспечение информационной, электрической и конструктивной совместимости указанных элементов.

## Классификация интерфейса ввода-вывода

В зависимости от использования можно выделить 3 типа интерфейса:

- машинно-ориентированный
- системный
- приборный

## Классификация интерфейса ввода-вывода

- Машинно-ориентированный интерфейс - интерфейсы, которые решают задачу подключения конкретного периферийного устройства (УВВ) к ЭВМ конкретного типа.
- Системный интерфейс - сопряжение устройств (модулей), имеющих системное назначение и исполнение, т.е. основное назначение которых создание системы. Системный интерфейс не решает конкретную задачу ввода/вывода, а, прежде всего, стандартизует устройства для решения этой задачи.
- Приборный интерфейс предназначен для соединения различных приборов, в т.ч. работающих автономно от ЭВМ.

## Классификация интерфейса ввода-вывода

В зависимости от типа соединяемых устройств различаются:

- внутренний интерфейс ЭВМ (например, интерфейс системной шины, НМД), предназначенный для сопряжения элементов внутри системного блока ПЭВМ;(внутриплатный, межплатный и межблочный)
- интерфейс ввода-вывода — для сопряжения различных устройств с системным блоком (клавиатурой, принтером, сканером, мышью, дисплеем и др.);
- интерфейсы межмашинного обмена (для обмена между разными машинами) — для сопряжения различных ЭВМ (например, при образовании вычислительных сетей);
- интерфейсы «человек — машина» — для обмена информацией между человеком и ЭВМ.

## Алгоритм работы ЭВМ

### Однопрограммный режим работы -

при обращении ЦП к внешнему устройству

продолжение выполнения основной программы

центральный процессором возможно только

после завершения операции ввода-вывода.

## Алгоритм работы ЭВМ

### Многопрограммный режим –

в один и тот же момент времени различные устройства ЭВМ выполняют либо разные программы, либо разные части одной и той же программы.

Задание для ЭВМ в виде программы на алгоритмическом языке - **ИСХОДНЫЙ МОДУЛЬ**.

Программа может сопровождаться управляющими предложениями, например, в виде bat-файла.

Управляющие предложения указывают операционной системе ЭВМ:

- на каком языке написана программа
- что с ней надо делать

Исходный модуль перед исполнением должен быть переведен на внутренний язык машины.

Эта операция выполняется специальной программой - **транслятором**.

Трансляторы выполняются в виде двух разновидностей:

- интерпретаторы
- компиляторы

Интерпретатор после перевода на язык машины каждого оператора алгоритмического языка немедленно исполняет полученную машинную программу.

Компилятор полностью переводит всю программу, представленную ему в виде исходного модуля (ИМ), на язык машины. Получаемая при этом машинная программа представляет собой **объектный модуль (ОМ)**, который не готов для исполнения

Результат работы компилятора может быть:

- записан в библиотеку объектных модулей (БОМ),

- передан другим программам для дальнейшей обработки, поскольку полученная машинная программа не готова к исполнению по двум причинам

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

---

## Организация работы ЭВМ при выполнении задания пользователя

1. **Неразрешенные внешние ссылки** (т.е. обращение к программам, которые не содержатся в исходном модуле, но необходимы для работы основной программы, например к стандартным программам алгоритмического языка, таким, как **извлечение квадратного корня, вычисление тригонометрических функций и т.д.**)
2. **Объектный модуль** представляет собой машинную программу в **условных адресах** - каждый объектный модуль начинается с «нулевого» адреса, тогда как для исполнения программа должна быть «привязана» к конкретным **физическим адресам** основной памяти

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

---

## Организация работы ЭВМ при выполнении задания пользователя

Недостающие программы должны быть взяты из **библиотек компилятора** (которые могут быть написаны в виде исходных, либо в виде объектных модулей) и добавлены к основной программе.

Эту операцию выполняет **редактор связей**.

В результате работы редактора связей образуется **загрузочный модуль (ЗМ)**, который помещается в соответствующую библиотеку программ (БЗМ).

В загрузочном модуле все ссылки разрешены, т.е. он содержит все необходимые стандартные программы, но привязки к памяти у загрузочного модуля нет

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

---

## Организация работы ЭВМ при выполнении задания пользователя

Привязка к памяти загрузочного модуля производится **программой выборки**, которая переносит загрузочный модуль из библиотеки загрузочных модулей (обычно хранящейся на магнитном носителе) в основную память.

Во время этого переноса корректируются адреса, учитывая, с какого адреса основной памяти размещается загрузочный модуль.

После перемещения загрузочного модуля в основную память программа выборки инициирует ее выполнение

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

---

## Организация работы ЭВМ при выполнении задания пользователя

Сокращение времени выполнения программы - за счет предварительных трансляции программы и редактировании связей.

При отладке программы используется **загрузчик** - программа, сочетающая в себе функции редактирования связей и загрузки полученной машинной программы в основную память для исполнения.

