## 31. Понятие виртуального адресного пространства процесса, для х86 и х64 архитектур. Последовательность выделения физической памяти процессу. (Резервирование регионов и выделение памяти под регионы.)(лекция 6-7, слайд 43-48)

### Понятие виртуального адресного пространства процесса

Виртуальное адресное пространство процесса - это совокупность адресов, который может быть использован процессом для обращения к памяти + диапазон адресов для размещения библиотек используемых процессом.

### Для х86 (32 битная система)

Максимальный размер выделяемой виртуальной памяти = 4 Гб

* + 2/3 Гб - пространство процессов пользователя
  + 2/3 Гб - системное пространство

### Для х64 (64 битная система)

Максимальный размер выделяемой виртуальной памяти = 17 Тб(теоретически)

Максимальный размер выделяемой виртуальной памяти = 16 Тб(реально)

* + 8/7 Тб - пространство процессов пользователя
  + 8/7 Тб - системное пространство

### Последовательность выделения физической памяти процессу

#### 1 этап. Выделение региона

ОС выделяет автоматически регионы (для кода, данных, стека, кучи) в момент загрузки исполняемого файла. Остальные регионы процесс может выделить сам во время работы с помощью системных функций.

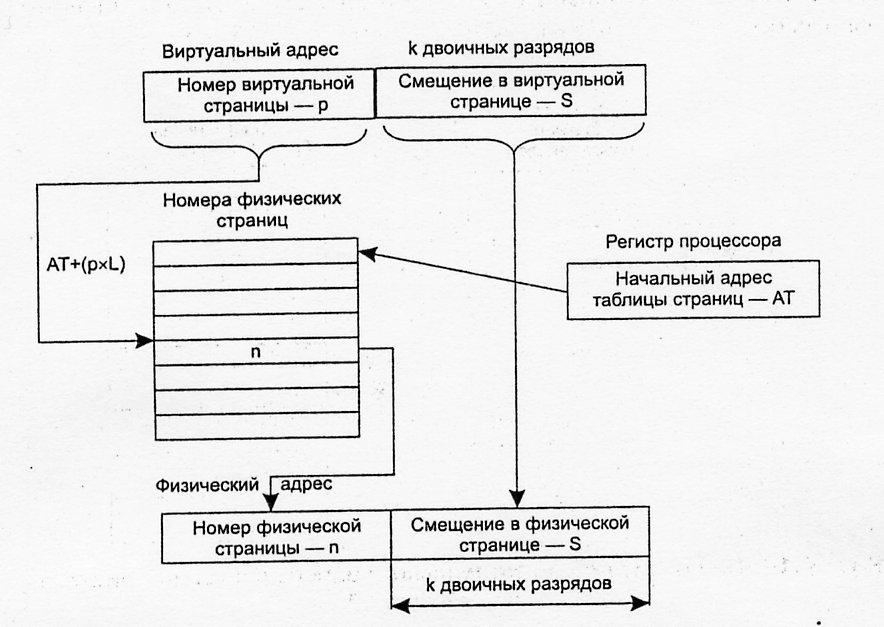
#### 2 этап. Выделение физической памяти региону

ОС выделяет регионам (или его частям) физическую оперативную память и загружает туда процесс. Если ОП не хватает, то ОС загружает в ОП часть процесса, а остальная часть находится на жестком диске в файле подкачки. По мере надобности страницы подкачиваются в ОП по прерыванию отсутствия страницы (paging).

## 32. Вычисление физического адреса памяти по виртуальному адресу при страничном разделении памяти (таблицы страниц.)

Произошло обращение к памяти по некоторому виртуальному адресу. *Аппаратными схемами процессора выполняются следующие действия:*

