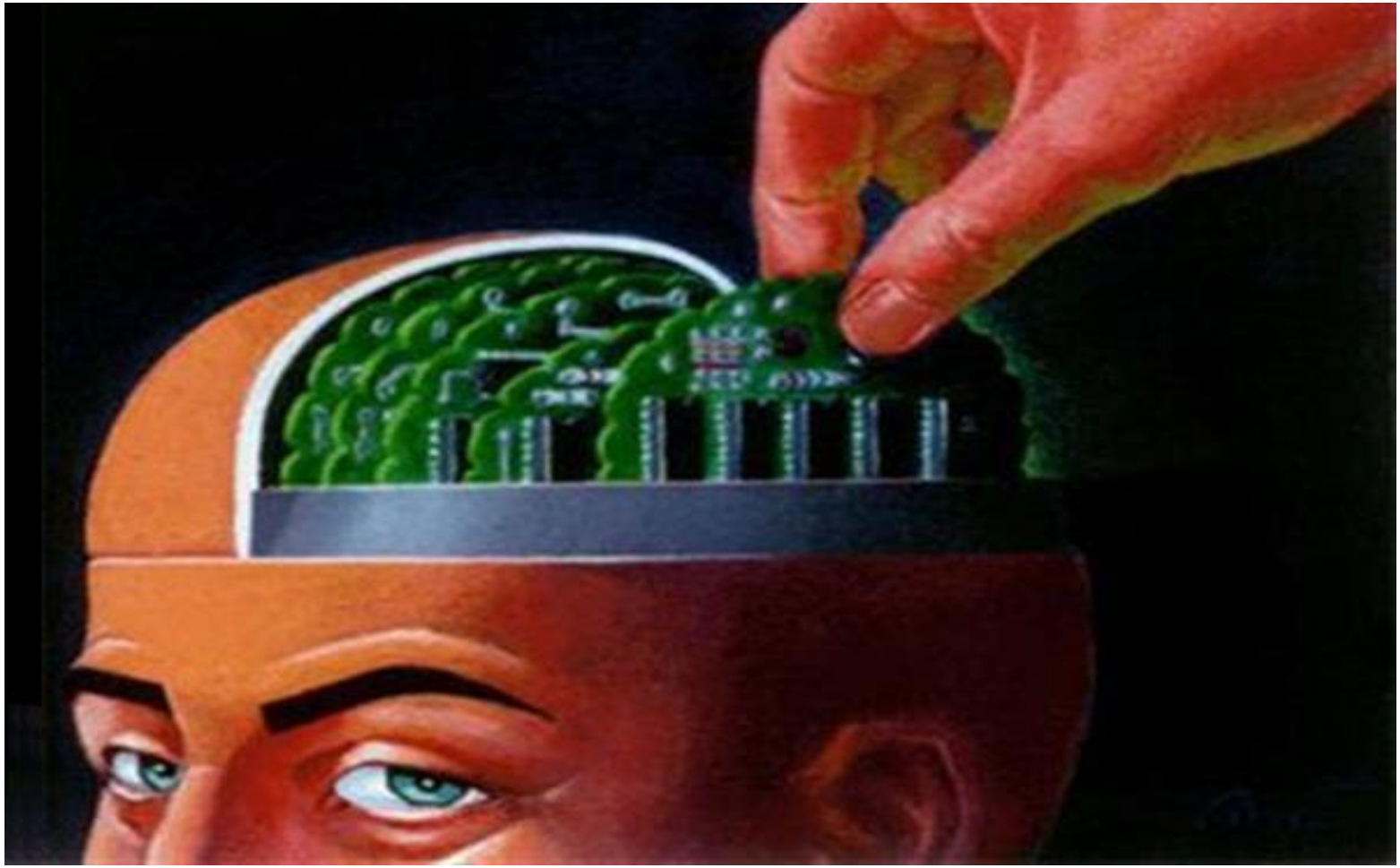
Операционные системы

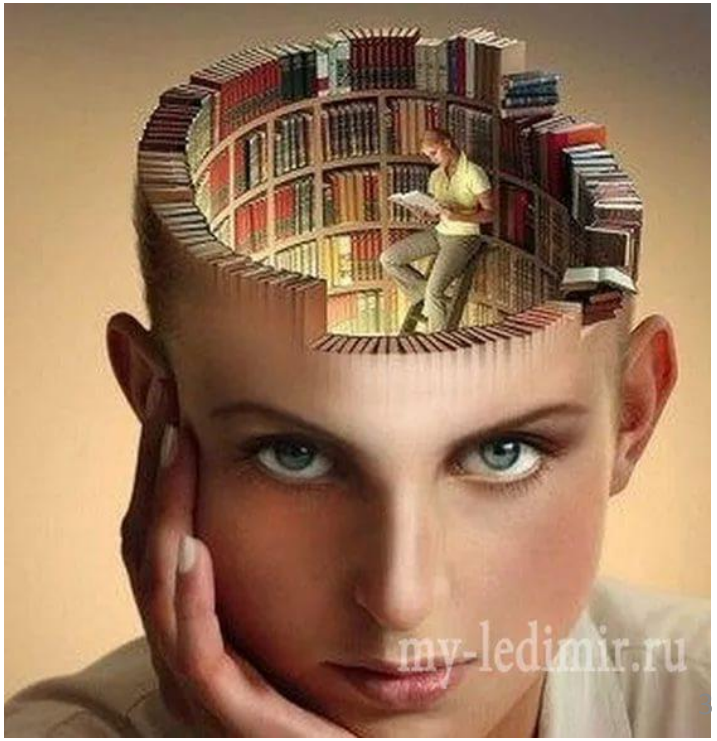
Тема 4. Управление памятью



Управление памятью

Память является важнейшим ресурсом, требующим тщательного управления со стороны мультизадачной операционной системы.

Распределению подлежит вся оперативная память, не занятая операционной системой.



Физическая память представляет собой множество ячеек, которые пронумерованы, к каждой ячейке можно обратиться, указав ее порядковый номер (адрес). Количество ячеек физической памяти ограничено и фиксировано.

Функции ОС по управлению памятью:

(1/2)

- отслеживание свободной и занятой памяти;
- выделение памяти процессам и освобождение памяти при завершении процессов;
- вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти не достаточны для размещения в ней всех процессов;
- возвращение процессов в оперативную память, настройка адресов программы на конкретную область физической памяти;

Функции ОС по управлению памятью:

(2/2)

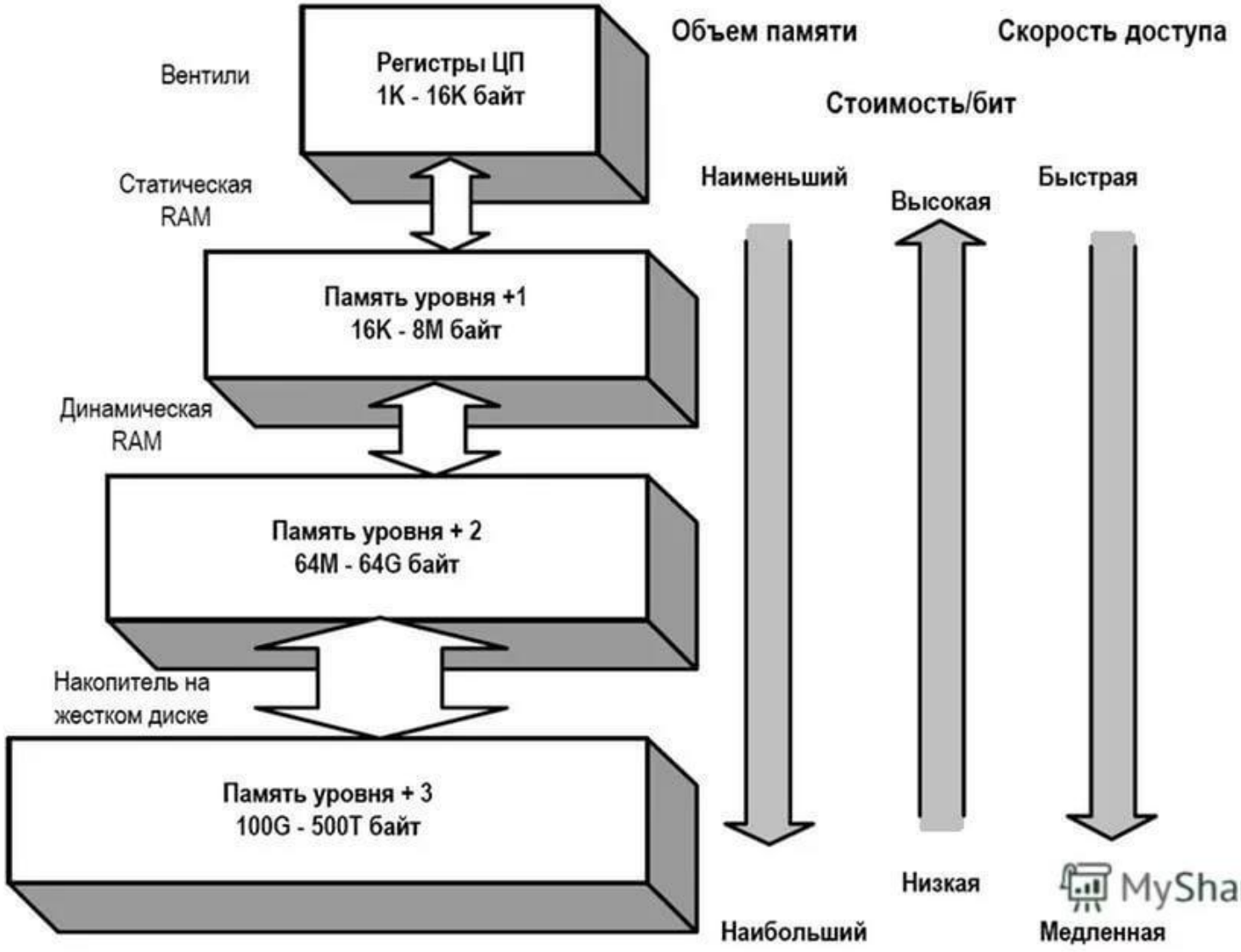
- распределение ресурса типа «память» между различными, конкурирующими за нее процессами (т.к. памяти всегда не хватает, это ограниченный ресурс по своей сути);
- защита доступа к области памяти одного процесса от другого;
- абстрагировать доступ к физической памяти для программистов.

