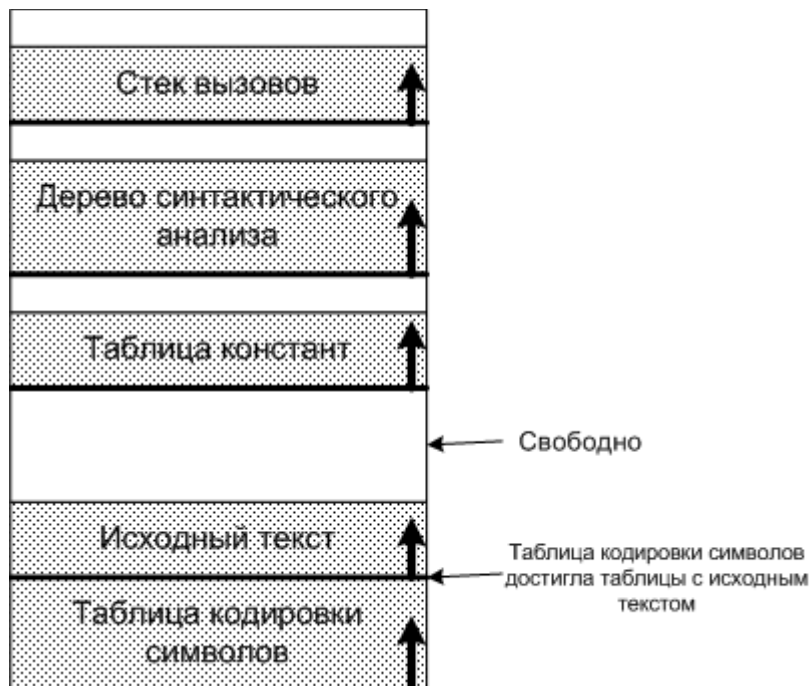


8.1 Основные понятия сегментации памяти

Рассмотрим пример, когда программа использует одно адресное пространство.



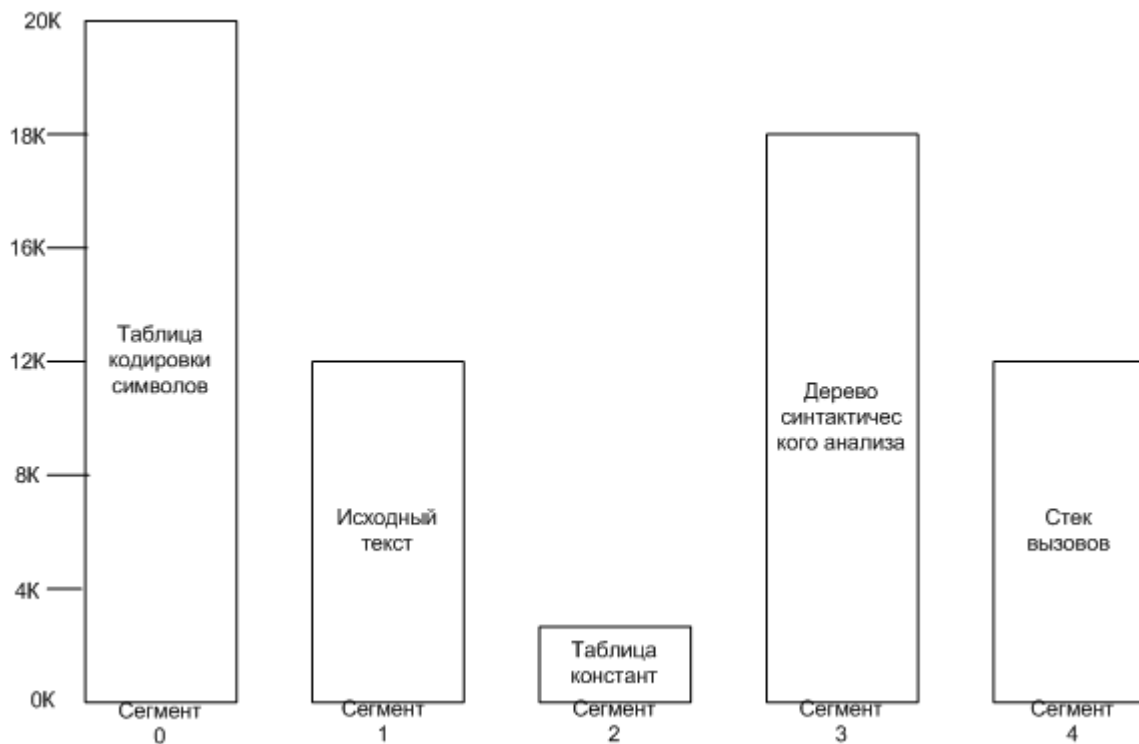
программа использует одно адресное пространство