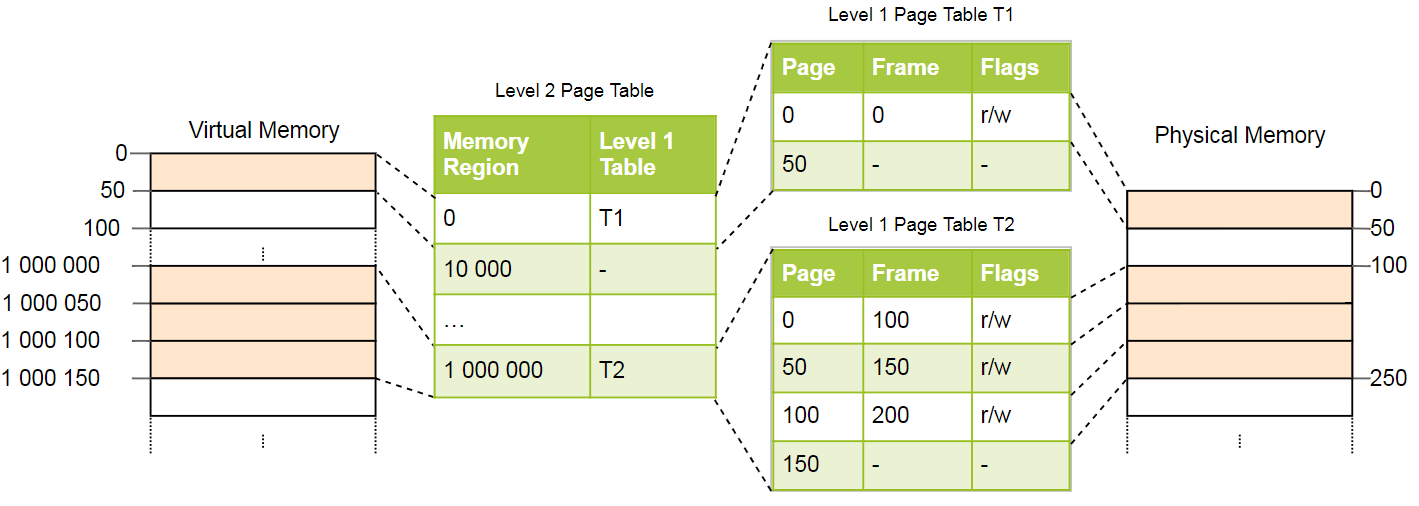
1. Из специального регистра процессора извлекается адрес AT таблицы страниц активного процесса. На основании начального адреса таблицы страниц, номера виртуальной страницы p(старшие разряды виртуального адреса) и длины отдельной записи в таблице страниц L(системная константа) определяется адрес нужного дескриптора в таблице страниц: a=AT+(p\*L).
2. Из этого дескриптора извлекается номер соответствующей физической страницы – n.
3. К номеру физической страницы присоединяется смещение s (младшие разряды виртуального адреса).



## 33. Методы увеличения скорости преобразования адреса при страничном разделении памяти (двухуровневые таблицы, ассоциативный буфер быстрого преобразования TLB)

### Двухуровневые таблицы

Дополнительная таблица, которая называется таблицей страниц *второго уровня*, выполняет преобразование между областями адресов и таблицами страниц первого уровня.

Пусть каждая таблица страниц уровня 1 отвечает за область размером 10\_000. 

Страница 0 попадает в первую область 10\_000 байт, поэтому использует первую запись таблицы страниц второго уровня. Эта запись указывает на таблицу страниц T1 первого уровня, которая определяет, что страница 0 ссылается на фрейм 0.

Страницы 1\_000\_000, 1\_000\_050 и 1\_000\_100 попадают в 100-ю байтовую область 10\_000, поэтому используют 100-ю запись таблицы страниц уровня 2. Эта запись указывает на другой таблицу T2 первого уровня, которая переводит три страницы во фреймы 100, 150 и 200. Обратите внимание, что адрес страницы в таблицах первого уровня не содержит смещения региона, поэтому, например, запись для страницы 1\_000\_050 составляет всего 50.

У нас по-прежнему 100 пустых записей в таблице второго уровня, но это гораздо меньше, чем прежний миллион. Причина экономии в том, что не нужно создавать таблицы страниц первого уровня для несопоставленных областей памяти между 10\_000 и 1\_000\_000.

### Ассоциативный буфер быстрого преобразования TLB

Большинство реально применяющихся схем виртуальной памяти используют специальный высокоскоростной кэш для записей таблицы страниц, который обычно называют буфером быстрой трансляции адресов - TLB.