Концепция иерархии памяти

Управление памятью базируется на ***концепция иерархии памяти***, согласно которой: компьютеры обладают несколькими мегабайтами очень быстродействующей, дорогой и энергозависимой памяти, несколькими гигабайтами памяти, средней как по скорости, так и по цене, а так-же несколькими терабайтами памяти на довольно медленных, сравнительно дешевых дисковых накопителях.

Иерархия памяти

Иерархия памяти в компьютере



Концепция иерархии памяти

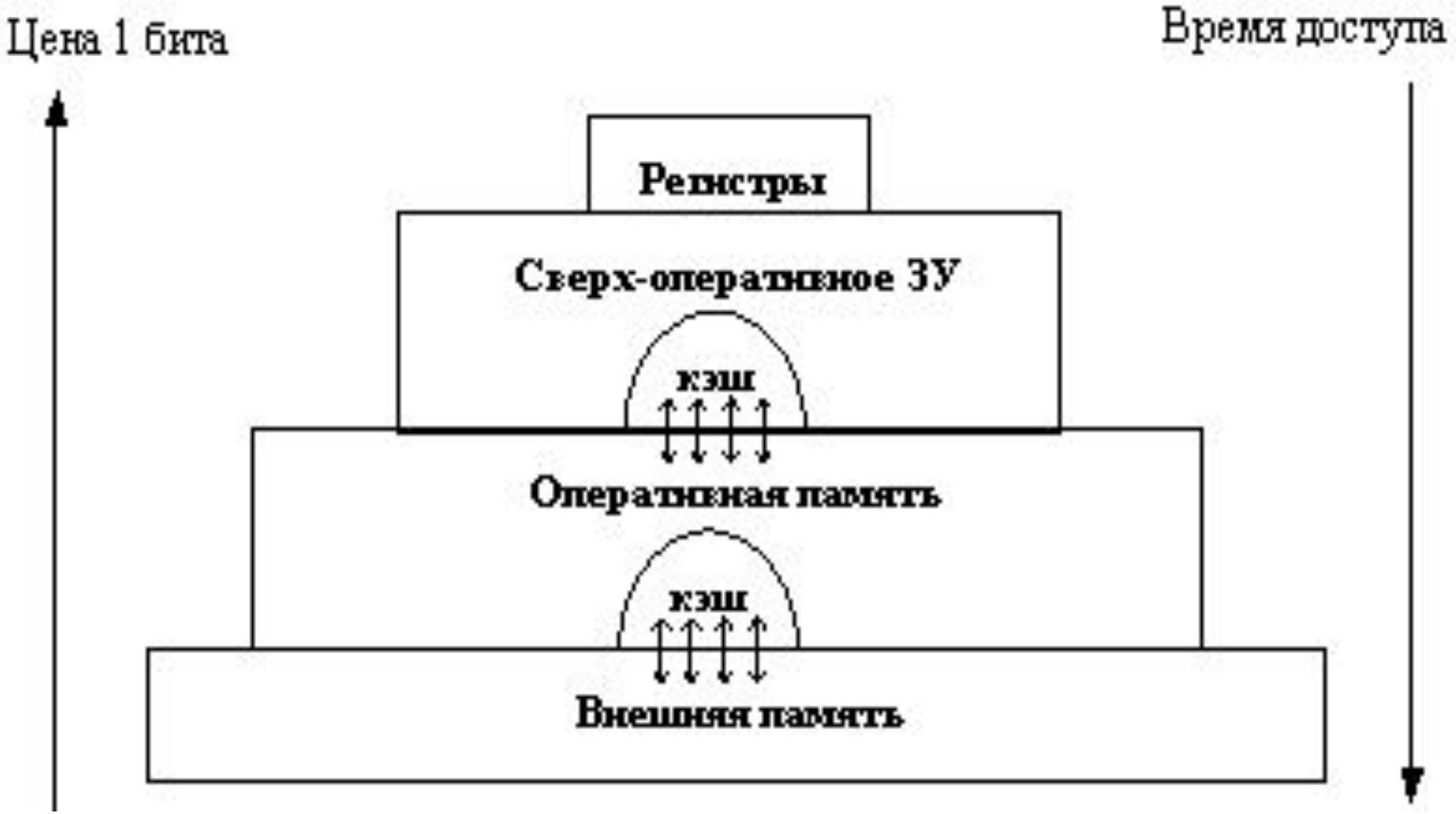
Превратить эту иерархию в абстракцию и управлять этой абстракцией — и есть задача операционной системы.

Часть операционной системы, которая управляет иерархией памяти, называется ***диспетчером памяти***.

Управление памятью

Кэш-память - это способ организации совместного функционирования двух типов запоминающих устройств, отличающихся временем доступа и стоимостью хранения данных, который позволяет уменьшить среднее время доступа к данным за счет динамического копирования в "быстрое" ЗУ наиболее часто используемой информации из "медленного" ЗУ.

Иерархия памяти



Алгоритм запроса к оперативной памяти

- Просматривается содержимое кэш-памяти с целью определения, не находятся ли нужные данные в кэш-памяти;
- Если данные обнаруживаются в кэш-памяти, то они считываются из нее, и результат передается в процессор;
- Если нужных данных нет, то они вместе со своим адресом копируются из оперативной памяти в кэш-память, и результат выполнения запроса передается в процессор.

Управление памятью

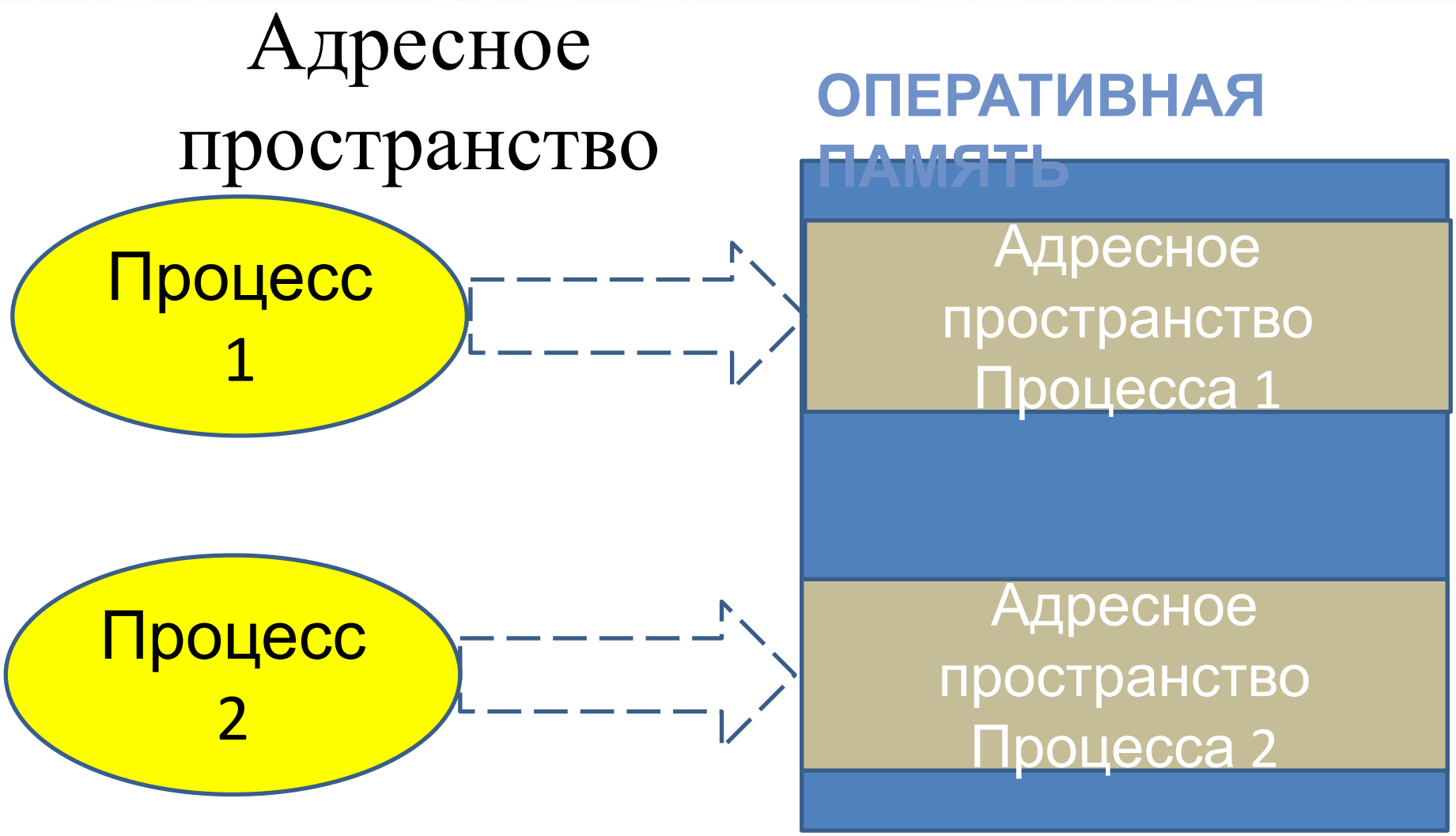
На практике в кэш-память считывается не один элемент данных, к которому произошло обращение, а целый блок данных, это увеличивает вероятность так называемого "попадания в кэш", то есть нахождения нужных данных в кэш-памяти.

