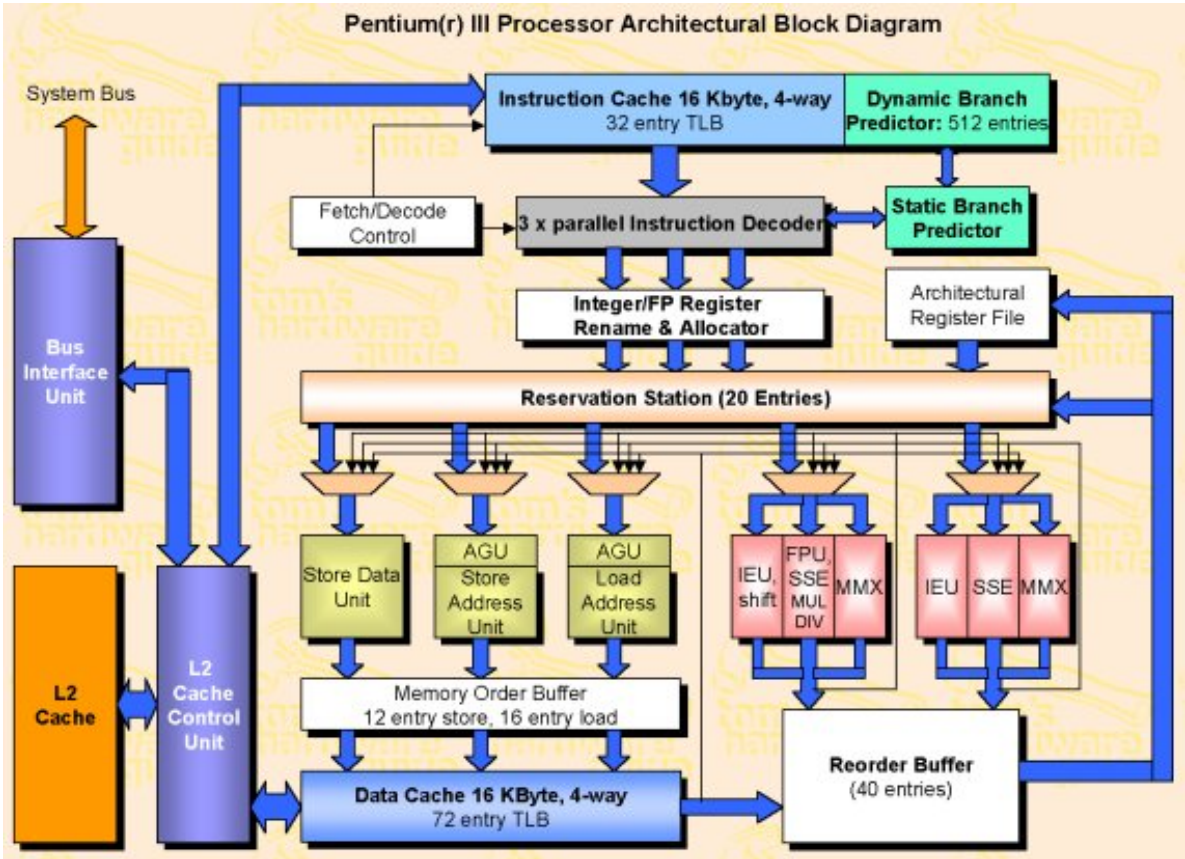
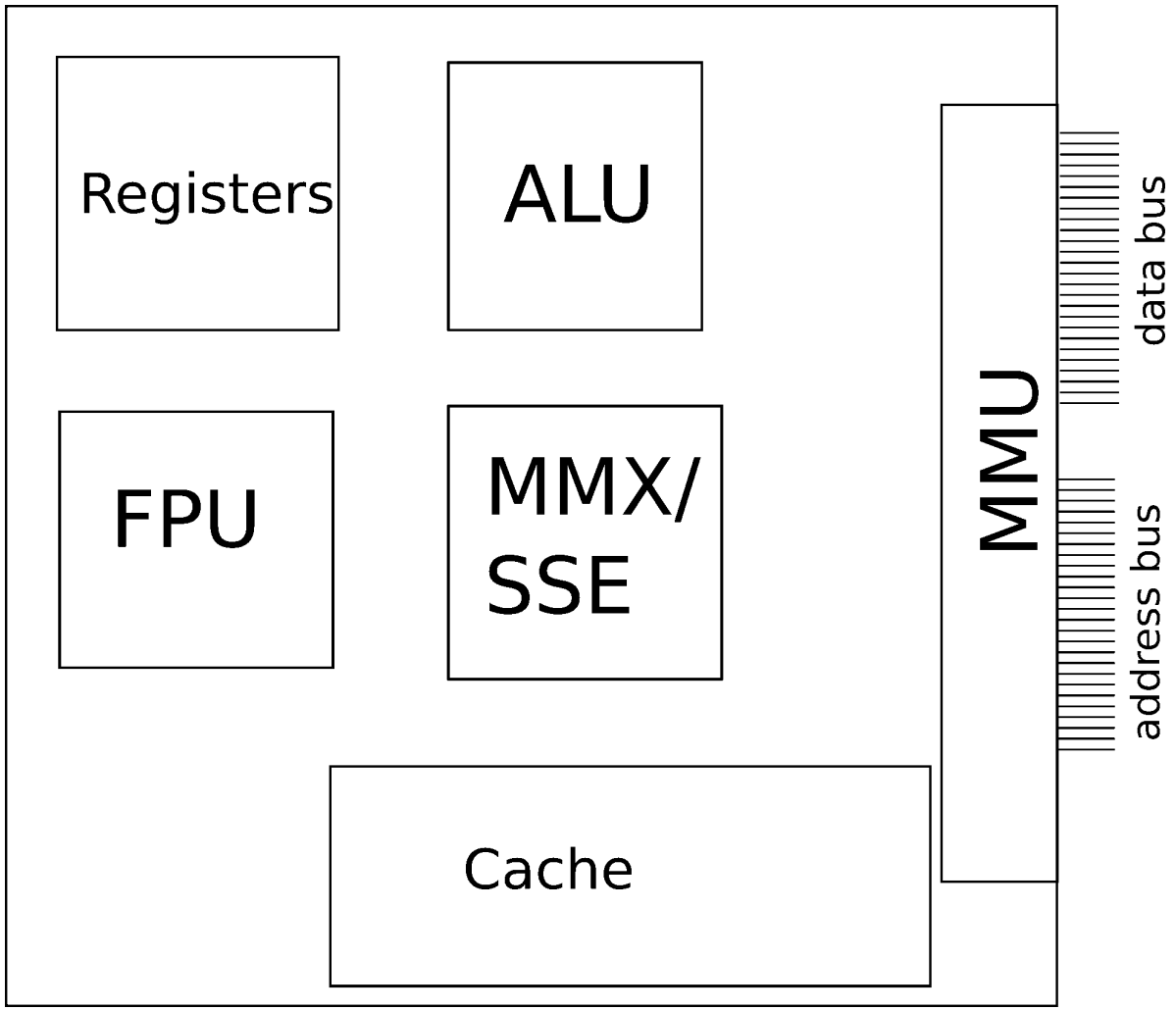
Архитектура процессоров ix86