Этот кэш функционирует так же, как и обычный кэш памяти, и содержит те записи таблицы страниц, которые использовались последними.

Получив виртуальный адрес, процессор просматривает TLB. Если требуемая запись найдена, процессор получает адрес физической страницы и формирует реальный адрес.

Если запись в TLB не найдена, то процессор использует номер виртуальной страницы в качестве индекса для таблицы страниц процесса и просматривает соответствующую запись.

Если бит присутствия в ней установлен, значит, искомая страница находится в основной памяти, и процессор получает номер физической страницы из записи таблицы страниц, а затем формирует реальный физический адрес.

Одновременно вносится использованная запись таблицы страниц в TLB.

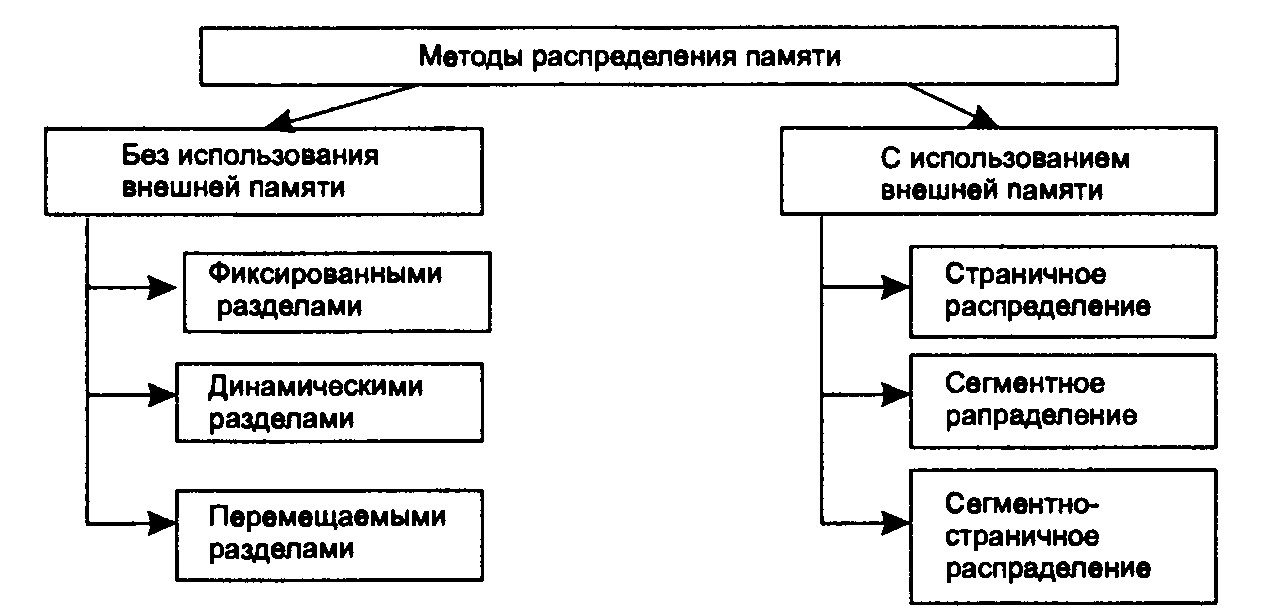
Если бит присутствия данной виртуальной страницы не установлен, это означает, что искомой страницы в оперативной памяти нет.

В этом случае процессор генерирует сигнал страничного прерывания, активизирующий операционную систему, которая загружает требуемую страницу в оперативную память и обновляет таблицу страниц.