Адресное пространство

Между процессом и памятью, выделенной для этого процесса, существует однозначное соответствие, которое вводится понятием *адресное пространство*.

Адресное пространство — это набор адресов, который может быть использован процессом для обращения к памяти. У каждого процесса имеется собственное адресное пространство, независимое от адресного пространства, другого процесса.

Управление памятью



Адресное пространство

Аналогия из бытовой сферы.

Возьмем телефонные номера. Во многих странах местный телефонный номер состоит обычно из семизначного числа. Поэтому адресное пространство телефонных номеров для конкретного города простирается от 0000000 до 9999999. Но имеются ещё коды страны и города, которые однозначно определяют адресное пространство телефонных номеров для конкретного города.

Дефицит памяти

На практике суммарный объем оперативной памяти, необходимый для размещения всех процессов, зачастую значительно превышает имеющийся объем ОЗУ.

Дефицит памяти

Пример

Во время запуска системы запускается процесс проверка наличия обновлений. Такой процесс может занять 5–10 Мбайт памяти. В этот же момент запускаются и остальные фоновые процессы проверяют наличие входящей почты, входящих сетевых подключений и многое другое.

Дефицит памяти

Все это до того, как будет запущена первая пользовательская программа.
А современные солидные пользовательские прикладные программы вроде Photoshop могут требовать для запуска 500 Мбайт памяти, а при начале обработки данных занимать множество гигабайт.

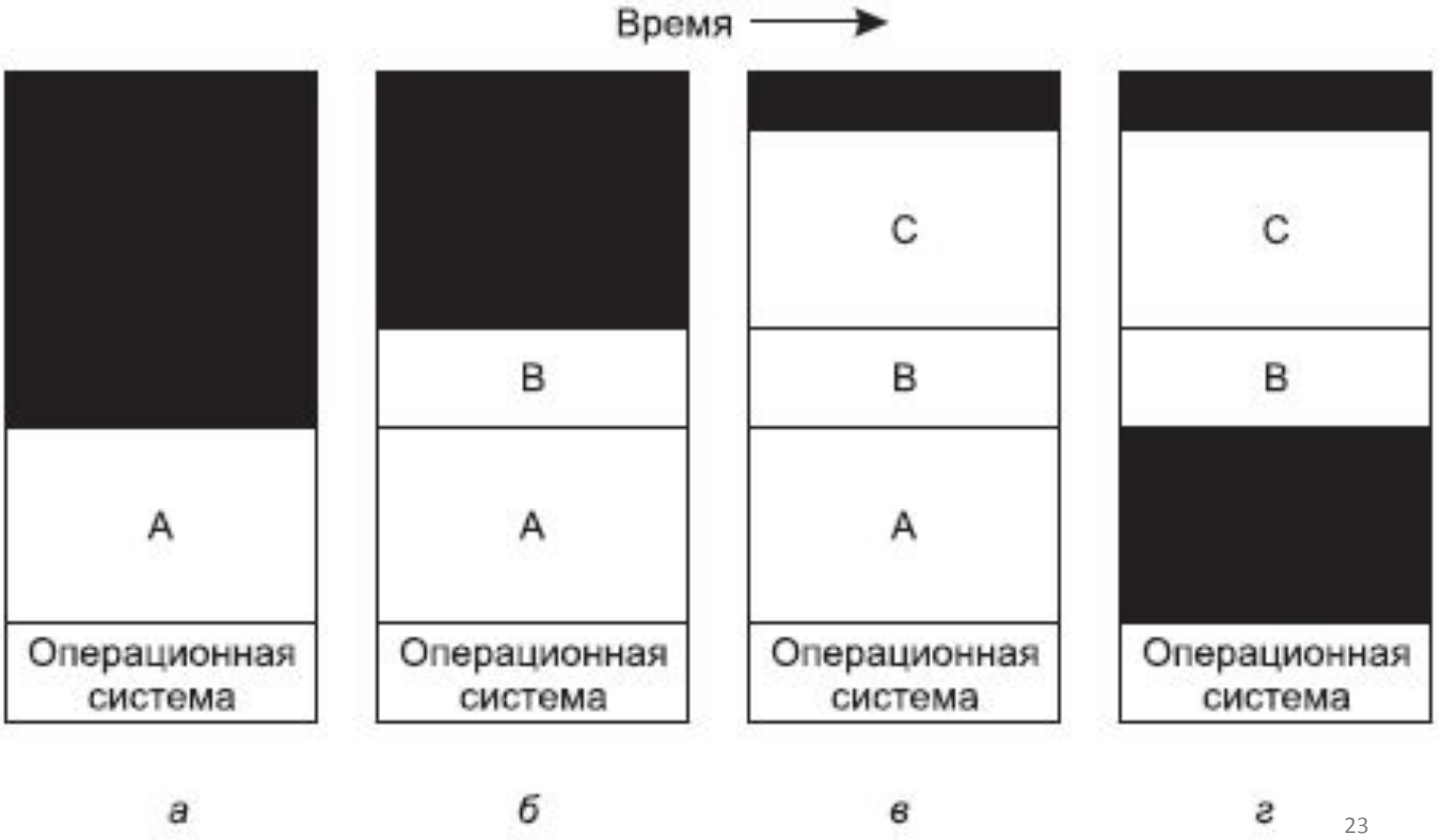
Дефицит памяти

Постоянное содержание всех процессов в памяти требует огромных объемов памяти и что приводит к дефициту памяти. Имеются две технологии преодоления дефицита памяти: *свопинг* и *виртуальная память*.

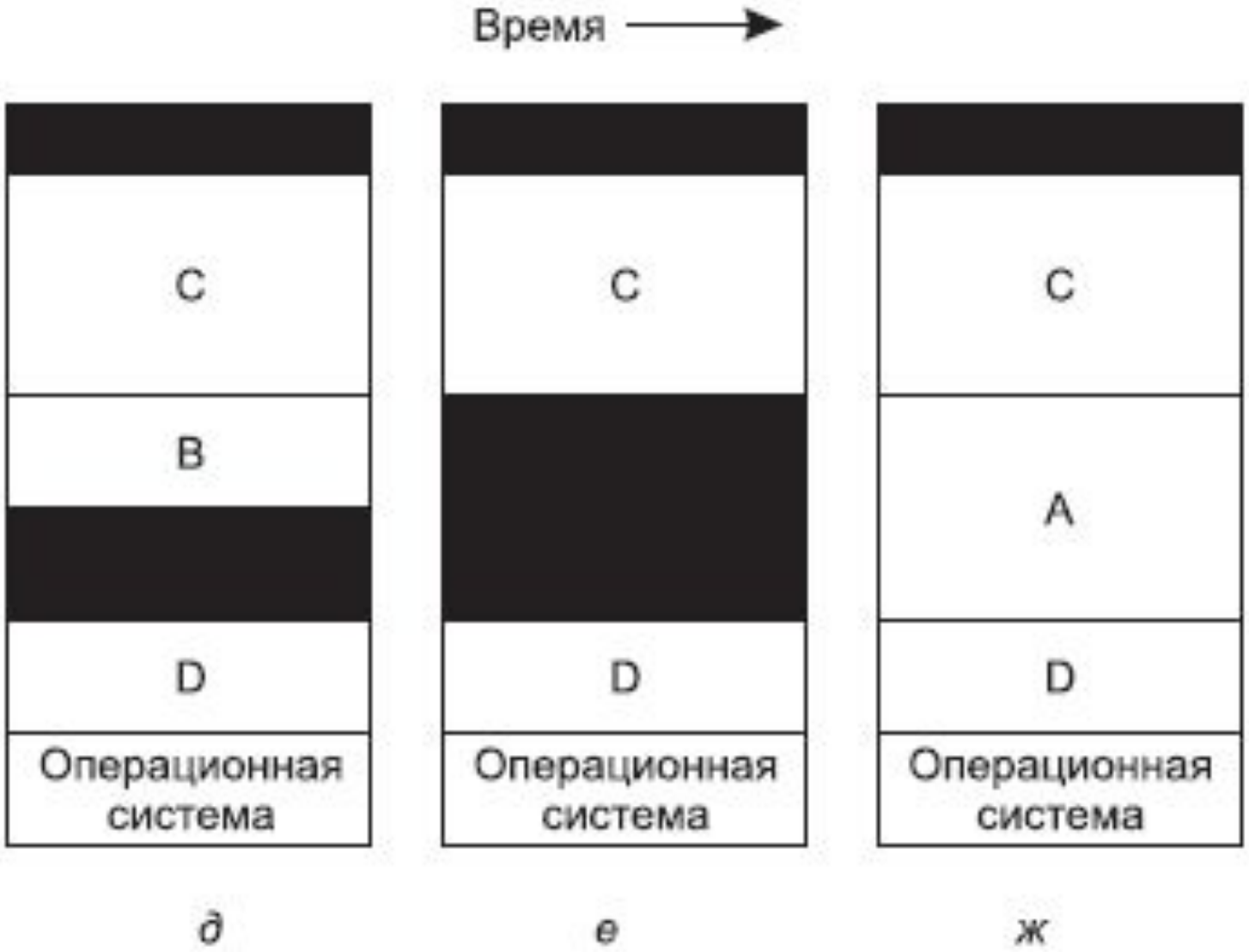
Свопинг

При *свопинге* неактивный процесс вместе с данными перемещается на диск, то есть в течение некоторого времени процесс может полностью отсутствовать в оперативной памяти. В случае его активации, этот процесс опять загружается в память и продолжает прерванное выполнение.