При использовании загрузчика многократные просчеты по программе проводить невыгодно, так как каждый раз приходится выполнять лишние операции редактирования связей

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Организация работы ЭВМ при выполнении задания пользователя

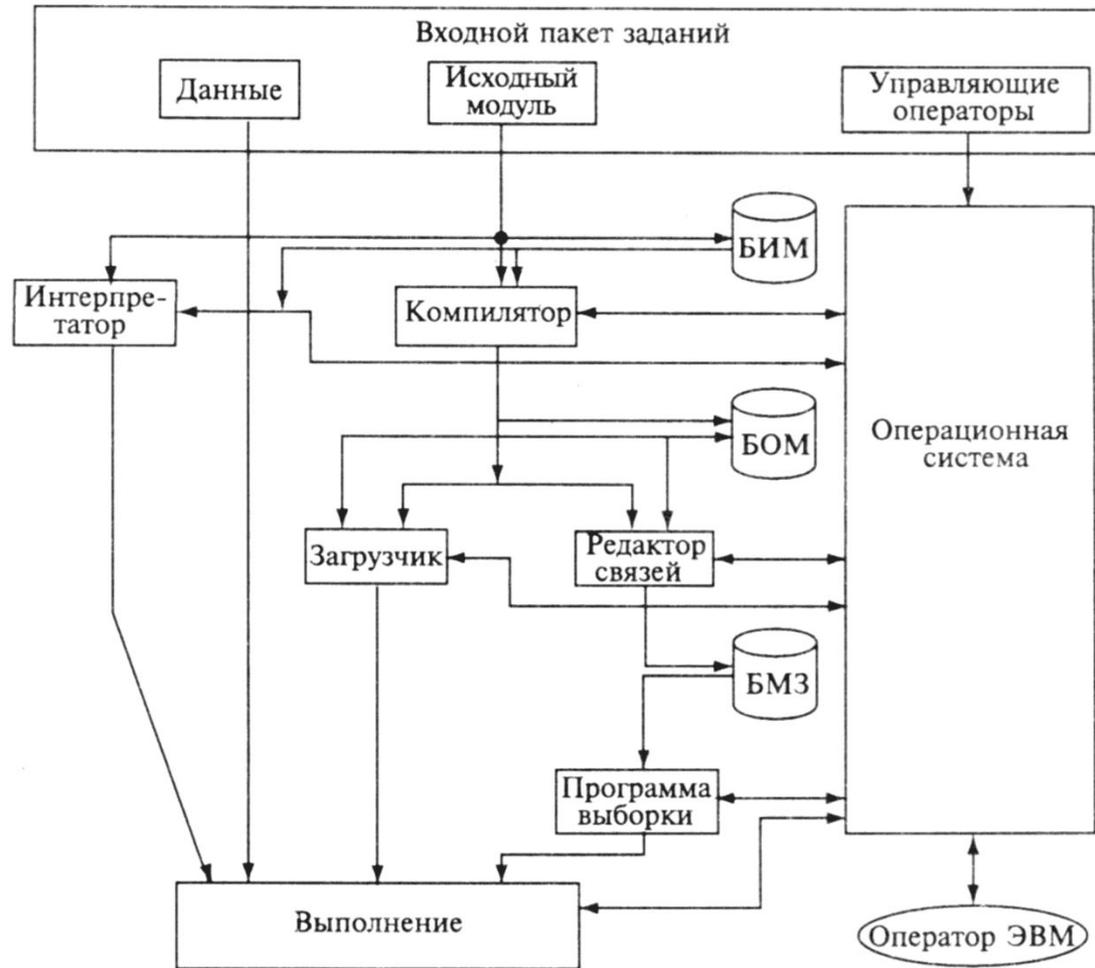


Рис. 1 - схема обработки заданий операционной системой

Отображение адресного пространства программы на основную память

Для выполнения программы при ее загрузке в основную память ей выделяется часть **машинных ресурсов**.

Они необходимы для размещения команд, данных, управляющих таблиц и областей ввода-вывода, т. е. производится трансляция адресного пространства откомпилированной программы в местоположение в реальной памяти

Отображение адресного пространства программы на основную память

Выделение ресурсов осуществляется:

- программистом (особенно, если он работает на языке, близком машинному)
- операционной системой

В зависимости от момента начала выделения ресурсов различают:

- статическое перемещение
- динамическое перемещение

Отображение адресного пространства программы на основную память

**Статическое перемещение** - выделение ресурсов производится перед выполнением программы, в результате которого программа «привязывается» к определенному месту в памяти вычислительной машины.