Современный процессор — сложное электронное устройство, содержащее порядка 10^9 транзисторов.

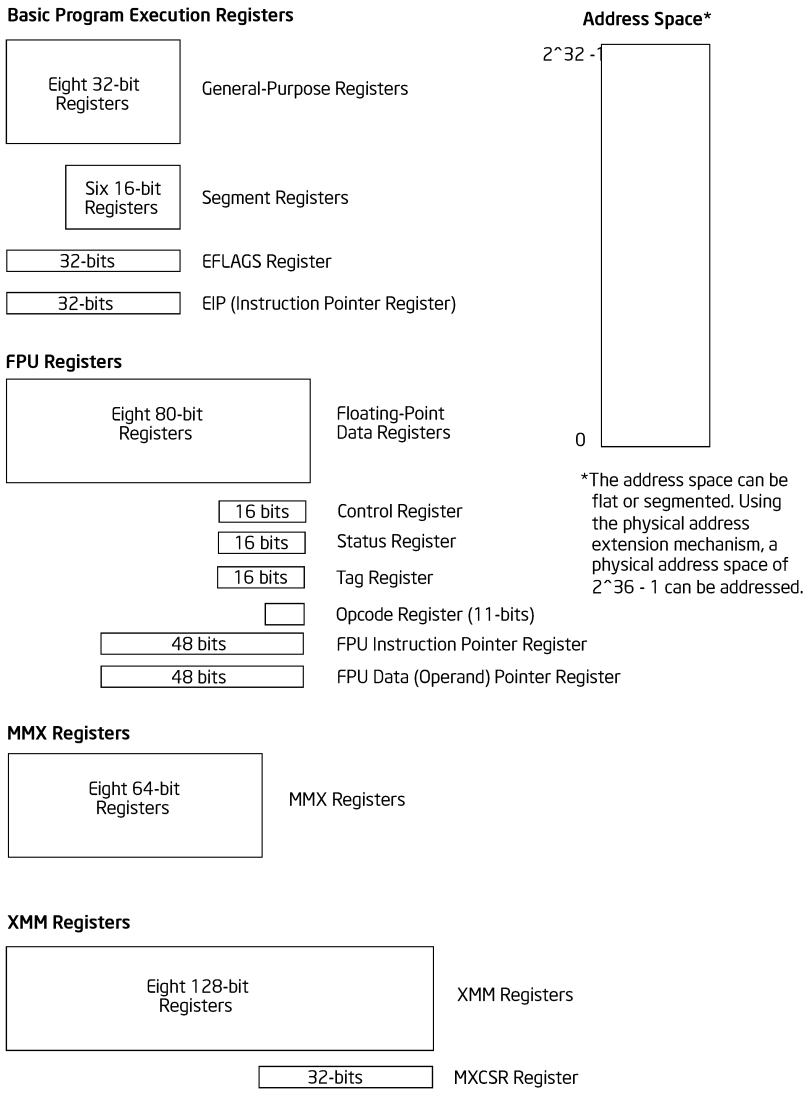


Блок-схема процессора Pentium PIII

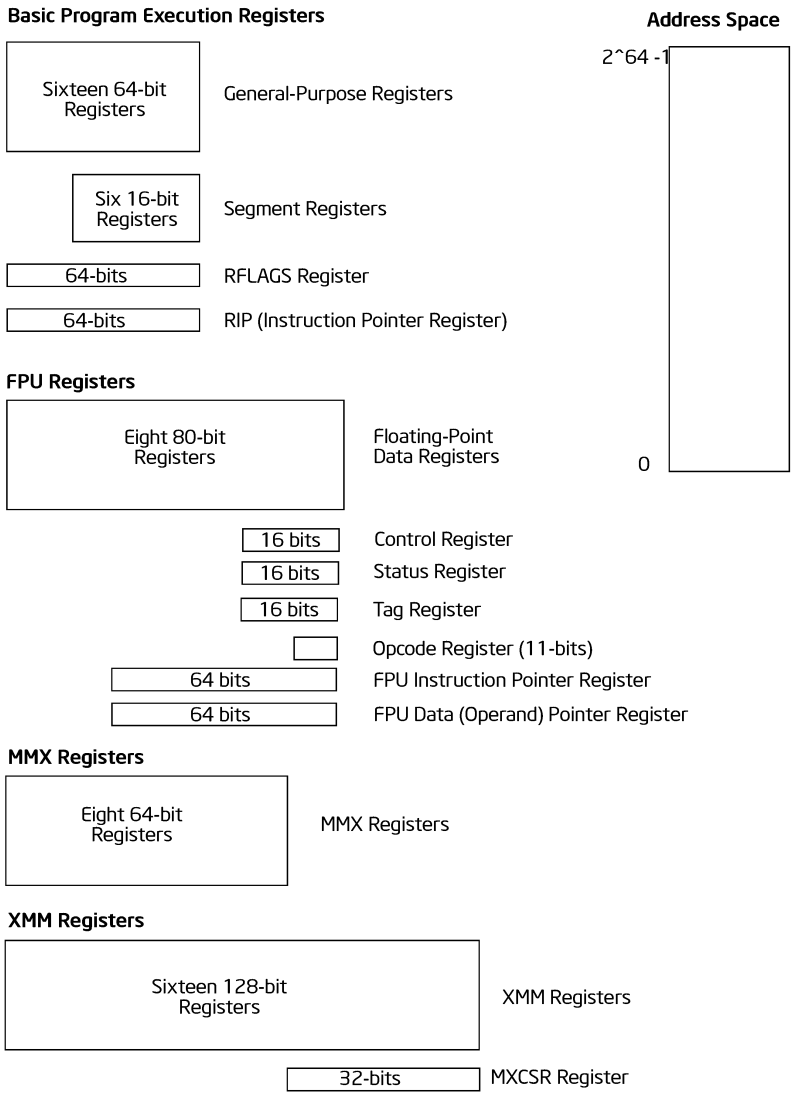
Основные блоки
Упрощённая схема



Среда выполнения процесса 32-bit



Среда выполнения процесса 64-bit

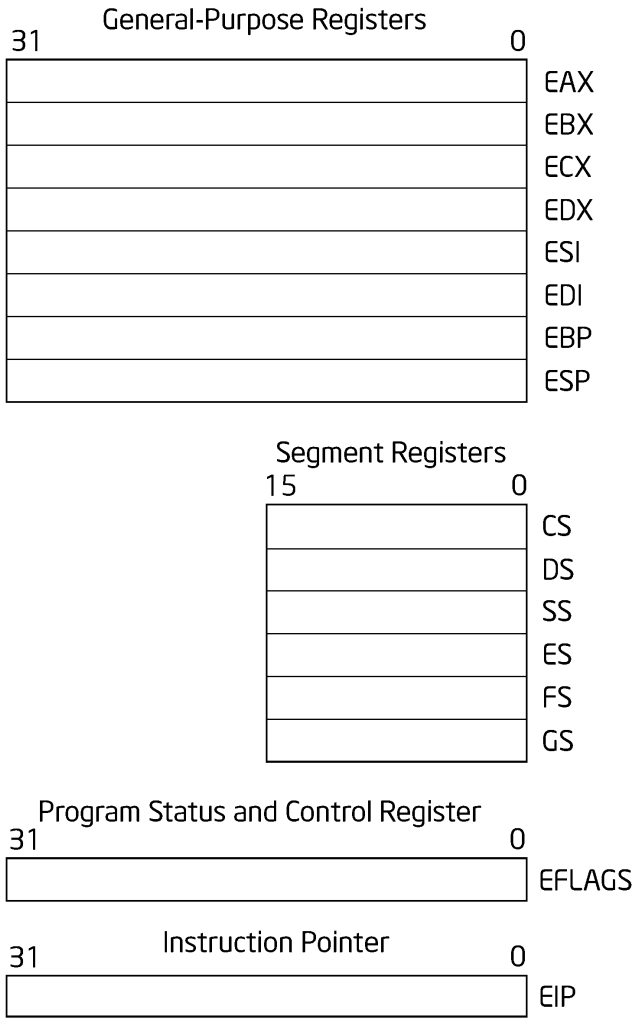


Регистры классифицируются на

- **Общего назначения** (32 bit) — EAX, EDX, ECX, EDX, EBP, ESI, EDI, ESP (+RAX–RSP,R8–R15);
- **Флагов** (32 bit) — EFLAGS (RFLAGS);
- **Указатель команд** (32 bit) — EIP (RIP);
- **Сегментные** (16 bit) — CS, DS, ES, SS, GS;
- **Управляющие** — CR0-CR3, GDTR, LDTR, IDTR, TR (+CR4, CR8) ;
- **Отладочные** — DR0–DR7;
- **Тестовые** — TR0–TR7;
- **FPU, MMX, SSE** — регистры сопроцессора, блоков MMX и SSE(XMM);
- **MSR** — model- specific registrs;
- **MTTR** — memory type registers . . .