Свопинг


Время
A
Операционная система
B
A
Операционная система
C
B
A
Операционная система
C
B
Операционная система
а
б
в
г

Свопинг


Время
C
B
D
Операционная система
C
D
Операционная система
C
A
D
Операционная система
д
е
ж

Виртуальная память

Несмотря на быстрый рост объемов памяти, объемы, требующиеся программному обеспечению, растут намного быстрее. Тенденция к использованию мультимедиа предъявляет еще большие требования к объему памяти.

Виртуальная память

Последствия такого развития выразились в необходимости запуска программ, объем которых не позволяет им поместиться в памяти, при этом возникает потребность в системах, поддерживающих несколько одновременно запущенных программ, каждая из которых помещается в памяти, но все вместе они превышают имеющийся объем памяти.

Виртуальная память

Свопинг для данного случая не слишком привлекателен, так как обычный диск обладает пиковой скоростью передачи данных в несколько сотен мегабайт в секунду, а это означает, что свопинг программы объемом 1 Гбайт займет секунды, и еще столько же времени будет потрачено на загрузку другой программы в 1 Гбайт.

Виртуальная память

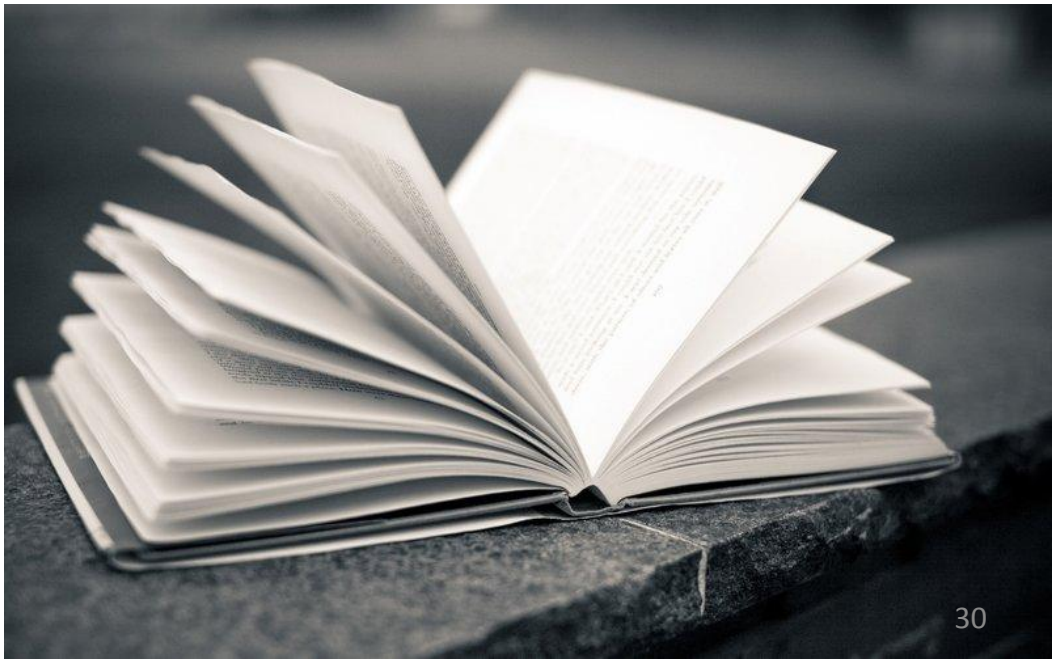
Кроме того, что для каждого процесса выделяется свое адресное пространство, ещё это пространство разбивается на отдельные фрагменты одинакового размера, которые называются **страницами**. Но это ещё не физическая память компьютера. Такая организация памяти называется ***виртуальной памятью***.

Виртуальная память

В основе технологии ***виртуальной памяти*** лежит идея, что у каждой программы имеется собственное адресное пространство, которое разбивается на участки, называемые ***страницами***. Каждая ***страница*** представляет собой непрерывный диапазон адресов. Эти страницы отображаются на физическую память, но для запуска программы одновременное присутствие в памяти всех страниц необязательно.

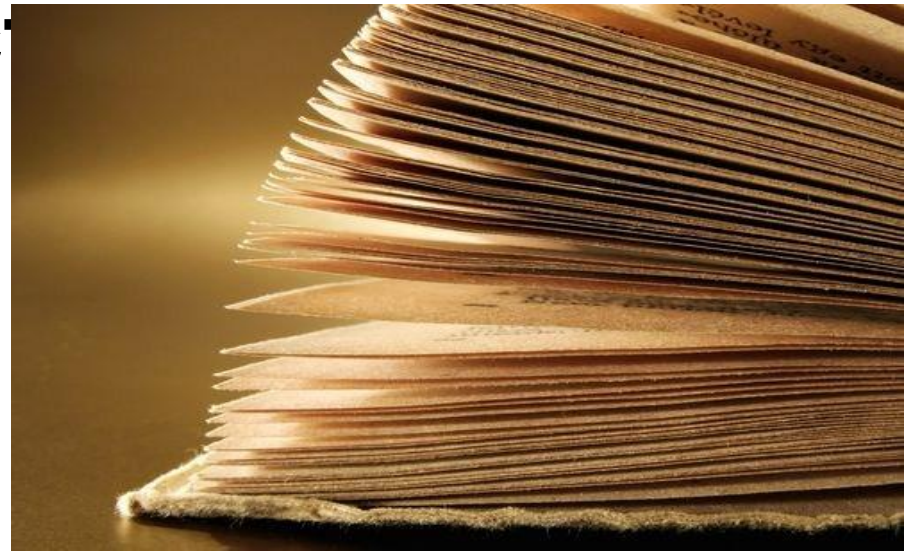
Страничное распределение:

При таком способе все фрагменты программы, на которые она разбивается (кроме последней части) получаются одинаковыми. Одинаковыми должны быть и единицы памяти, предоставляемые для размещения фрагментов программы.

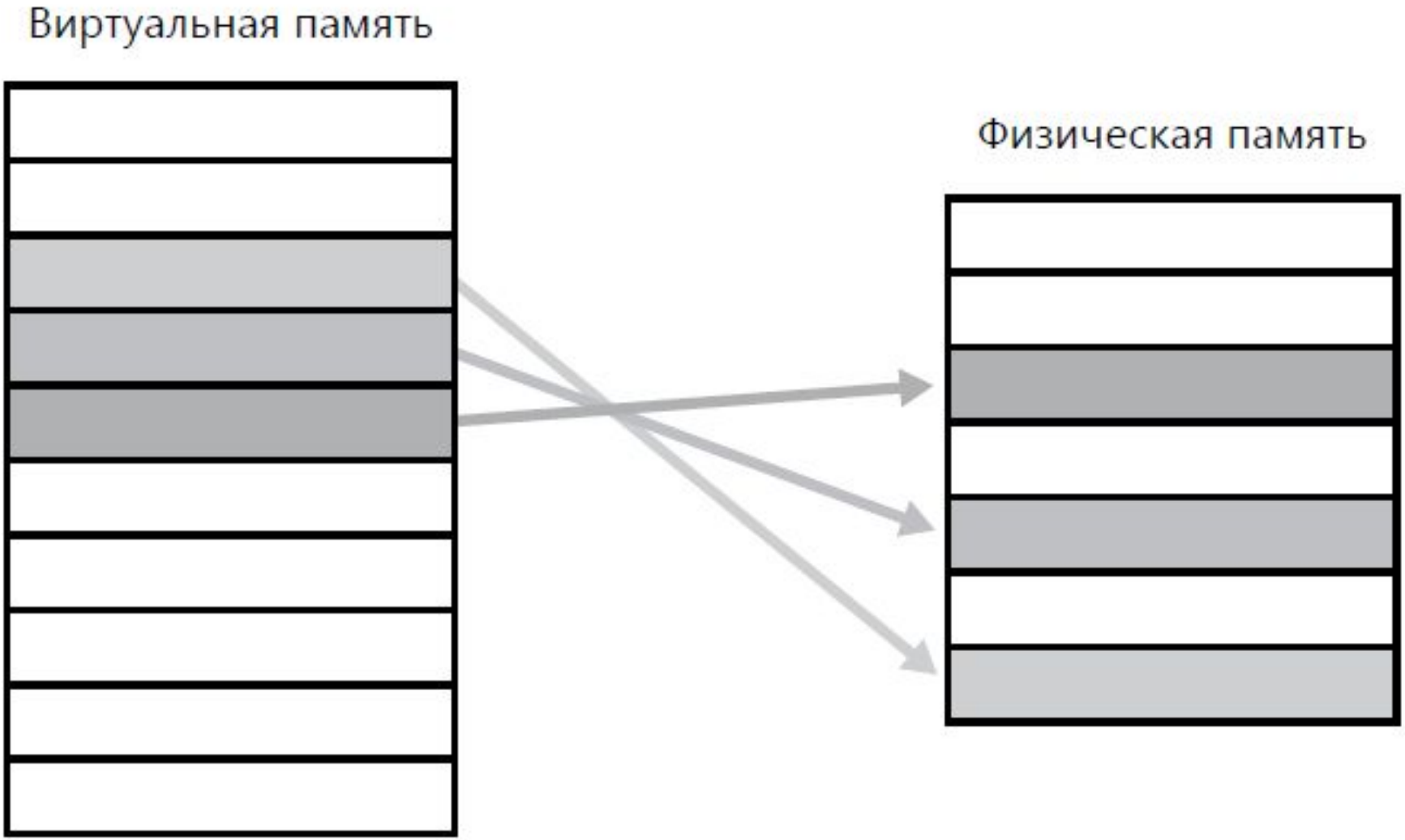


Эти одинаковые части называются страницами:

- оперативная память разбивается на физические страницы;
- программа разбивается на виртуальные ст



Управление памятью



Виртуальная память

Когда программа ссылается на часть своего адресного пространства, находящегося в физической памяти, то операция выполняется сразу.