**Динамическое перемещение** - выделение ресурсов производится в процессе выполнения программы.

В этом случае программа не привязана к определенному месту в реальной памяти

При статическом перемещении могут встретиться два случая:

- реальная память больше требуемого адресного пространства программы
- реальная память меньше требуемого адресного пространства программы

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Отображение адресного пространства программы на основную память

Реальная память больше требуемого адресного пространства программы

Загружаемая программа А является абсолютной программой, так как никакого изменения адресов в адресном пространстве, подготовленном компилятором, при загрузке в основную память не происходит - программа располагается с 0-го адреса реальной памяти

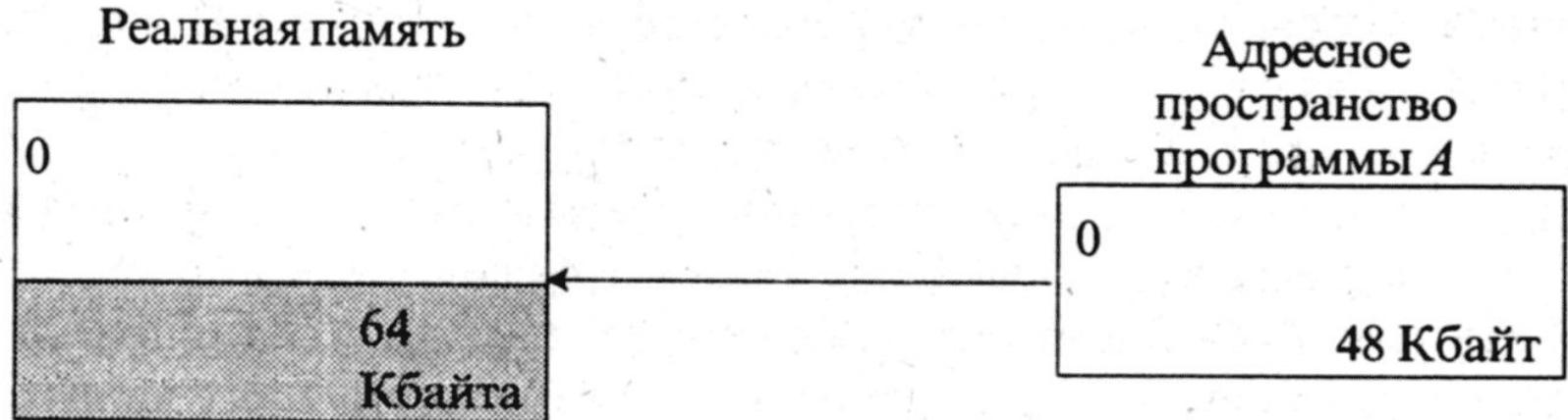


Рис. 2 - загрузка программы в избыточную реальную память

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

---

## Отображение адресного пространства программы на основную память

Реальная память меньше требуемого адресного пространства программы

Методы решения проблемы:

- создание **оверлейной структуры** (разбиение программы на части, вызываемые в ОП по мере необходимости)
- создание **реентерабельных модулей** программы (т.е. допускающими одновременную работу модуля по нескольким обращениям из разных частей программы или из различных программ)

Отображение адресного пространства программы на основную память

**Абсолютный модуль** - размещение программы в памяти всегда с одного и того же адреса, необязательно нулевого.

В этом случае могут возникнуть две ситуации:

- в памяти невозможно разместить одновременно все программы
- программы размещаются с «зазором»

## Отображение адресного пространства программы на основную память

При мультипрограммном режиме, если имеем программы А, В и С, для которых известно, что программа А выполняется при размещении в памяти с адреса 60 Кбайт до 90 Кбайт, В - с 60 Кбайт до 90 Кбайт, С - с 50 Кбайт до 120 Кбайт, организовать их совместное выполнение невозможно, так как им необходим один и тот же участок реальной памяти.

Эти программы будут ждать друг друга, либо их нужно заново редактировать с другого адреса

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Отображение адресного пространства программы на основную память

При работе в мультипрограммном режиме, особенно для систем со статическим перемещением, может сложиться ситуация, когда между программами образуются незанятые участки памяти - фрагментация памяти