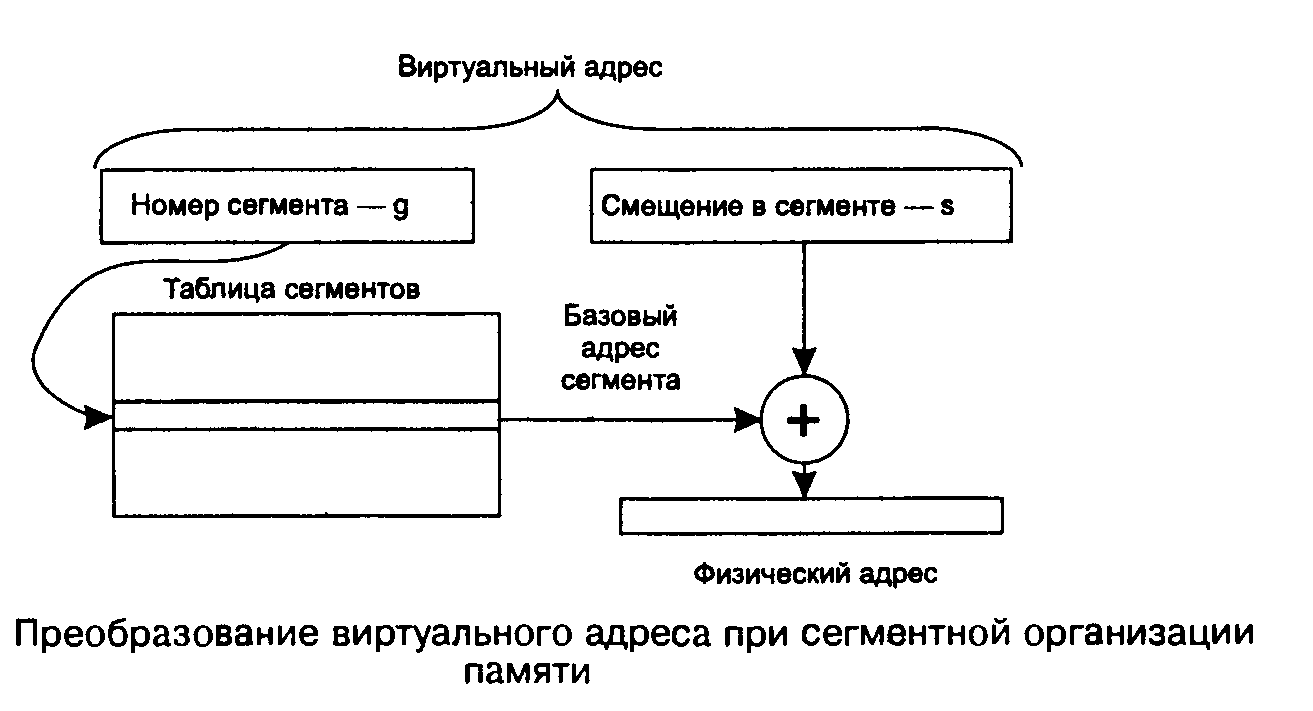
## 

## 34. Методы распределения памяти. Сегментное распределение виртуальной памяти.

### Методы распределения памяти



### Сегментное распределение виртуальной памяти



Достоинства:

* Гибкий дифференцированный доступ к различным частям процесса в зависимости от прав доступа (коду, данным, стеку)