Регистры

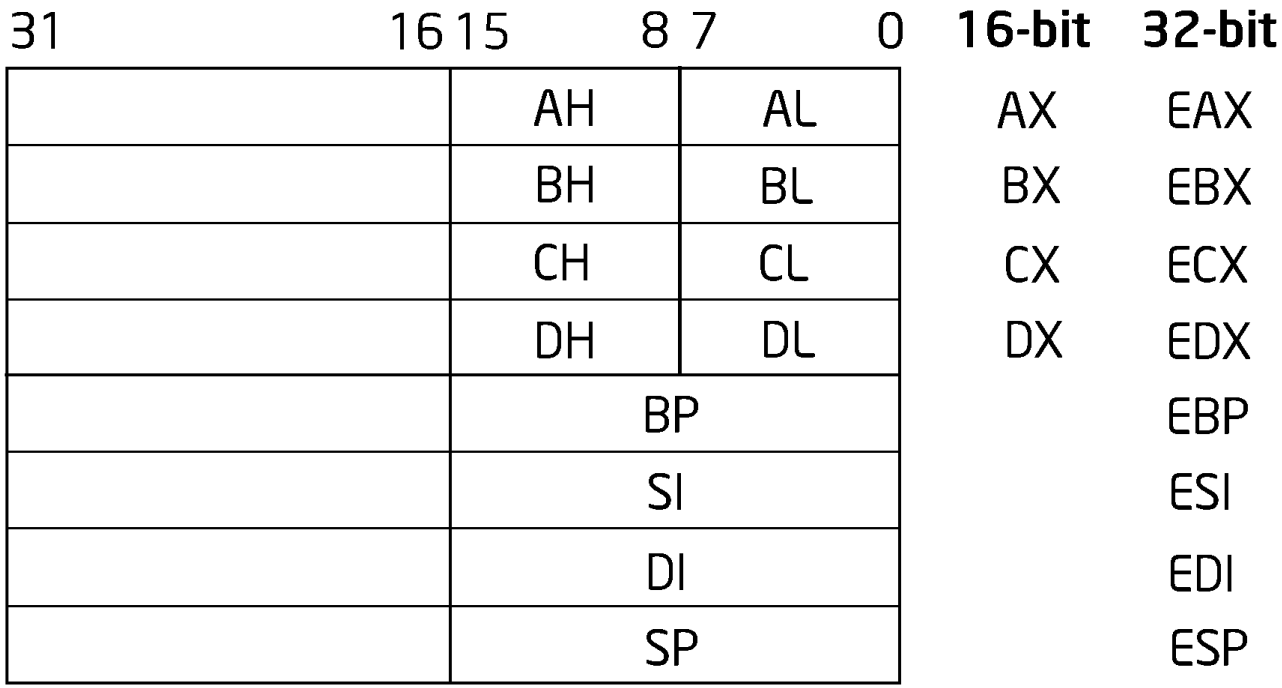
Основные регистры, доступные приложению



Регистры общего назначения

Составные части регистров

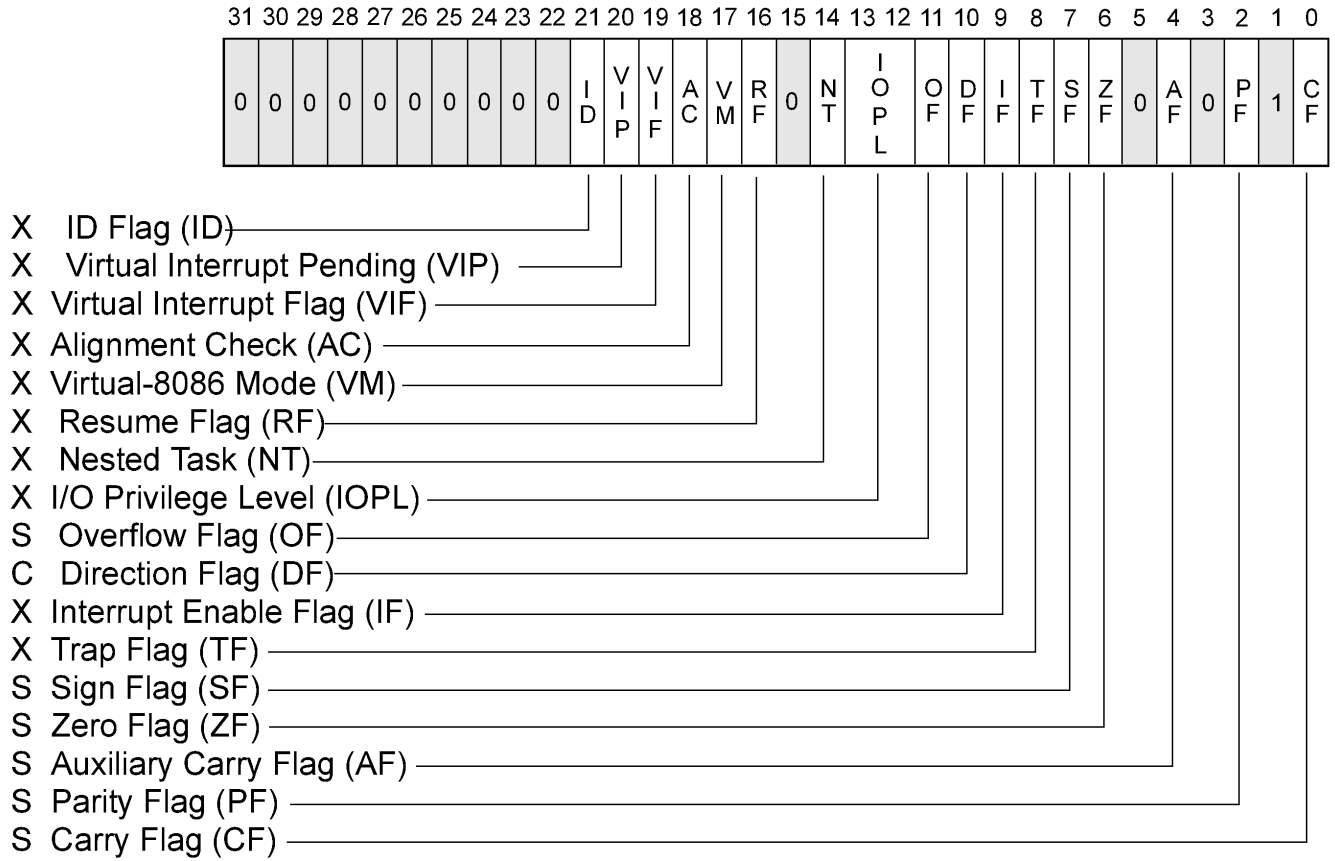
Возможен доступ к отдельным подмножествам регистров.