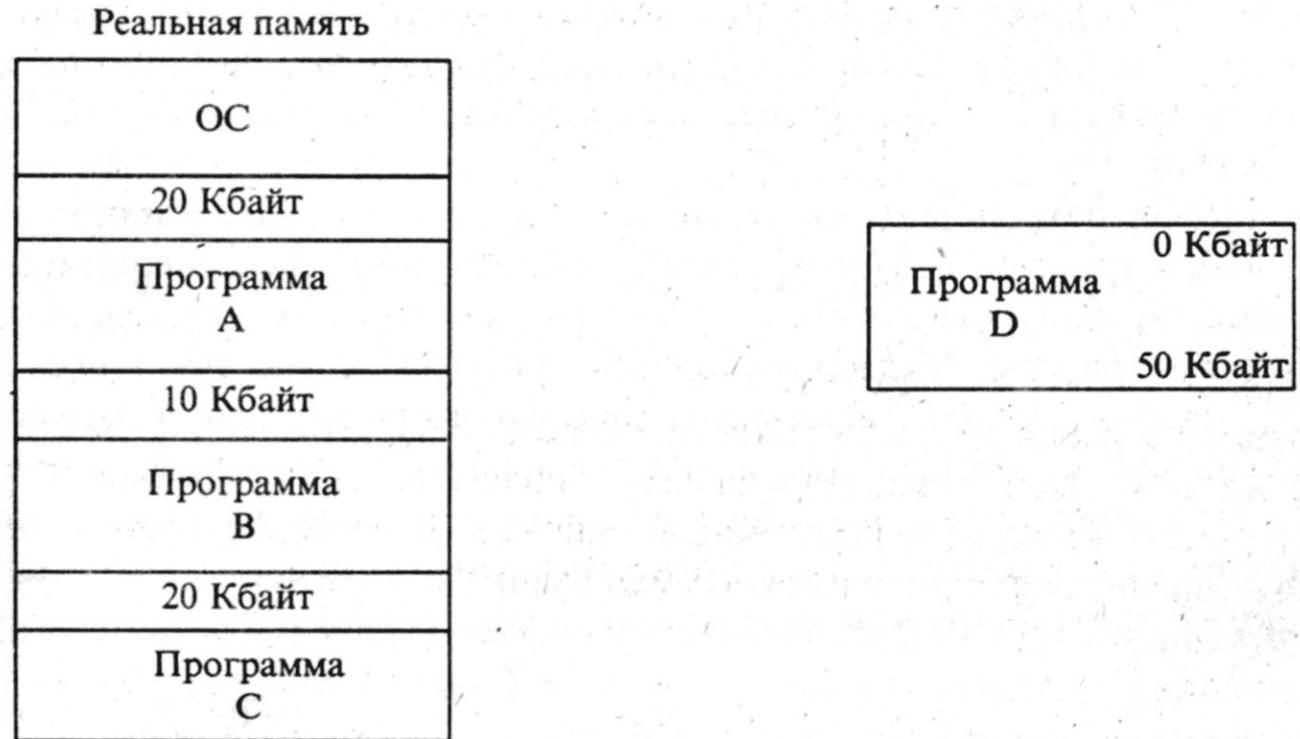


Рис. 3 - фрагментация реальной памяти

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Отображение адресного пространства программы на основную память

В системах с динамическим перемещением программ перемещающий загрузчик размещает программу в свободной части памяти и допускает использование ее несмежных участков

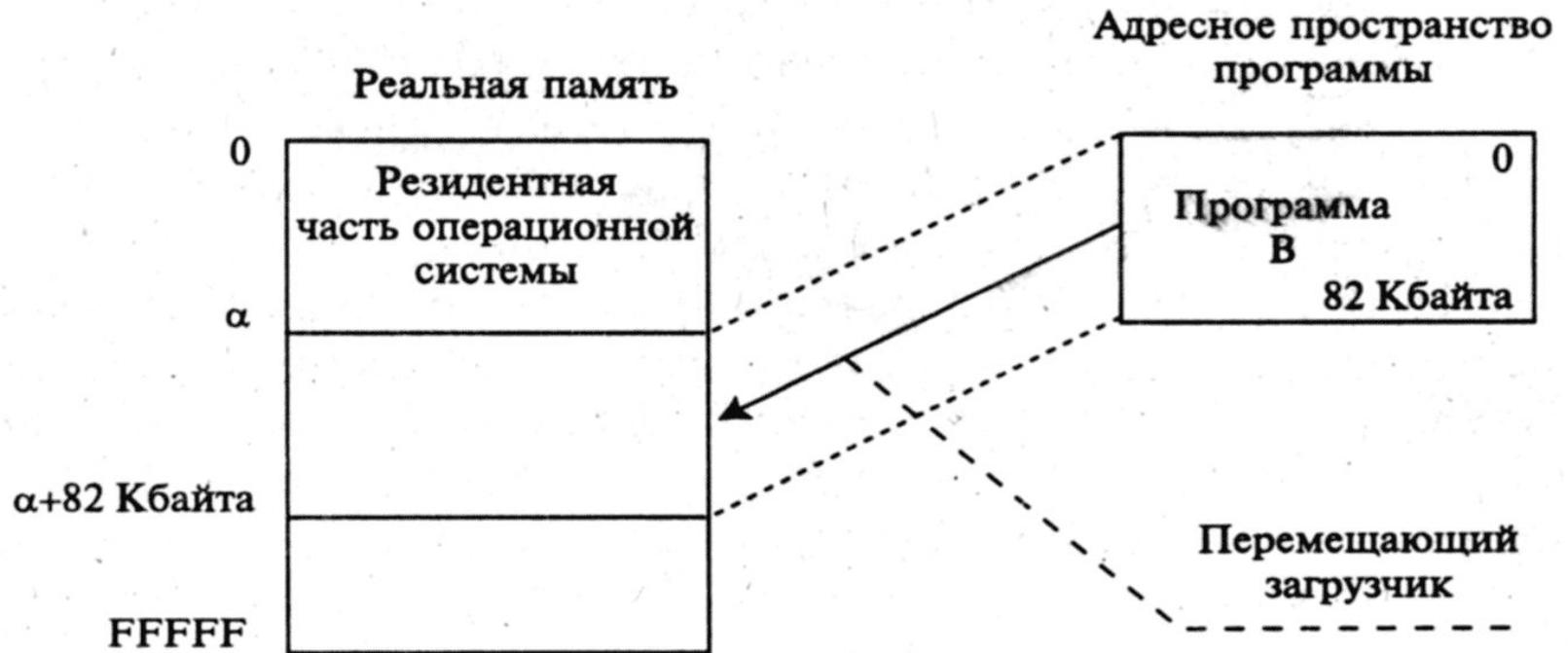


Рис. 4 - размещение программы в свободной части ОП

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Отображение адресного пространства программы на основную память