Страничный способ управления памятью

Виртуальная память

Когда программа ссылается на часть своего адресного пространства, которое ***НЕ находится*** в физической памяти, операционная система предупреждается о том, что необходимо получить недостающую часть, загружает эту часть в ОП и повторно выполняет команду, которая не была выполнена.

Виртуальная память

Виртуальная память неплохо работает и в многозадачных системах, когда в памяти одновременно содержатся составные части многих программ. Пока программа ждет считывания какой-либо собственной части, время центрального процессора может быть отдан другому процессу.

Средство поддержки виртуального адресного пространства решает следующие задачи:

- размещает данные в запоминающих устройствах разного типа, например, часть программы в оперативной памяти, а часть на диске;
- перемещает по мере необходимости данные между запоминающими устройствами разного типа, например, подгружает нужную часть программы с диска в оперативную память;
- преобразует виртуальные адреса в физические.

Страничный способ организации виртуальной памяти – способ разрывного размещения задач в памяти, при котором все фрагменты задачи имеют одинаковый размер, кратный степени двойки (для использования операции конкатенации).

При таком способе все фрагменты программы, на которые она разбивается (кроме последней части) получаются одинаковыми. Одинаковыми должны быть и единицы памяти, предоставляемые для размещения фрагментов программы.

Эти одинаковые части называются страницами:

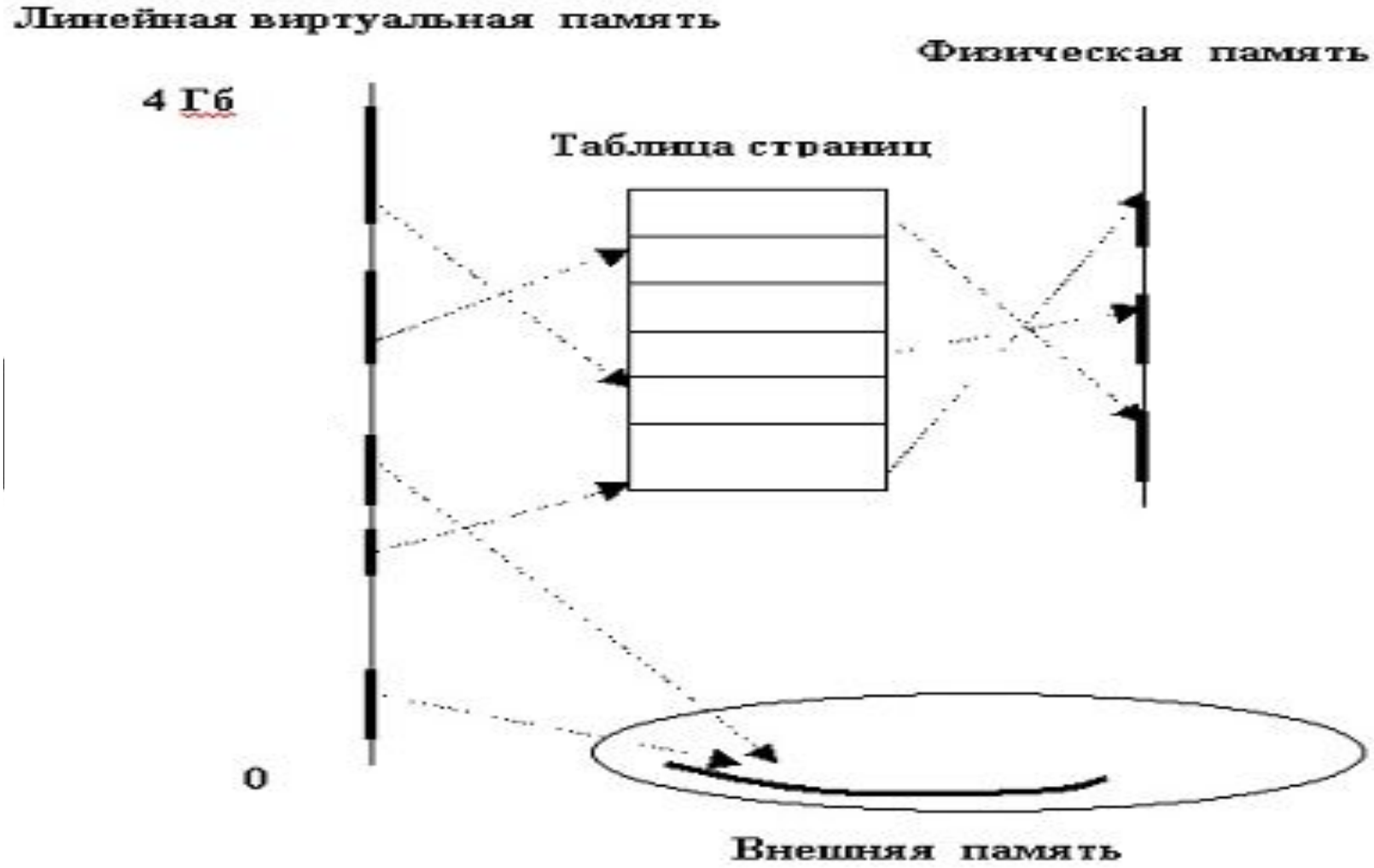
- оперативная память разбивается на физические страницы;
- программа разбивается на виртуальные страницы.

Страничная организация памяти

С точки зрения программиста:

- Процессам виртуальное адресное пространство представляется непрерывным, от байта 0 до байта N
- N зависит от аппаратной поддержки (например 32бит. - адр.пространство 4Гб), делится соответственно.
- В реальности виртуальные страницы распределены по страницам физической памяти далеко не непрерывно и не один к одному. Это два разных мира – физические страницы и виртуальные страницы. Это ключевой аспект, который

Управление памятью



Виртуальный адрес состоит из двух полей:

- указатель на часть программы (с которой идет работы) для определения местоположения этой части;
- относительный адрес нужного фрагмента памяти (по отношению к найденному адресу).

- **Часть виртуальных страниц располагается в оперативной памяти, а часть – во внешней (файл подкачки, страничный файл, swap-файл).**

Трансляция адресов

Трансляция виртуального адреса:

Виртуальный адрес состоит из двух частей: номер виртуальной страницы (VPN) и смещение внутри страницы

Номер виртуальной страницы (**VPN**- virtual page number) это индекс в таблице страниц (Pagetable)

Запись в таблице страниц (**PTE** – page table entry) содержит номер фрейма (**PFN** –page frame number)

Фрейм – это страница физической памяти.

Номер фрейма – это номер физической страницы.

Смысл таблицы страниц – одна запись в таблице страниц (PTE) на одну страницу виртуального адресного пространства (VPN), отображает VPN на PFN. Какая **виртуальная страница** соответствует какому **фрейму физической** памяти.