Регистры общего назначения
64-х битный режим

В 64-битном режиме 64-битные регистры обозначаются RAX, RBX, . . . RSP, RIP.
Добавляются еще 8 регистров: R8 — R15.
32-х битные подмножества обозначаются R8D-R15D.
16-и битные подмножества обозначаются R8W-R15W.
8-и битные подмножества обозначаются R8L-R15L.
Появляется доступ к младшим байтам других регистров: SIL, DIL, BPL, SPL.

Регистр флагов



Регистр флагов представляет собой набор бит, отображающих состояние процессора.

Пример работы с регистрами

```
MOV AL, 12;
XOR EAX, EAX;
SUB AX, 100h;
ADD EAX, EDX;
MOV EBP, ESP;
SHL ESI, 4;
PUSHF
POP ECX;

ADD EDX, var1;
MOV EAX, [ESI+12];   // DS: by default
MOV EBX, [ESP-4];    // SS: by default
ADD EBX, [EBP+20h]; // SS: by default
AND EAX, ES:[ESI+EBX*2];
```

Сегментные/селекторные регистры
Обзор

Следующие 16-ти битные регистры определяют расположение сегментов в памяти.

- **CS** — определяет сегмент кода;
- **DS** — определяет сегмент данных;
- **SS** — определяет сегмент стека;
- **ES** — определяет дополнительный сегмент (цель);
- **FS** — определяет дополнительный сегмент ;
- **GS** — определяет дополнительный сегмент.

Способ определения задаётся режимом работы процессора.

Сегментные/селекторные регистры

Visible Part	Hidden Part	
Segment Selector	Base Address, Limit, Access Information	CS
		SS
		DS
		ES
		FS
		GS

Пример работы с сегментными регистрами:

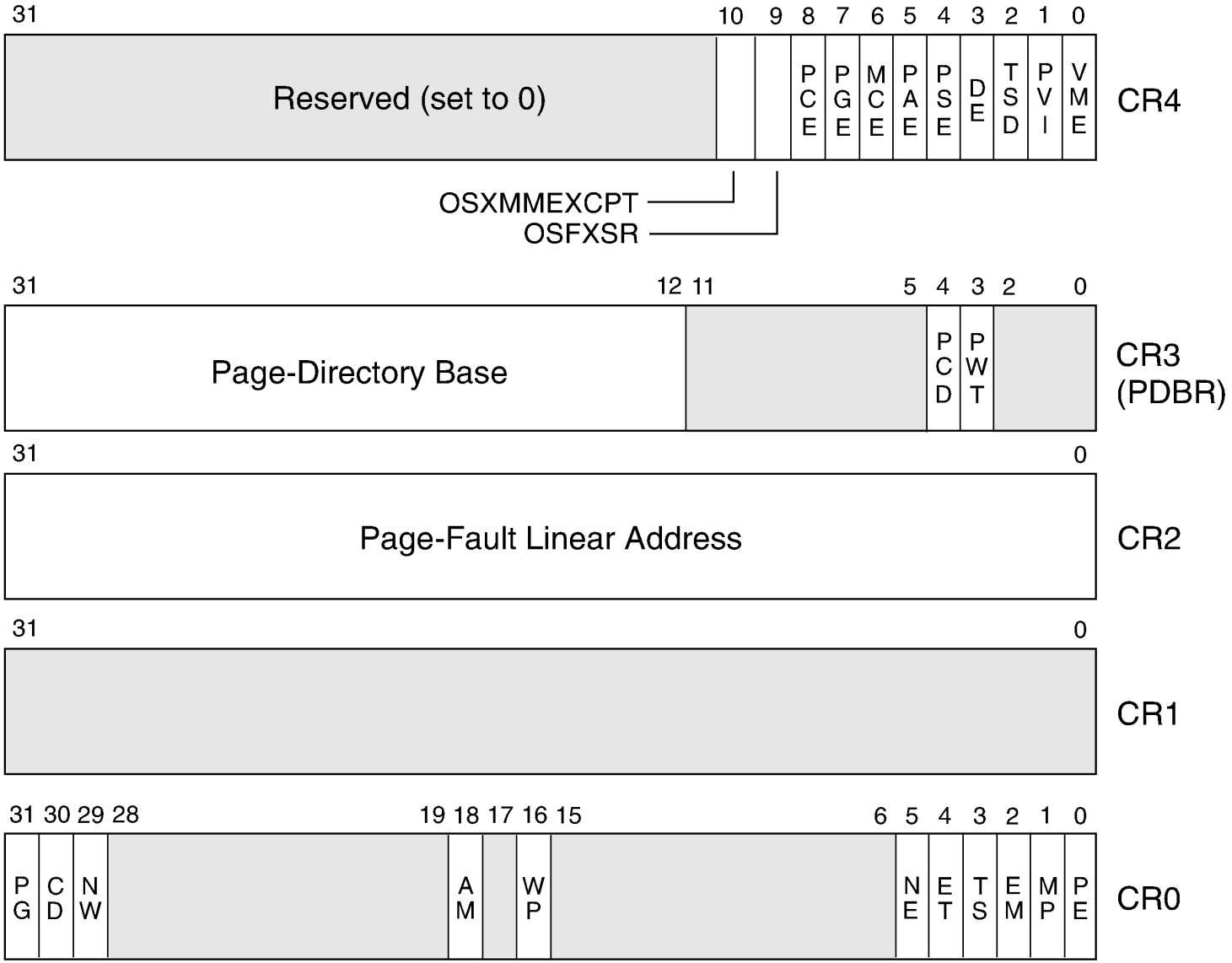
```
MOV AX, A000h;
MOV ES, AX;
PUSH ES;
POP DS;
MOV EAX, ES:[EDI+223]
```