**Сегментирование программы** - метод борьбы с фрагментацией - адресное пространство программы может быть разбито на отдельные сегменты, слабо связанные между собой

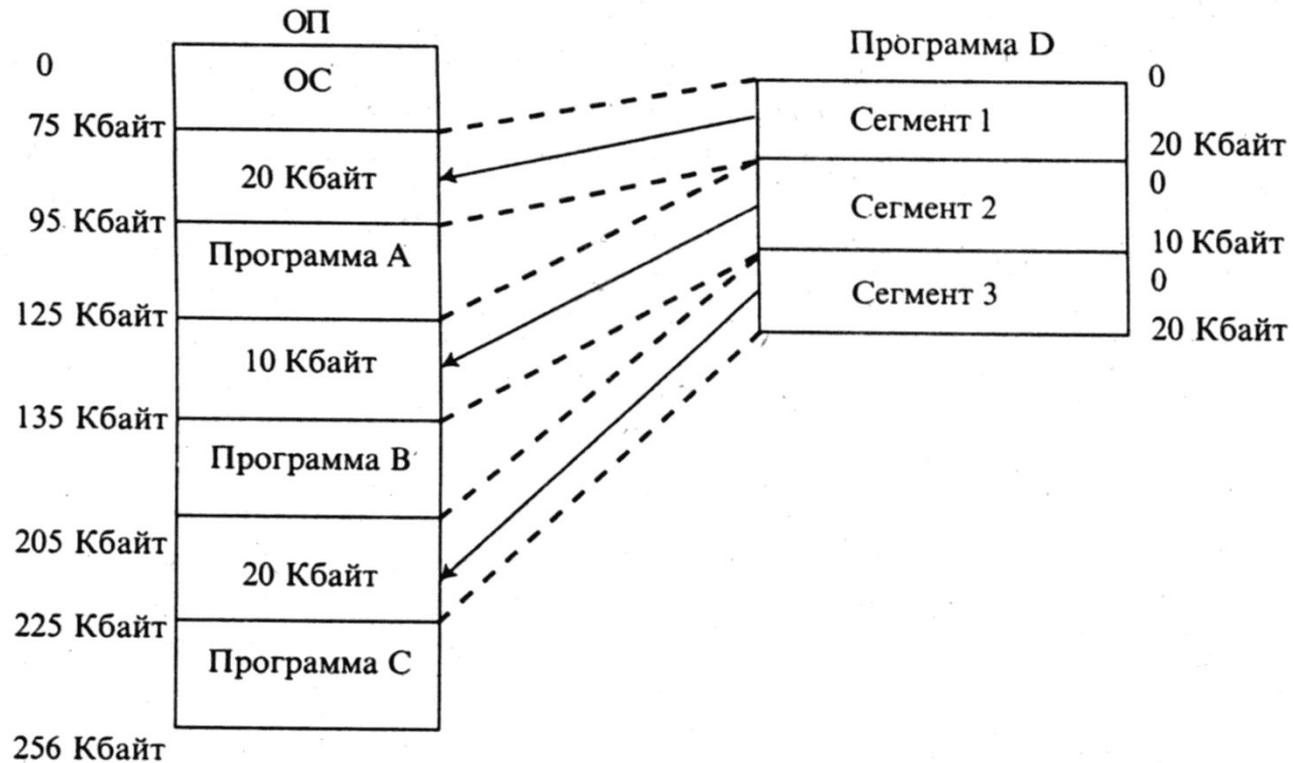


Рис. 5 - фрагментация ОП. Загрузка сегментированной программы

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Отображение адресного пространства программы на основную память

Адреса в каждом сегменте начинаются с 0.

**Статическое перемещение** - статическая трансляция адресов - неэффективно, т.к. надо каждый адрес предварительно преобразовать в неизменяемый реальный адрес основной памяти.

**Динамическое перемещение** - динамическая трансляция адресов (ДТА) - трансляция адресов каждой команды производится в процессе ее выполнения (процесс преобразования адресов подстраивается под сложившуюся ситуацию)