Недостатки такой системы:

Один участок может полностью заполниться, но при этом останутся свободные участки. Можно конечно перемещать участки, но это очень сложно.

Эти проблемы можно решить, если дать каждому участку независимое адресное пространство, называемое **сегментом**.

Рассмотрим то же пример с использованием сегментов:



Сегментированная память

Каждый сегмент может расти или уменьшаться независимо от других.

Сегмент - это логический объект.

В этом случае адрес имеет две части:

- номер сегмента
- адрес в сегменте

Преимущества сегментации:

- Сегменты не мешают друг другу.
- Начальный адрес процедуры всегда начинается с $(n,0)$. Что упрощает программирование.
- Облегчает совместное использование процедур и данных.
- Раздельная защита каждого сегмента (чтение, запись).

8.2 Реализация сегментации

Если страницы имеют фиксированный размер, то сегменты нет.

У сегментов так же, как и у страниц, существует проблема фрагментации.

Т.к. памяти часто не хватает, стали использовать страничную организацию сегментов. При которой в памяти может находиться только часть сегмента.

8.2.1 Сегментация с использованием страниц: MULTICS

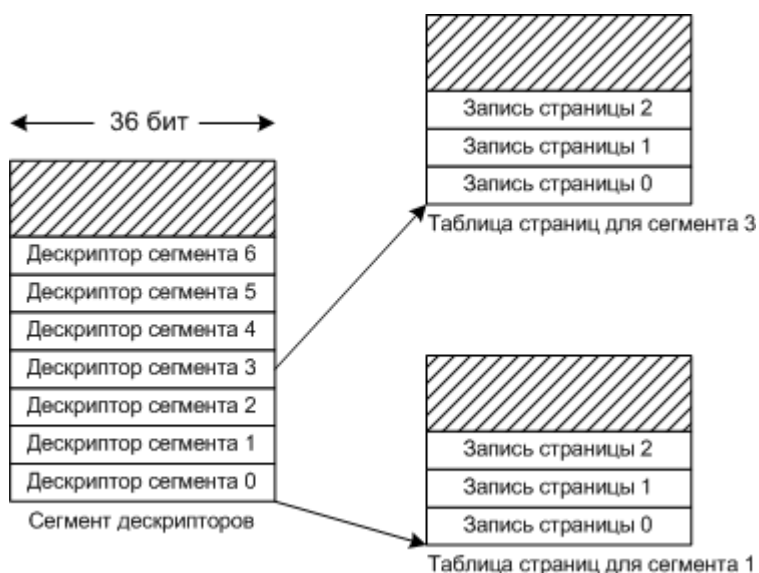
В одной из первых, где была применена страничная сегментация, была система MULTICS.

Каждая программа обеспечивалась до 2^{18} сегментов (более 250 000), каждый из которых мог быть до 65 536 (36-разрядных) слов длиной.

Таблица сегментов - хранит дескриптор для каждого сегмента. У каждой программы своя таблица.

Т.к. записей в таблице более 250 000, она сама разбита на страницы.