Трансляция адресов

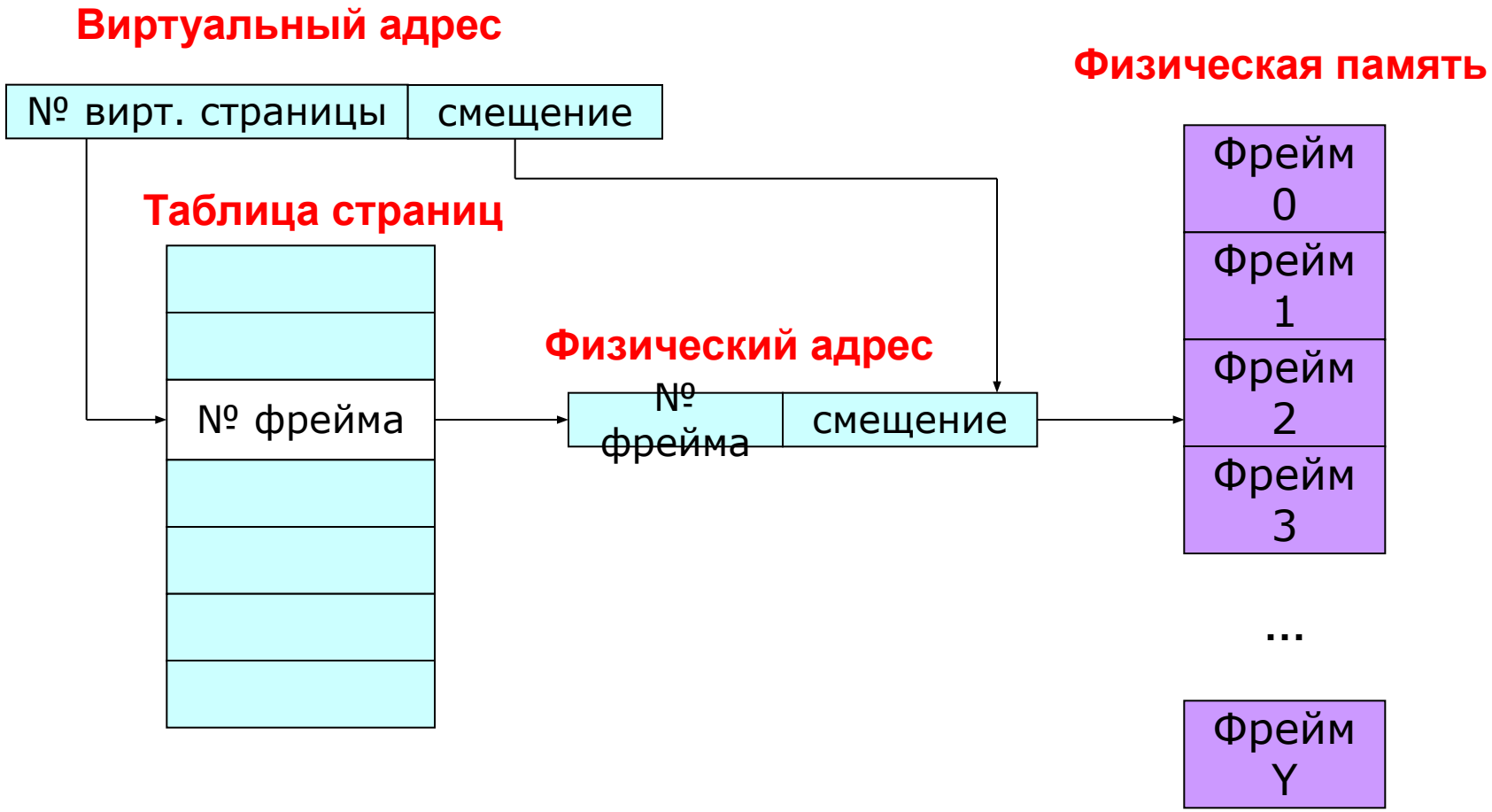


Таблица страниц

Запись в таблице – дескриптор страницы

- Номер физической страницы, в которую загружена данная виртуальная страница
- Признак присутствия = 1, если виртуальная страница находится в ОП
- Признак модификации страницы 1, когда производится запись по адресу, относящемуся к данной странице
- Признак обращения (бит доступа) 1, при каждом обращении по адресу, относящемуся к данной странице

- Признаки присутствия, модификации и обращения в большинстве моделей процессоров устанавливаются аппаратно при выполнении операций с памятью
- Сами таблицы страниц размещаются в оперативной памяти
- Адрес таблицы страниц включается в контекст соответствующего процесса
- При активизации очередного процесса ОС загружает адрес его таблицы страниц в специальный регистр процессора

Страничное прерывание

- Номер виртуальной страницы определяется нужный элемент таблицы страниц извлечение информации о станице (в том числе адрес в ОП)
- Анализ признака присутствия, если страница в ОП есть преобразование адреса (ВА – ФА), если страницы нет, то страничное прерывание (действия)
- Попытка загрузить страницу в ОП, возможно решение о выгрузке другой страницы на диск
- Для выгружаемой страницы обнуляется признак (бит) присутствия и проверяется признак модификации, если она была, то страница записывается на диск
- Иногда обнуление освобождаемой страницы

Для хранения информации о положении вытесненной страницы в страничном файле ОС может использовать поля таблицы страниц или другую системную структуру данных, например дескриптор процесса.

- Виртуальный адрес (P,Sv)
- Физический адрес (N, Sf)
- Задача подсистемы виртуальной памяти сосоит в отображении (P,Sv) на (N,Sf)
- Основные принципы:

  - объем страницы кратен степени 2

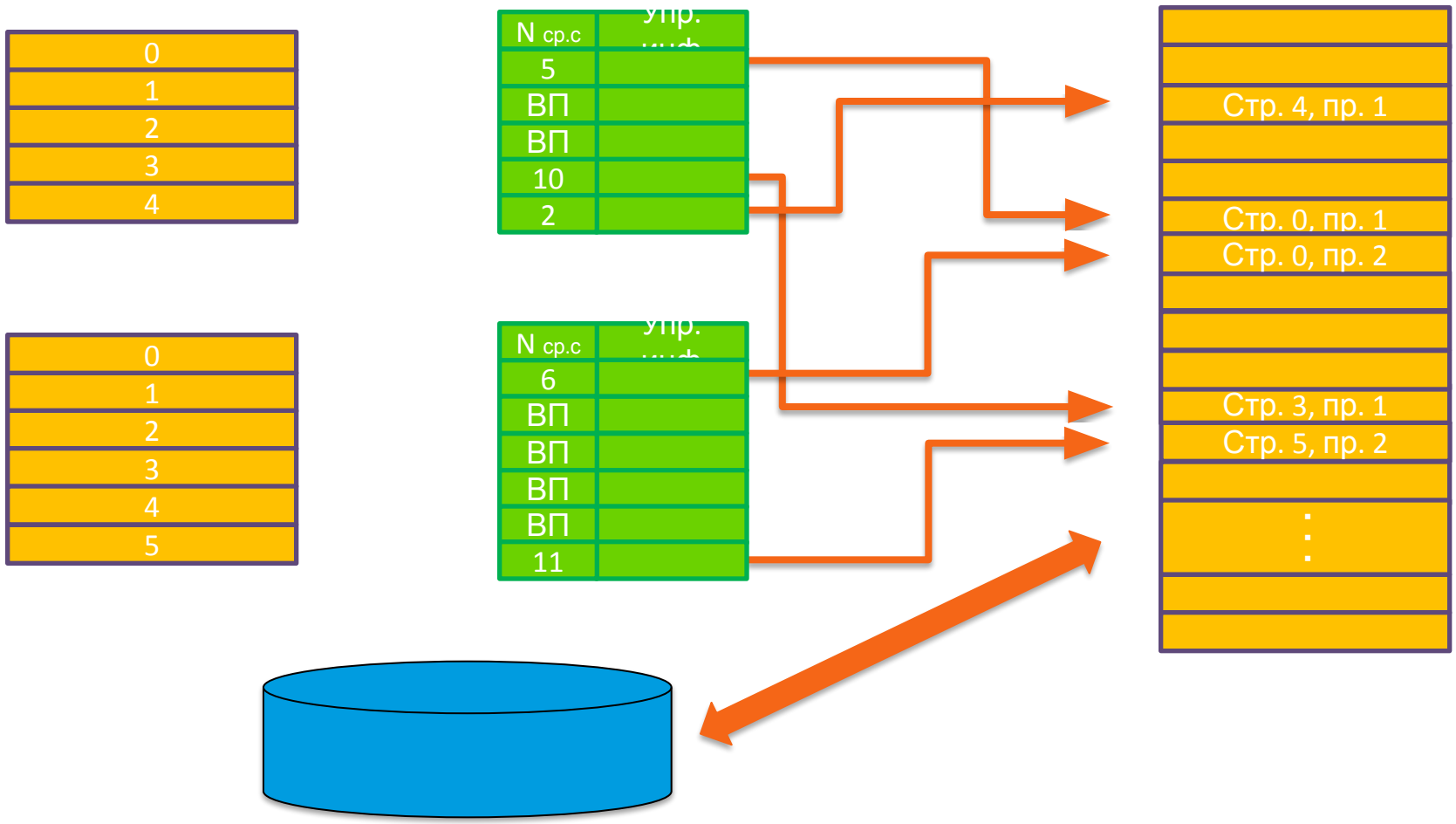
  - в пределах страницы непрерывная последовательность виртуальных адресов однозначно отображается в непрерывную последовательность физических адресов. Значит смещения в виртуальных и физических номерах равны (p11’).

S может быть получено простым отделением K младших разрядов в двоичной записи адреса, а оставшиеся старшие разряды адреса – двоичная запись номера страницы.

Предположим, что размер страницы 1 кб (2 в 10), тогда адрес 5071 восьмеричный = 101 000 111 001 двоичный. Тогда это – страница 10 и смещение 1 000 111 001.