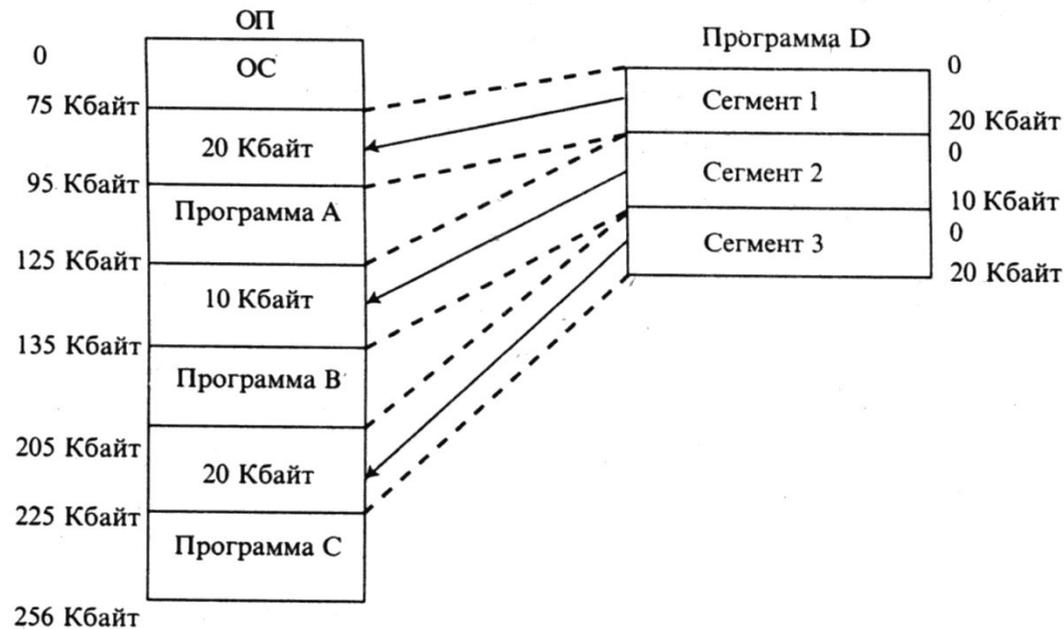
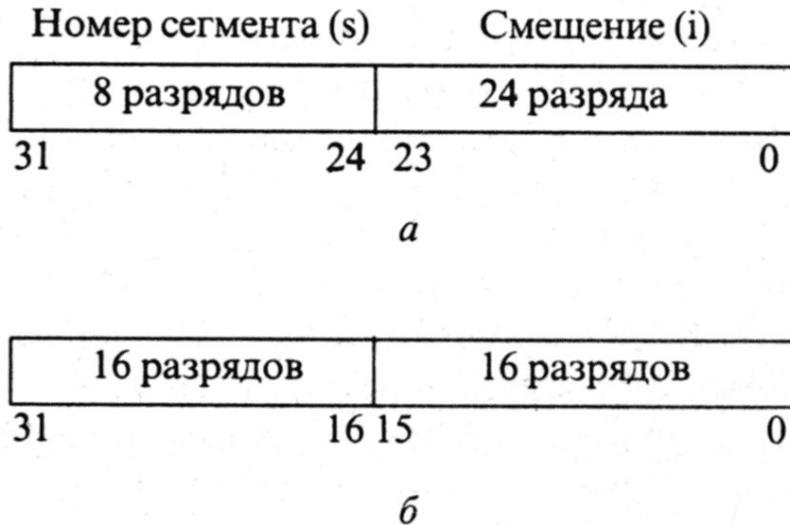


Рис. 5 - фрагментация ОП. Загрузка сегментированной программы

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Отображение адресного пространства программы на основную память



имя сегмента - номер,  
состоящий из двух частей:

$s, i,$

где  $s$  - номер сегмента,  $i$  - адрес  
внутри сегмента

Рис. 6 - форма имени сегмента: а - при выделении номеру сегмента 8 разрядов; б - при выделении номеру сегмента 16 разрядов

Отображение адресного пространства программы на основную память

Структура сегмента определяет:

- количество возможных сегментов
- максимальное количество адресов в сегменте (максимальное смещение любого адреса в сегменте)

При 32-разрядном имени сегмента:

- 8-разрядный номер сегмента: 256 сегментов и 4М адресов в каждом сегменте
- 16-разрядный номер сегмента: 64К сегментов и 64К адресов в каждом сегменте

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

## Отображение адресного пространства программы на основную память

Для динамической трансляции адресов (т.е. при определении абсолютных адресов по известным относительным, содержащим номер сегмента и смещение) операционная система строит специальные таблицы, устанавливающие соответствие между сегментируемым адресным пространством программы и действительными адресами сегментов в реальной памяти