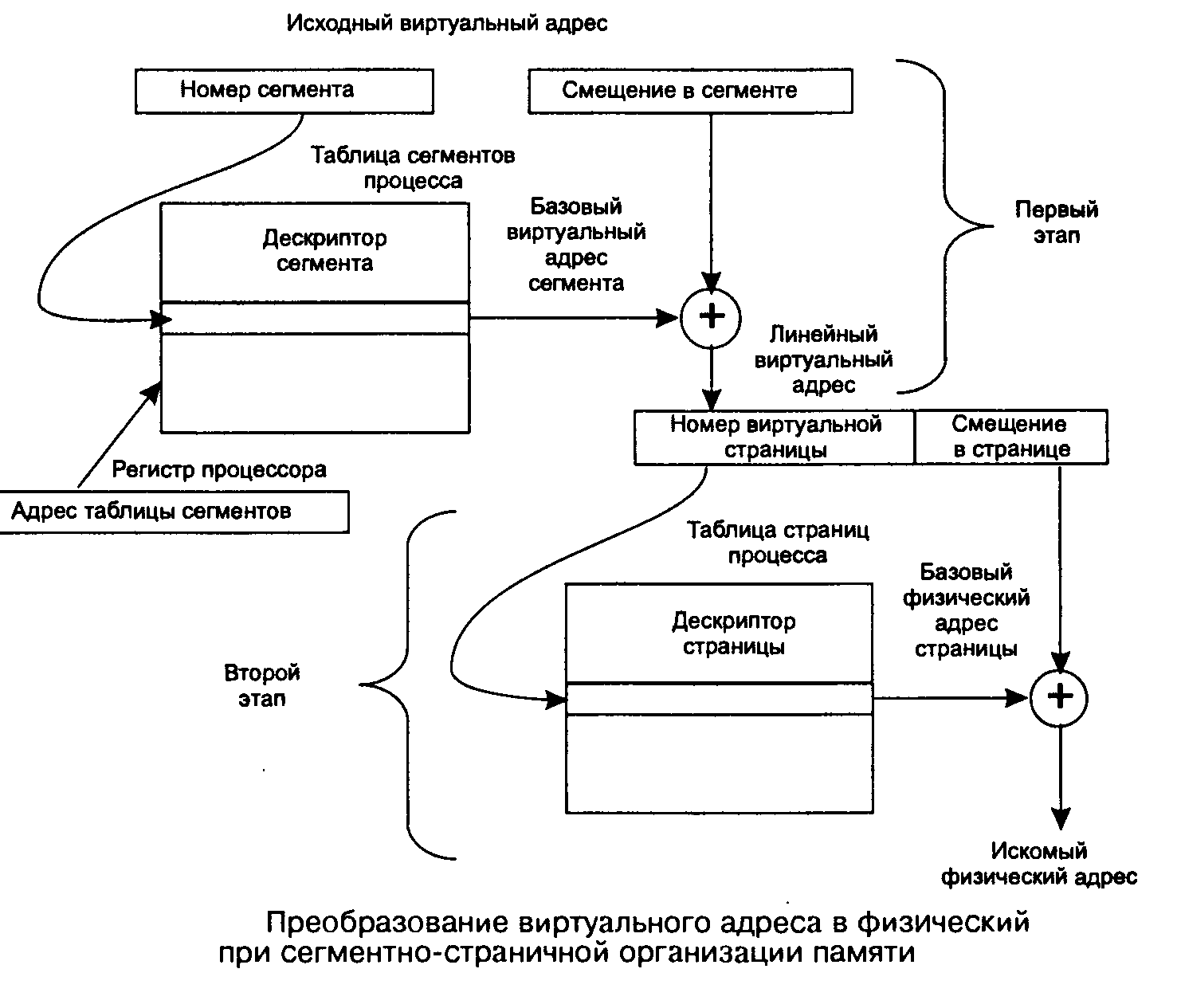
Недостатки:

* + Уменьшает быстродействие системы памяти
  + Повышенная фрагментация памяти

## 35. Методы распределения памяти. Сегментно - страничное распределение виртуальной памяти.

### Методы распределения памяти

### Сегментно - страничное распределение виртуальной памяти



* Сочетает достоинства страничной и сегментной организации.
* Сегмент разбивается на страницы. Hdd и ОП обмениваются между собой страницами.
* ОС организует две таблицы – сегментов и страниц содержащих дескрипторы (описатели) сегментов и страниц.
* Дескриптор сегмента содержит базовый адрес сегмента в виртуальном адресном пространстве. В результате его сложения со смещением получается линейный виртуальный адрес страницы , который на втором этапе (страничном) преобразуется в номер физической страницы с помощью таблицы страниц

## 36. Понятие файловой системы. Функции файловой системы. Имена, расширения, атрибуты, файлов. Основные системные вызовы для работы с файлами и каталогами.(лекция 8-9, слайд 2-6)

### Понятие файловой системы

Файловая система— это часть операционной системы, включаю­щая:

* + совокупность всех файлов на диске, организованных особым образом;
  + структуры данных, используемых для управления файлами (  
    каталоги файлов, дескрипторы файлов, таблицы распреде­ления свободного и занятого пространства и др.);
  + системные функции для работы с файлами (создание, уничтожение, чтение, запись, имено­вание , поиск файлов и др.)

Файловая система экранирует пользователя от физической реализации файловой системы.

Одна и та же операционная система может работать с различными файловыми системами.