Управляющие регистры
Обзор

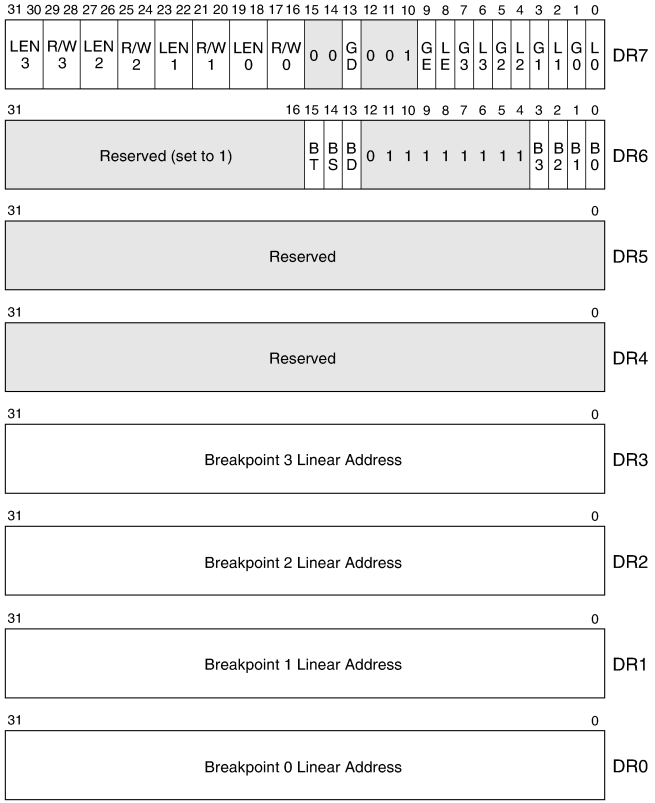
Управляющие регистры предназначены для управления режимами работы процессора, страничным и сегментным механизмом, настройкой таблиц дескрипторов.
CR0 — набор бит режима работы.
CR2 — адрес страничной ошибки.
CR3 — адрес каталога страниц+.
GDTR/LDTR — адрес GDT/LDR — глобальной/локальной таблицы дескрипторов.
IDTR — адрес IDN — таблицы дескрипторов прерываний.
TR — селектор сегмента задачи . . .

Управляющие регистры

Регистры CR0–CR4



Отладочные регистры
Обзор



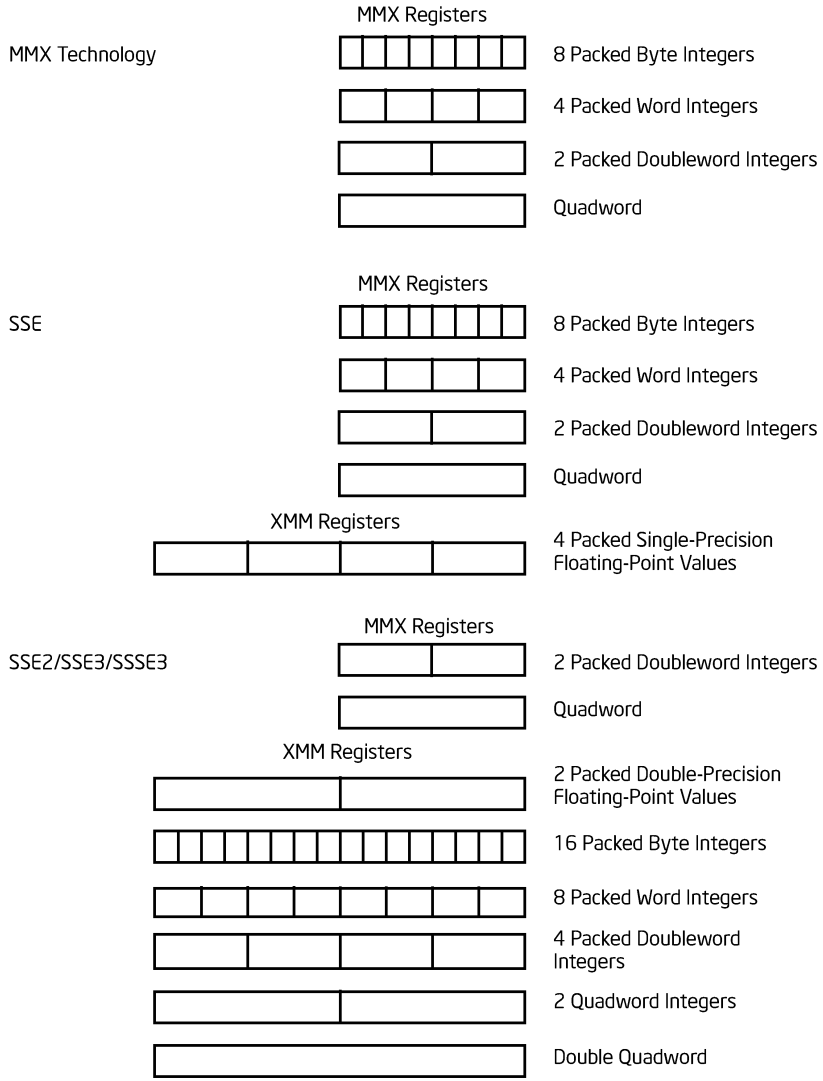
Отладочные регистры предназначены для аппаратной отладки. Позволяют установит точку останова (breakpoint) по чтению, записи и выполнению участка памяти.