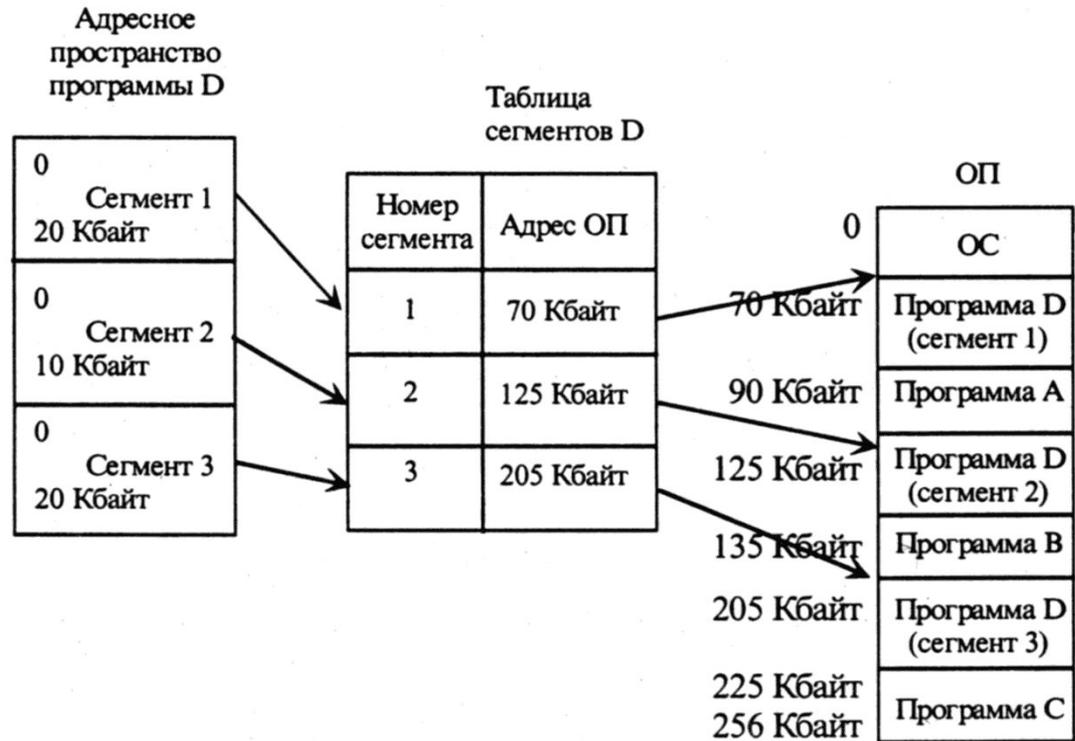


Рис. 7 - динамическая трансляция адресов при сегментной организации программы

# ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

---

## Отображение адресного пространства программы на основную память

Таблицу сегментов содержит каждая выполняемая программа.

В дополнение к таблице сегментов для динамической трансляции адреса используется специальный управляющий регистр, называемый **регистром начала таблицы сегментов** (PHTC или STOR - segment table origin register).

В этот регистр занесен адрес таблицы сегментов выполняемой в данный момент программы

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

---

---

Алгоритм перемещения исполняемой программы в другую часть памяти (для исключения фрагментации):

1. Переслать команды и данные сегмента в другую часть памяти.
2. Строку таблицы сегментов для данного сегмента нужно изменить так, чтобы она содержала новый начальный адрес
3. Выполнение программы может быть продолжено.

Это дает возможность динамического управления реальной памятью в процессе выполнения программы

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

---

---

Использованием сегментации программ - уменьшение фрагментации основной памяти.

Полностью фрагментация не устраняется - остаются фрагменты, длина которых меньше длины сегментов программы.

Следующий шаг - разделение сегментов на одну или несколько единиц, называемых **страницами**

Размер страницы достаточно мал по сравнению с обычным размером сегментов, неиспользуемые фрагменты ОП значительно сокращаются в объеме - будет иметь место так называемая фрагментация внутри страниц.

Потери - существенно меньше

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

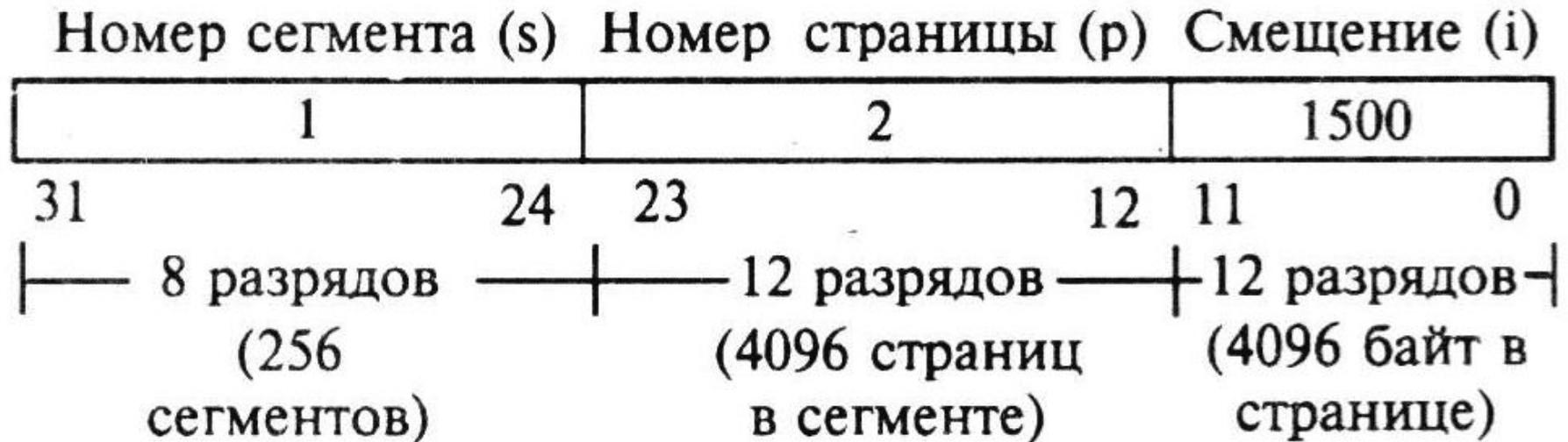
ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

Сегментно-страничная организация - еще один уровень в структуре адресного пространства программы.

Адресное пространство программы дробится на сегменты, внутри сегментов - на страницы, а внутри страниц - на адреса байтов.