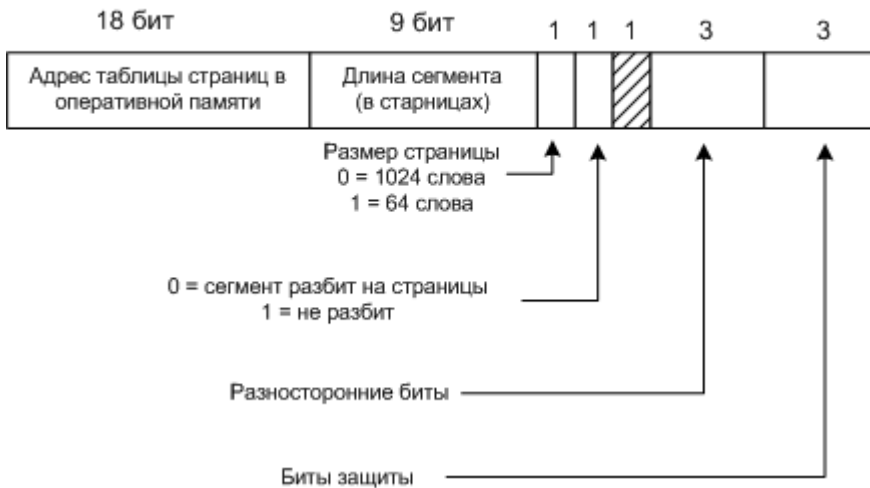
Сама таблица является отдельным сегментом.



Сегмент с таблицей дескрипторов указывающих на таблицы страниц для каждого сегмента

Нормальный размер страницы равен 1024 словам. Если сегмент меньше 1024, то он либо не разбит на страницы, либо разбит на страницы по 64 слова.

36 бит

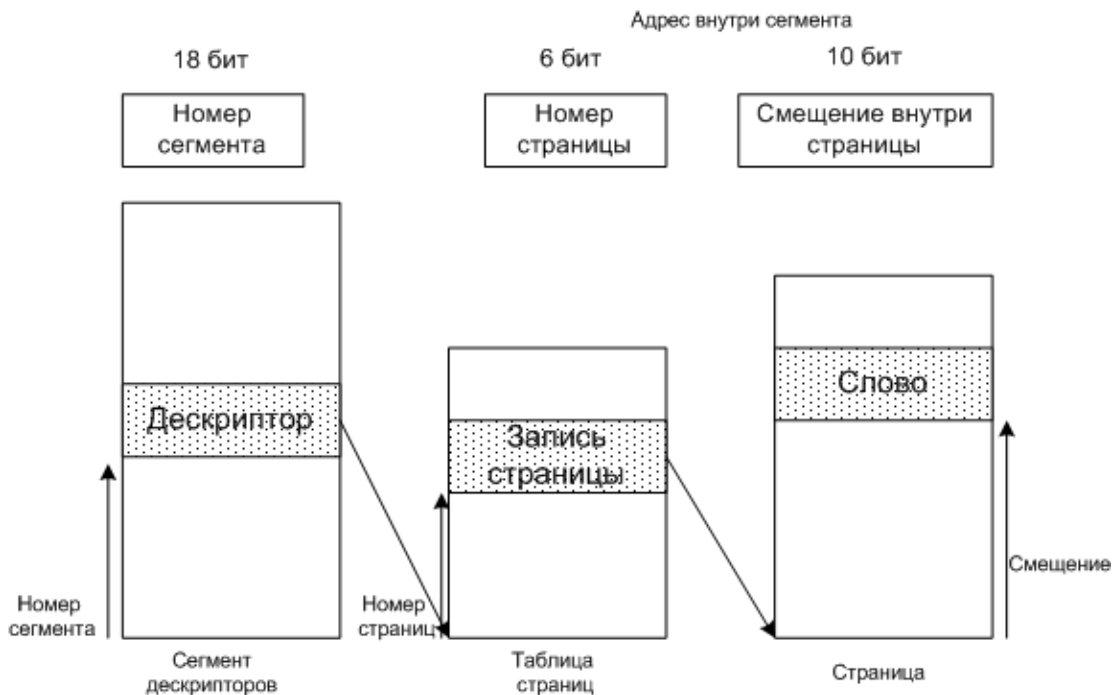


Дескриптор сегмента

Когда происходит обращение к памяти, выполняется следующий алгоритм:

1. По номеру сегмента находится дескриптор сегмента.
2. Проверяется, находится ли таблица страницы в памяти. Если в памяти, определяется ее расположение. Если нет, вызывается сегментное прерывание.
3. Проверяется, находится ли страница в памяти. Если в памяти, определяется ее расположение в памяти. Если нет в памяти, вызывается страничное прерывание.
4. К адресу начала страницы прибавляется смещение, в результате получаем адрес нужного слова в оперативной памяти.
5. Происходит запись или чтение.