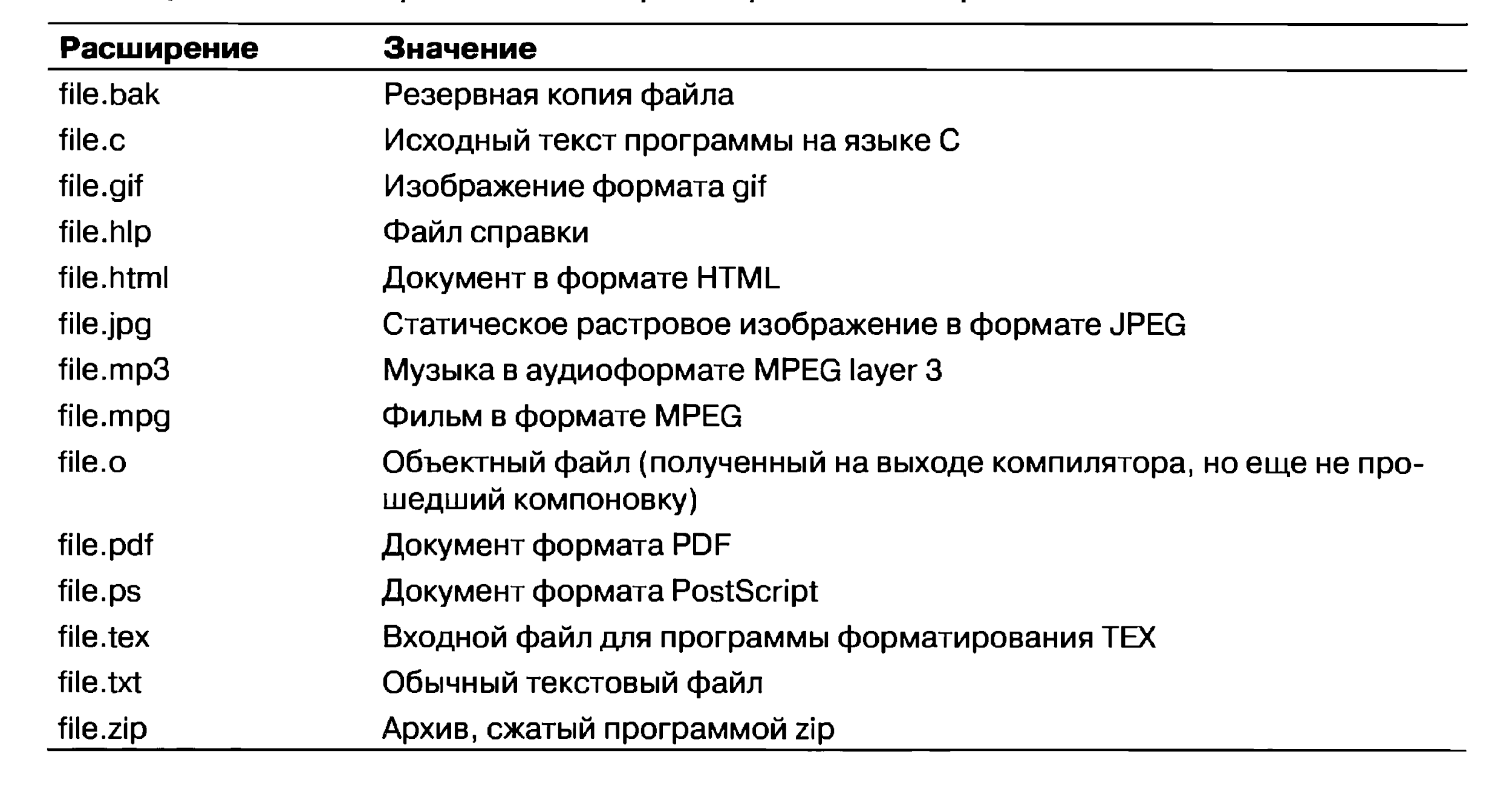
### Функции файловой системы

Нацелены на решение следующих задач:

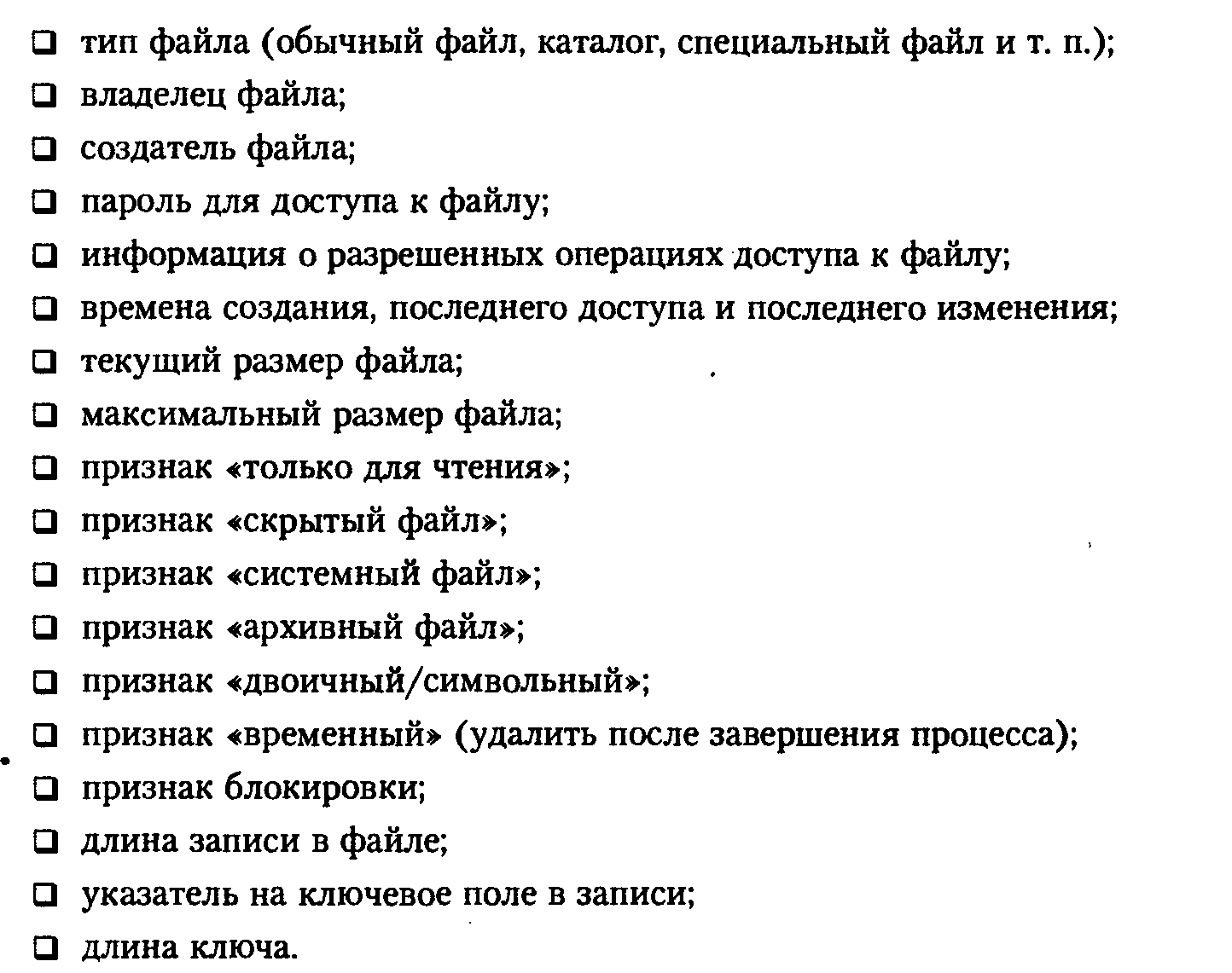
* именование файлов;
* программный интерфейс работы с файлами для приложений;
* отображения логической модели файловой системы на физическую организацию хранилища данных;
* организация устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;
* содержание параметров файла, необходимых для правильного его взаимодействия с другими объектами системы (ядро, приложения и пр.).

### Имена и расширения файлов

* Все поддерживают имена от одной до восьми букв.
* Многие файловые системы поддерживают имена длиной до 255 символов.

****

### Атрибуты файла (дополнительная информация о файле)

****