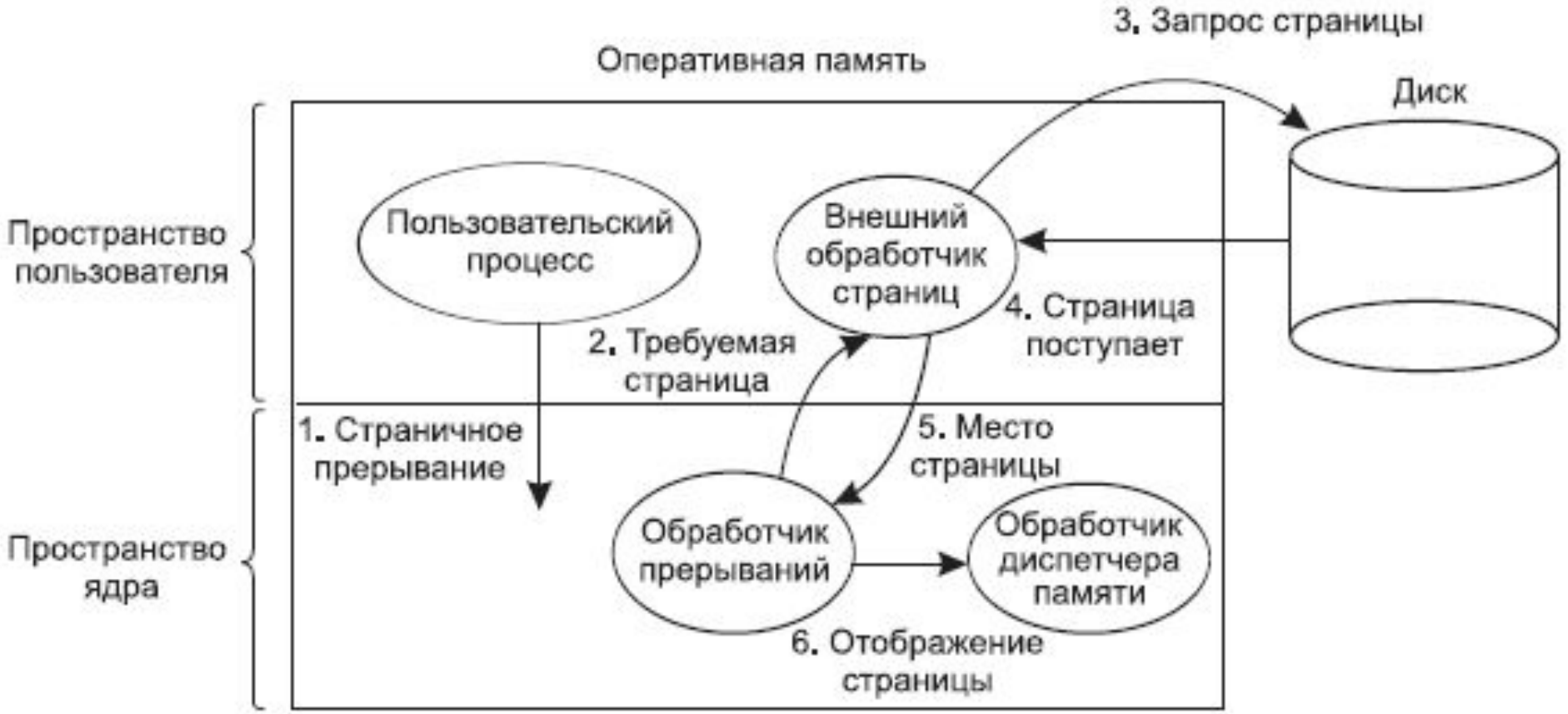
Т.о. таблица страниц содержит начальный физический адрес страницы в памяти.

- Пример: для адреса 5071 (восьмиричное) = 10 000 000 000 (двоичное) – адрес начала физической страницы. (Как бы страница 3). И смещение 1 000 111 001
- Схема преобразования виртуального адреса в физический при страничной организации (p12)

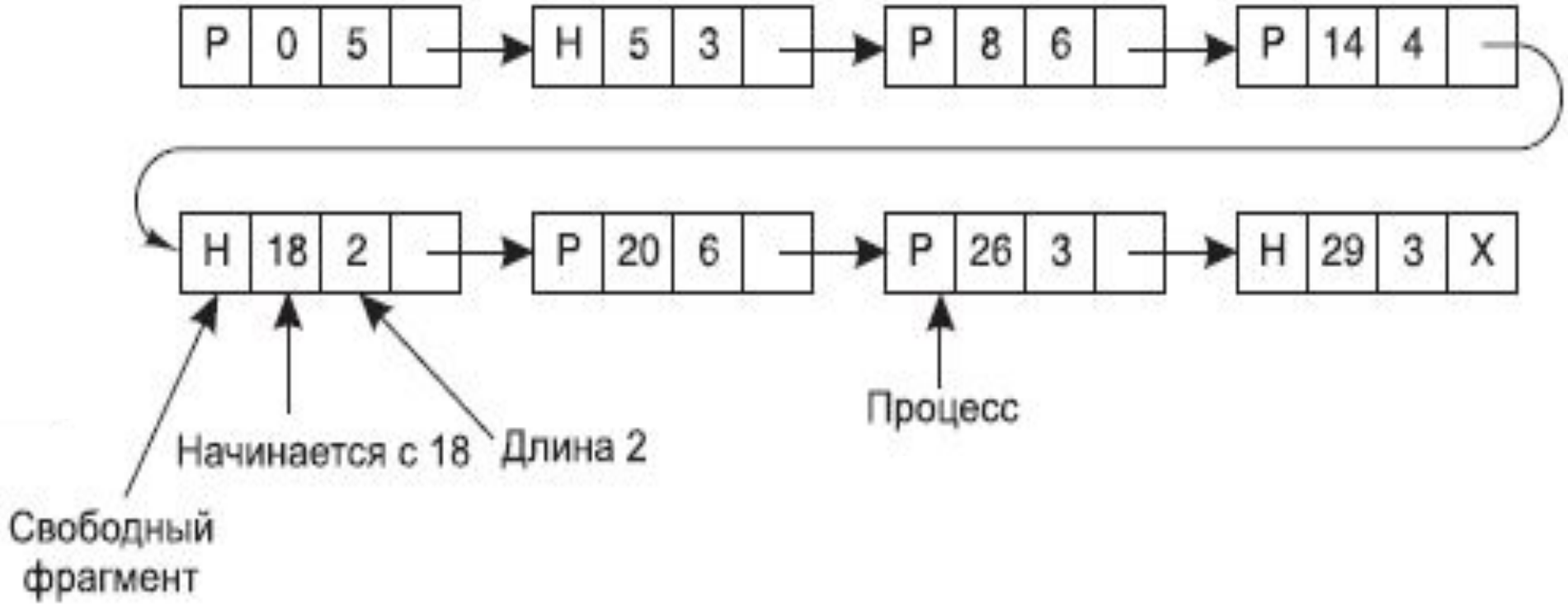

0
1
2
3
4
0
1
2
3
4
5
N ср.с
5
ВП
ВП
10
2
N ср.с
6
ВП
ВП
ВП
ВП
11
Стр. 4, пр. 1
Стр. 0, пр. 1
Стр. 0, пр. 2
Стр. 3, пр. 1
Стр. 5, пр. 2

Виртуальная память

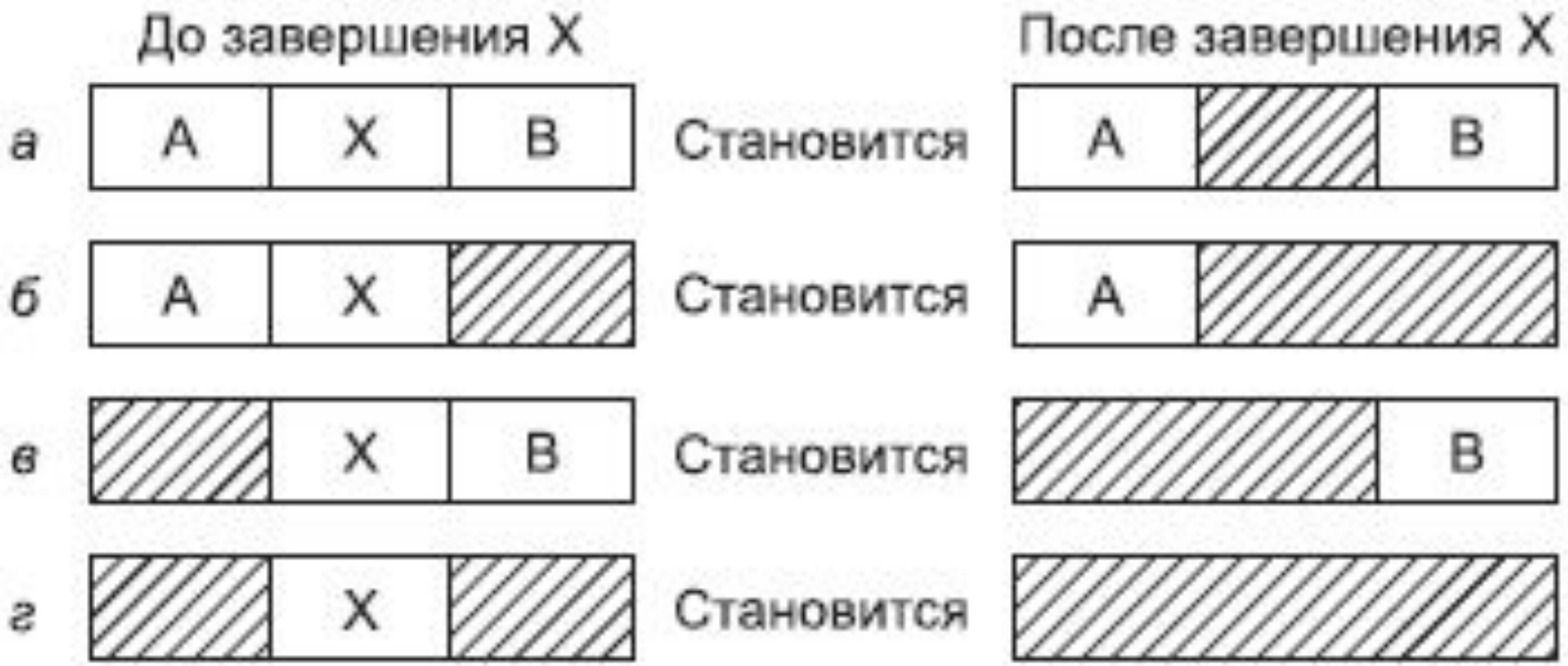
Виртуальная память



Управление свободной памятью



Управление свободной памятью



Задачи операционной системы :