Тестовые регистры
Обзор

Тестовые регистры TR0–TR7 предназначены для управления системой многоуровнего кеширования.

Регистры MMX, SSE ...

Обзор



Режимы работы процессора

У процессора есть несколько режимов работы.

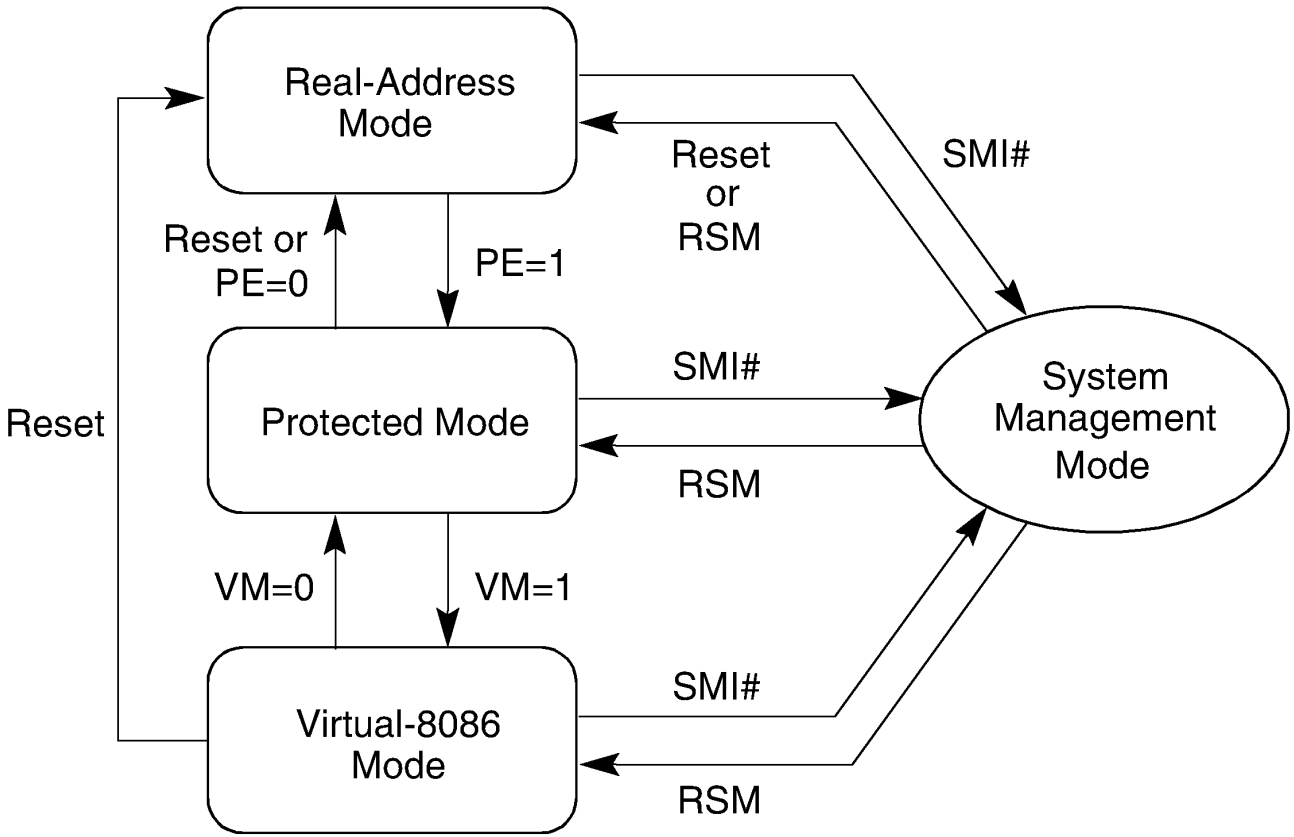
реальный режим — адресное пространство 20 бит, защиты нет, по умолчанию — 16-ти битные регистры.

зашищённый режим — основной режим для современных ОС. Адресное пространство — 32/36/40/64 бита. Есть защита, сегментно-страничная организация памяти. По умолчанию - 32/64 битные регистры.

режим эмуляции V86 — предназначен для эмуляции старых ОС.

SMM — специальный режим для аппаратно-зависимых действий.

Режимы работы процессора

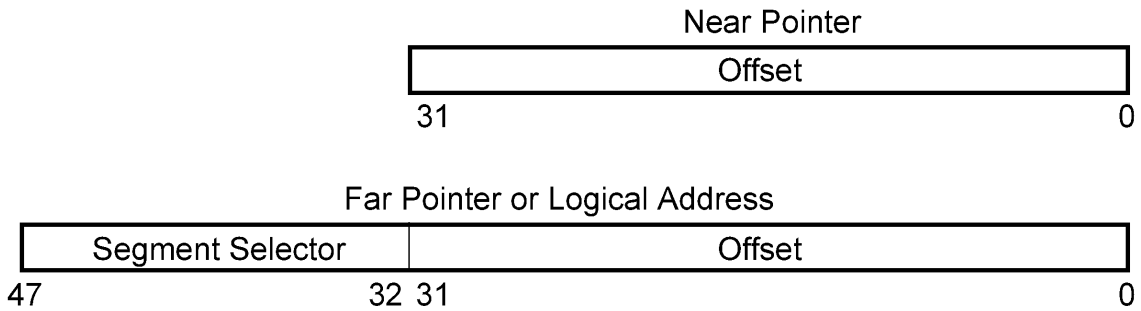


Способы указания логического адреса

Логический адрес — это адрес, как его задаёт программа.
Физический адрес — адрес, выставляемый процессором на шину адреса.