Виртуальный адрес MULTICS



Преобразование адреса в системе MULTICS

Так как такой алгоритм будет работать достаточно медленно. Аппаратура системы MULTICS содержит высокоскоростной буфер быстрого преобразования адреса (**TLB**) размером в 16 слов. Адреса 16 наиболее часто используемых страниц хранятся в буфере.

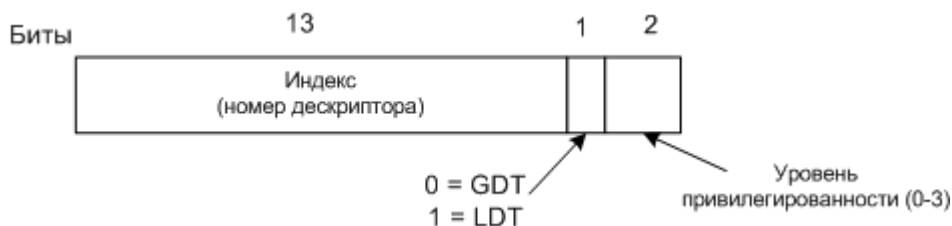
8.2.2 Сегментация с использованием страниц: Intel Pentium

Каждая программа обеспечивается до 16К сегментов, каждый из которых может быть до 1 млрд 36-разрядных слов длиной.

Основа виртуальной памяти системы Pentium состоит из двух таблиц:

- Локальная таблица дескрипторов LDT (Local Descriptor Table) - есть у каждой программы, и описывает сегменты программы.
- Глобальная таблица дескрипторов GDT (Global Descriptor Table) - одна для всех программ, и описывает системные сегменты (включая саму ОС).

Каждый селектор (указывает на дескриптор) представляет собой 16-разрядный номер.



Селектор в системе Pentium

13 битов определяют номер записи в таблице дескрипторов, поэтому эти таблицы ограничены, каждая содержит 8К (2^{13}) сегментных дескрипторов.

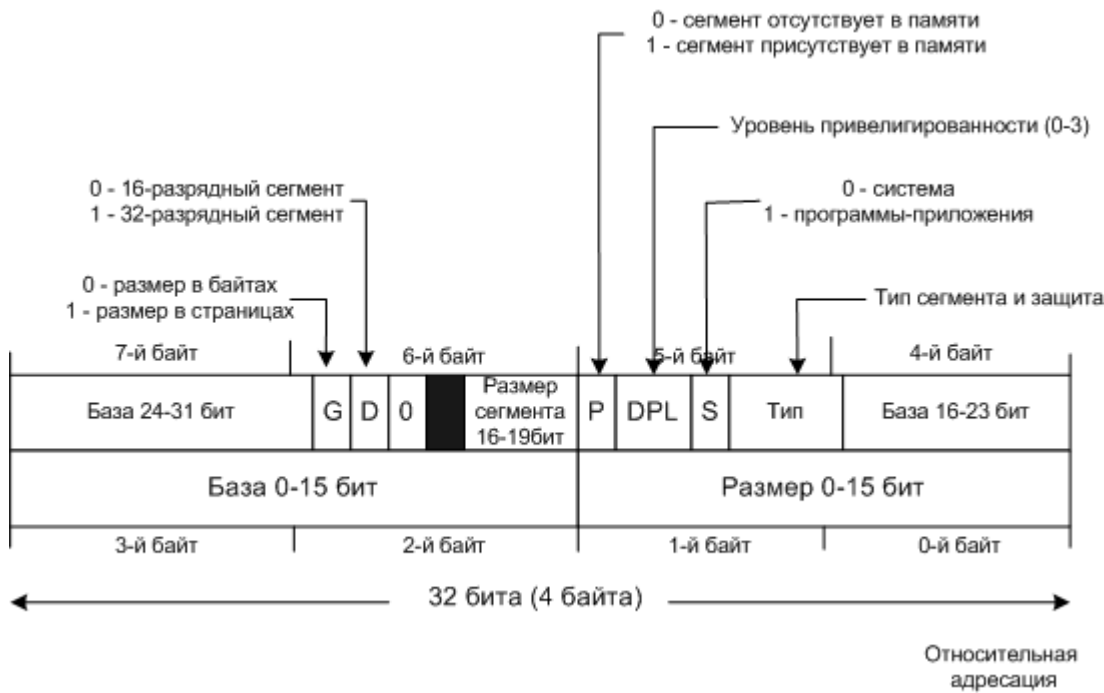
1 бит указывает тип используемой таблицы дескрипторов LDT или GDT.