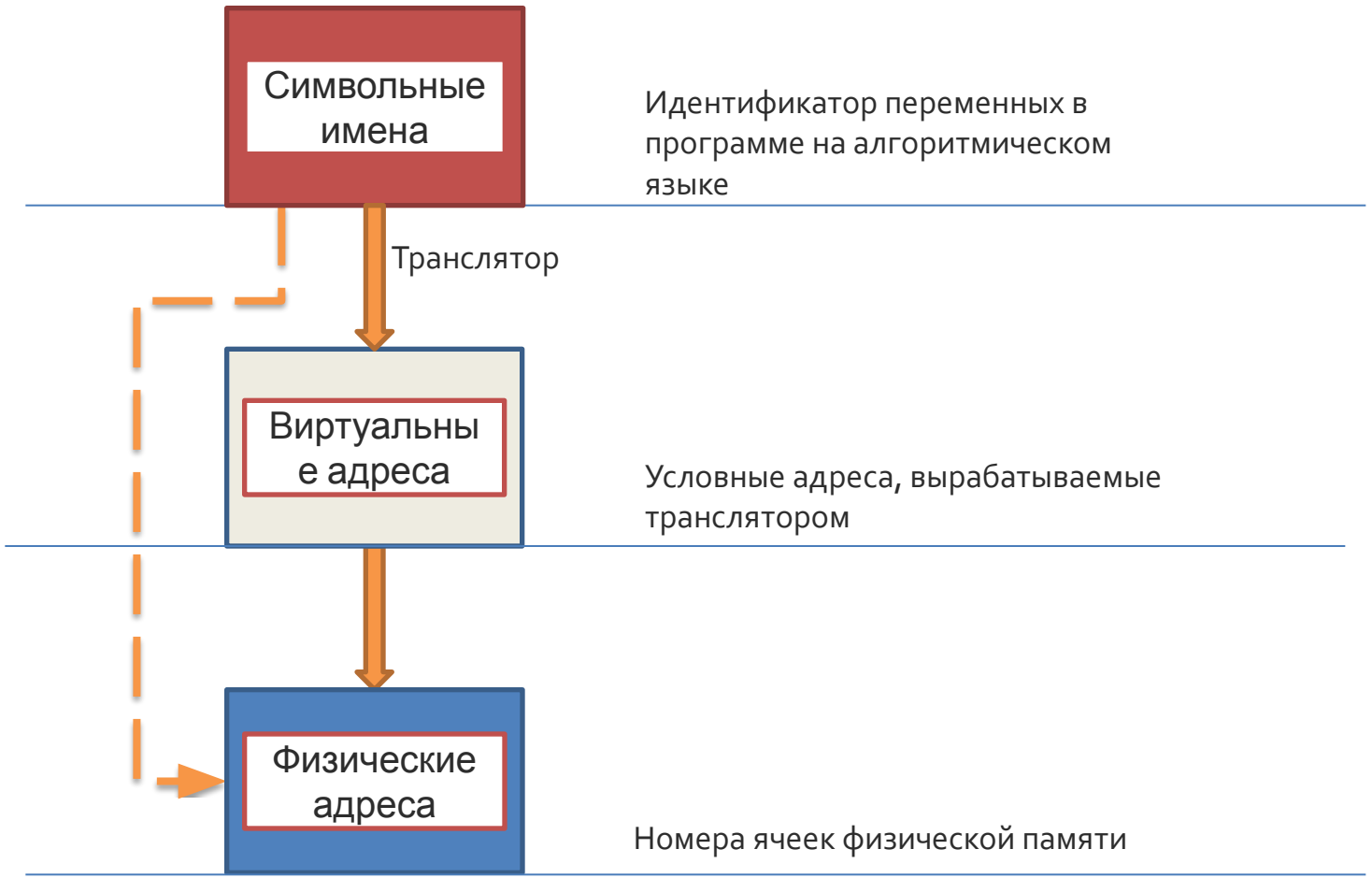
- ведение таблиц свободных и занятых областей, в которых указываются начальные адреса и размеры участков памяти;
- при поступлении новой задачи – анализ таблицы свободных областей и выбор раздела, размер которого достаточен для размещения поступившей задачи;
- загрузка задачи в выделенный ей раздел и корректировка таблиц свободных и занятых областей;
- после того, как задача завершится – освобождение памяти, которую занимала задача, т.е. выполнение корректировки таблиц свободных и занятых областей.

Программист обращается к памяти с помощью некоторого набора логических имен. Имена переменных и входных точек модулей составляют область имен.

Операционная система должна связать каждое указанное пользователем имя с физическим адресом памяти, т. е. осуществить отображение пространства имен на физическую память компьютера.

Это происходит в два этапа:

- посредством системы программирования;
- посредством операционной системы (с помощью специальных программных модулей управления памятью и использования соответствующих аппаратных средств вычислительной системы).


Символьные имена
Идентификатор переменных в программе на алгоритмическом языке
Транслятор
Виртуальные адреса
Условные адреса, вырабатываемые транслятором
Физические адреса
Номера ячеек физической памяти


Символьные имена
Идентификаторы переменных в программе на алгоритмическом языке
транслятор
Виртуальные адреса
Условные адреса, вырабатываемые транслятором
1) перемещающий загрузчик – статическое преобразование
2) динамическое преобразование аппаратными средствами
Физические адреса
Номера ячеек физической памяти

Между этими этапами обращение к памяти имеет форму виртуального.

Множество всех допустимых значений виртуального адреса для некоторой программы определяет ее виртуальное адресное пространство или виртуальную память.

Виртуальное адресное пространство зависит от:

- архитектуры процессора;
- системы программирования.

Виртуальное адресное пространство не зависит от:

- объема реальной физической памяти, установленной в компьютере.

Адреса команд и переменных в готовой машинной программе, подготовленной к выполнению системой программирования, как раз и являются виртуальными адресами.

СПАСИБО ЗА ВНИМАНИЕ!!!

Все методы управления памятью могут быть разделены на два класса:

- методы, которые используют перемещение процессов между оперативной памятью и диском,
- методы, которые не делают этого.

Методы распределения памяти без использования внешней памяти:

- Распределение памяти фиксированными разделами.
- Распределение памяти разделами переменной величины.
- Перемещаемые разделы.

Методы распределения памяти с использованием внешней памяти:

- Сегментное распределение;
- Страничное распределение;
- Сегментно-страничное распределение.

Методы распределения памяти

без использования внешней памяти:

- **Распределение памяти фиксированными разделами.**
- **Распределение памяти разделами переменной величины.**
- **Перемещаемые разделы.**

Самым простым способом управления оперативной памятью является разделение ее на несколько разделов фиксированной величины. Это может быть выполнено вручную оператором во время старта системы или во время ее генерации

Подсистема управления памятью в этом случае выполняет следующие задачи:

- сравнивая размер программы, поступившей на выполнение, и свободных разделов, выбирает подходящий раздел,
- осуществляет загрузку программы и настройку адресов.

Методы распределения памяти без использования внешней памяти:

- Распределение памяти фиксированными разделами.
- Распределение памяти разделами переменной величины.
- Перемещаемые разделы.

При распределение памяти разделами переменной величины

память машины не делится заранее на разделы. Сначала вся память свободна. Каждой вновь поступающей задаче выделяется необходимая ей память.

Если достаточный объем памяти отсутствует, то задача не принимается на выполнение и стоит в очереди.

После завершения задачи память освобождается, и на это место может быть загружена другая задача.

Таким образом, в произвольный момент времени оперативная память представляет собой случайную последовательность занятых и свободных участков (разделов) произвольного размера.

Методы распределения памяти без использования внешней памяти:

- Распределение памяти фиксированными разделами.
- Распределение памяти разделами переменной величины.
- Перемещаемые разделы.

Методы распределения памяти с использованием внешней памяти:

- Сегментное распределение;
- Страничное распределение;
- Сегментно-страничное распределение.

Виртуальный адрес можно представить состоящим из двух полей:

- указатель на часть программы (с которой идет работы) для определения местоположения этой части;
- относительный адрес нужной ячейки памяти (по отношению к найденному адресу).

Методы распределения памяти с использованием внешней памяти:

- **Сегментное распределение;**
- **Страничное распределение;**
- **Сегментно-страничное распределение.**

Для сегментного способа организации виртуальной памяти программу нужно разбить на части и уже каждой части выделить физическую память.

Каждый программный модуль или их совокупность могут быть восприняты как отдельные сегменты.

Каждый сегмент размещается в оперативной памяти как самостоятельная единица.

Логически обращение к элементам программы производится как указание имени сегмента и смещения относительно его

Физически имя (или порядковый номер) сегмента соответствует некоторому адресу, с которого этот сегмент начинается при его размещении в памяти, и смещение должно прибавляться к этому адресу.

Методы распределения памяти с использованием внешней памяти:

- Сегментное распределение;
- Страничное распределение;
- Сегментно-страничное распределение.

При таком способе все фрагменты программы, на которые она разбивается (кроме последней части) получаются одинаковыми. Одинаковыми должны быть и единицы памяти, предоставляемые для размещения фрагментов программы.

Эти одинаковые части называются страницами:

- оперативная память разбивается на физические страницы;
- программа разбивается на виртуальные страницы.

Методы распределения памяти с использованием внешней памяти:

- Сегментное распределение;
- Страничное распределение;
- Сегментно-страничное распределение.

При сегментно-страничном способе организации виртуальной памяти программа разбивается на логически законченные части – **сегменты**, виртуальный адрес содержит указание на номер соответствующего сегмента.