Структура адреса: (S, P, i)

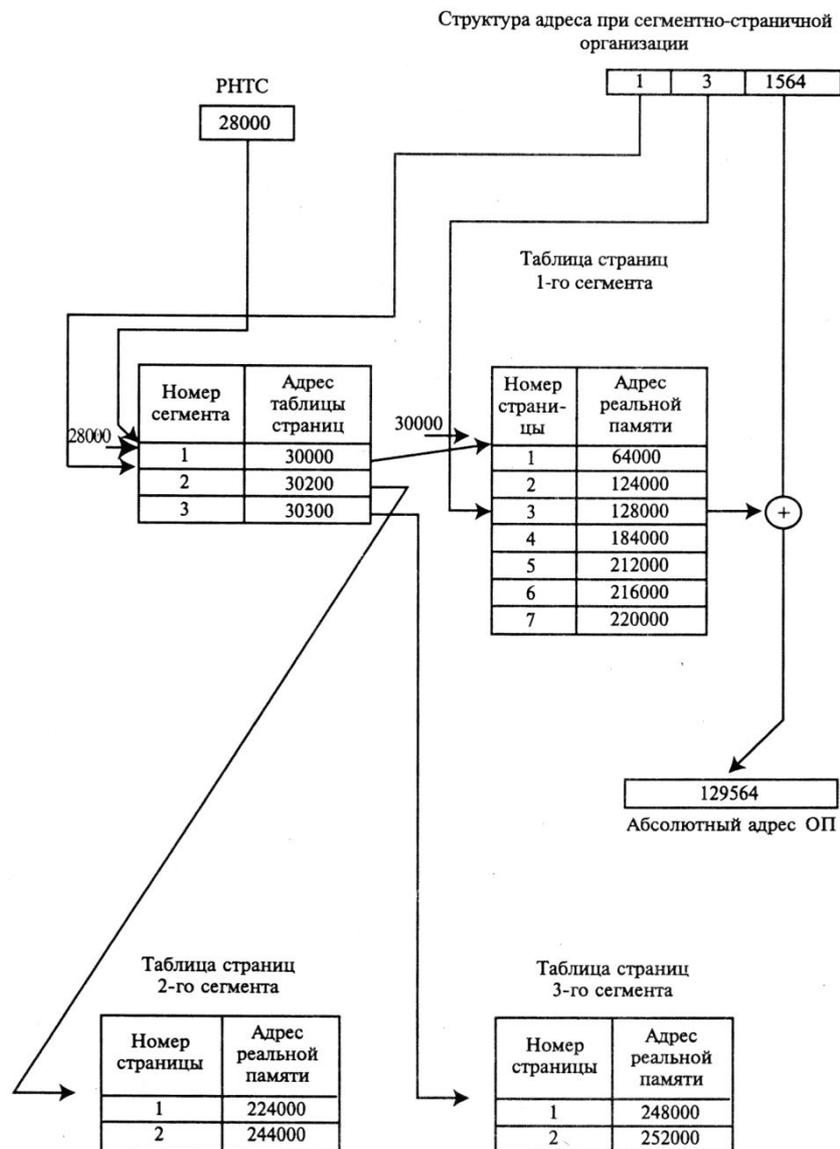


где  
S - имя сегмента внутри адресного пространства программы;  
P - имя внутри страницы;  
i - адрес внутри страницы

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

## ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

### ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ



# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

---

---

Виртуальная память - теоретически доступная пользователю основная память, объем которой определяется только разрядностью адресной части команды, но не существует в действительности.

Виртуальная память имеет сегментно-страничную или страничную организацию.

Размещается в

- страничных блоках основной памяти (page frames)
- ячейках внешней страничной памяти (slot).

Внешняя страничная память является частью внешней памяти(например, на жестком магнитном диске)

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

---

---

Загрузить программу в виртуальную память -  
значит переписать несколько программных страниц  
из внешней страничной памяти в основную память.  
Этот механизм называется **принудительным  
страничным обменом**

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

---

---

Максимальный размер виртуальной памяти определяется только длиной физического адреса (32 бита):  $2^{32} = 4\text{Гбайта}$ .

Размер страницы в IBM PC фиксирован - 4КБайта (для адресации байтов внутри страницы необходимо 12бит).

Адрес виртуальной памяти состоит из двух частей: номера страницы (20 бит) и смещения (12 бит)

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

---

---

Номера виртуальных страниц необходимо преобразовывать в номера физических страниц (слотов), в которых должны содержаться такие данные, как имя накопителя, номер цилиндра, номер головки, номер трека, номер сектора и т.д. Это преобразование осуществляется при помощи таблицы страниц

# ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

## ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

### ОТОБРАЖЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПРОГРАММЫ НА ОСНОВНУЮ ПАМЯТЬ

---

---

При 20-битном адресе номера страницы требуется линейная таблица, содержащая 1М элементов (элемент таблицы - 4байта, таблица - 4Мбайта).

Линейный адрес делится

не на две части (номер виртуальной страницы - 20бит, смещение - 12бит),  
а на три: каталог - 10бит, таблица - 10бит, смещение - 12бит.

В основной памяти:

- каталог
- активные таблицы страниц