Уровни	
Ядро	0
Системные узлы	1
Библиотеки совместного доступа	2
Пользовательские программы	3

Уровни привилегированности в системе Pentium

Уровни привилегированности запрещают выполняемому коду обратиться к более низкому уровню.

С учетом максимального размера сегмента - 4 Гбайта - каждая задача, при чисто сегментной организации виртуальной памяти, работает в виртуальном адресном пространстве в 64 Тбайта (4 Гбайта * 16К, где $16К=8К*2$ т.к. LDT и GDT).



Дескриптор программного (не данных) сегмента в системе Pentium (всего 8 байт (64 бита)).

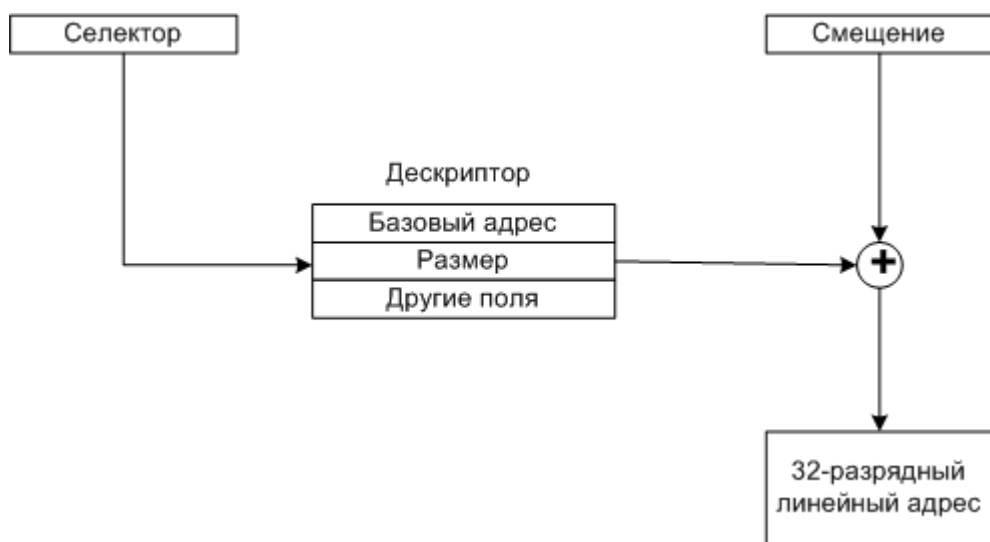
База (Base) - базовый адрес сегмента (32-бита), разделен на три части из-за совместимости с i286, в котором это поле имеет только 24 бита.

Размер (Limit) - размер сегмента (20 бит), разнесен на две части.

Если размер сегмента указан в страницах, он может достигать 2^{32} байтов ($2^{20} * 4\text{Кбайт}$ (2^{12}) (размер страницы в Pentium)).

Алгоритм получение физического адреса:

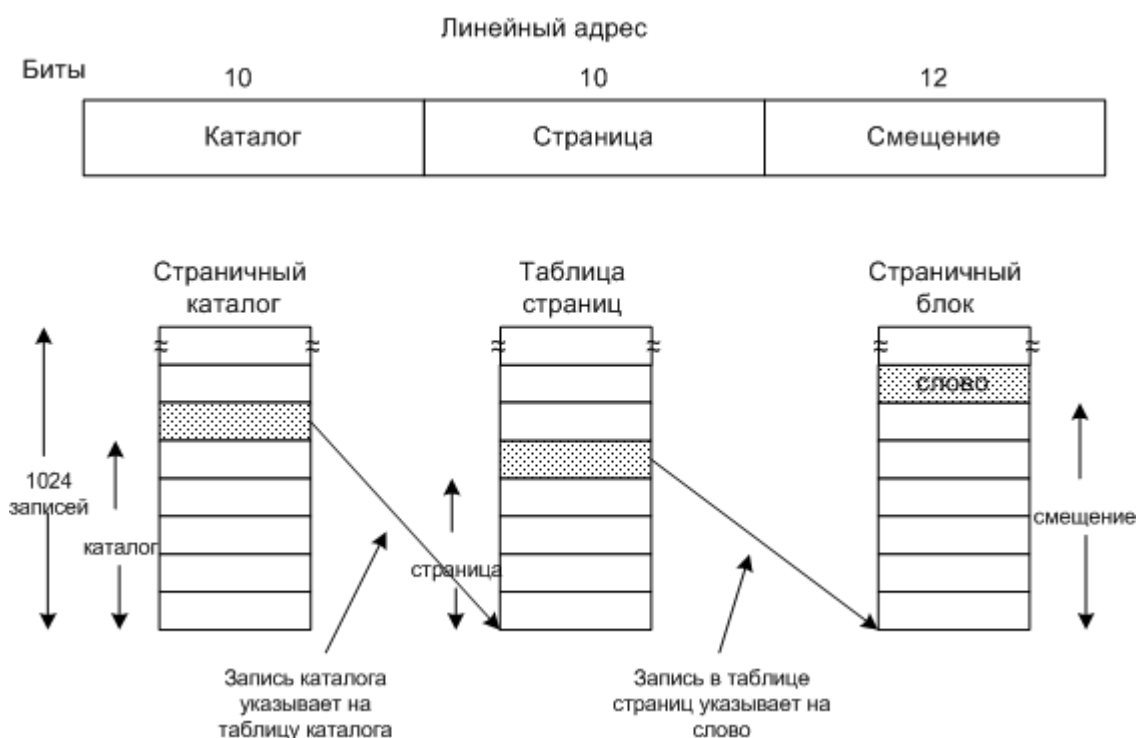
1. Селектор загружается в регистр (для сегмента команд в CS, для сегмента данных в DS).
2. Определяется глобальный или локальный сегмент (LDT или GDT).
3. Дескриптор извлекается из LDT или GDT, и сохраняется в микропрограммных регистрах.
4. Если дескриптор в памяти и смещение не выходит за пределы сегмента, программа может продолжить работу, если нет, происходит прерывание.
5. Система Pentium прибавляет базовый адрес к смещению, и получает **линейный адрес**,
- если страничная организация памяти не используется, то он является физическим адресом (адрес получен),
- если страничная организация памяти используется, то он является виртуальным адресом.
6. В случае, если используется страничная организация памяти, линейный адрес переводится в физический с помощью таблицы страниц.



Преобразование пары (селектора, смещение) в физический адрес

При 32-разрядном ($2^{32}=4\text{Гбайт}$) адресе и 4Кбайтной странице, сегмент может содержать 1 млн страниц (4Гбайт/4Кбайта). Поэтому используется двухуровневое отображение (создана таблица (**страничный каталог**) содержащая список из 1024 таблиц страниц), благодаря чему можно снизить количество записей в таблице страниц до 1024.

В этом случае сегмент в 4 Мбайта (1024 записи по 4 Кбайта страницы), будет иметь страничный каталог только с одной записью (и 1024 в таблице страниц), вместо 1 млн в одной таблице.



Отображение линейного адреса на физический адрес

В системе Pentium также есть буфер быстрого преобразования адреса (**TLB**), в котором хранятся наиболее часто используемые комбинации Каталог-Страница на физический адрес страничного блока. Только если комбинация в TLB отсутствует, выполняется это алгоритм.

8.3 Особенности реализации в UNIX

В LUNIX системе на 32-разрядной машине каждый процесс получает 3Гбайта виртуального пространства для себя, и 1Гбайт для страничных таблиц и других данных ядра.

На компьютерах Pentium, используется двухуровневые таблицы страниц, и размер страниц фиксирован 4Кбайта

На компьютерах Alpha, используется трехуровневые таблицы страниц, и размер страниц фиксирован 8Кбайт

Для каждой программы выделяется 3 сегмента:

1. Код программы (только для чтения)
2. Данные
3. Стек