При адресации объектов в памяти возможны 2 случая:

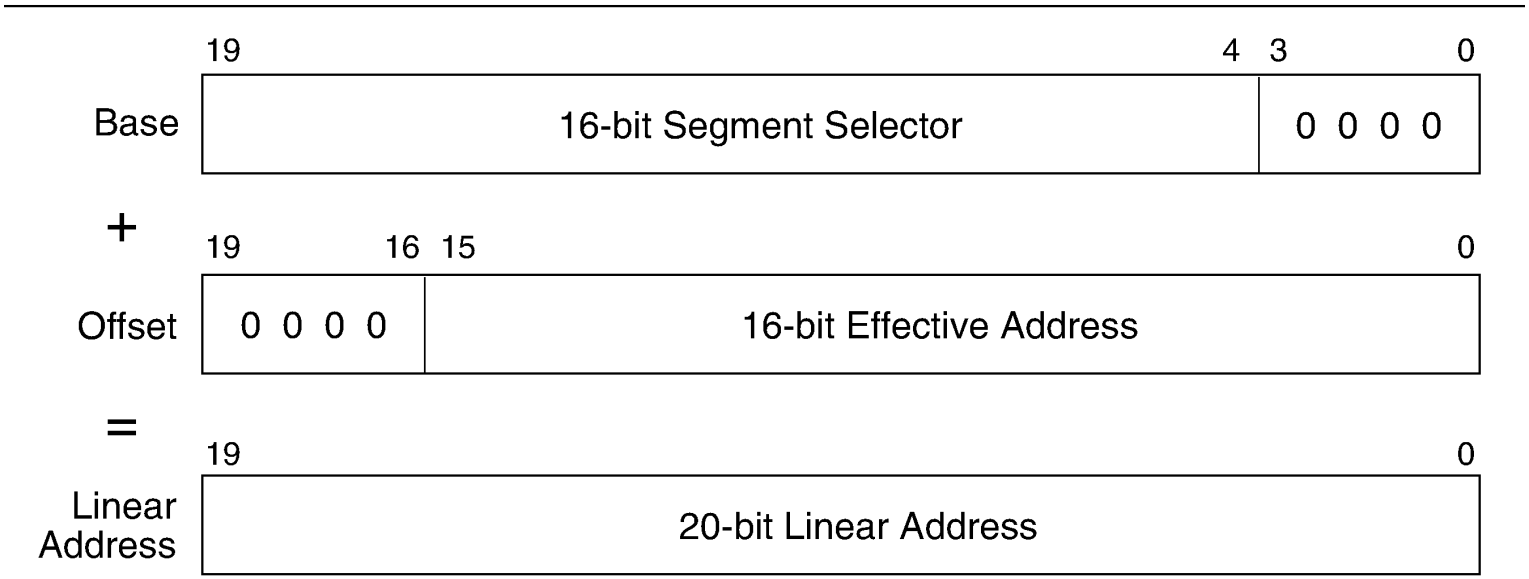
- Указывается только смещение — ближний (near) адрес. Сегмент - по умолчанию.
- Указывается и сегмент, и смещение — дальний (far) адрес.



Адресация в реальном режиме работы

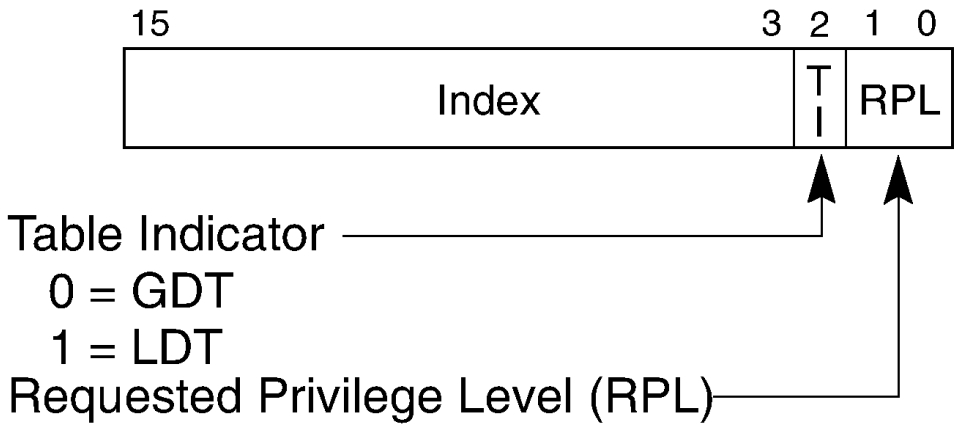
В реальном режиме значение сегментного регистра (16 bit) сдвигается влево на 4, и складывается со смещением (16 bit):

```
phys_addr = linear_addr = segment << 4 + offset;
```



Адресация в защищённом режиме работы
Логический адрес

В защищенном режиме смещение составляет 32 бита. Сегментная компонента логического адреса определяет дескриптор сегмента в одной из таблиц дескрипторов.



Адресация в защищённом режиме работы

Регистры таблиц дескрипторов

Регистр **GDTR** (48 бит) непосредственно описывает расположение и размер глобальной таблицы дескрипторов (GDT). Аналогично **IDTR** — таблицу дескрипторов прерываний (IDT).

System Table Registers

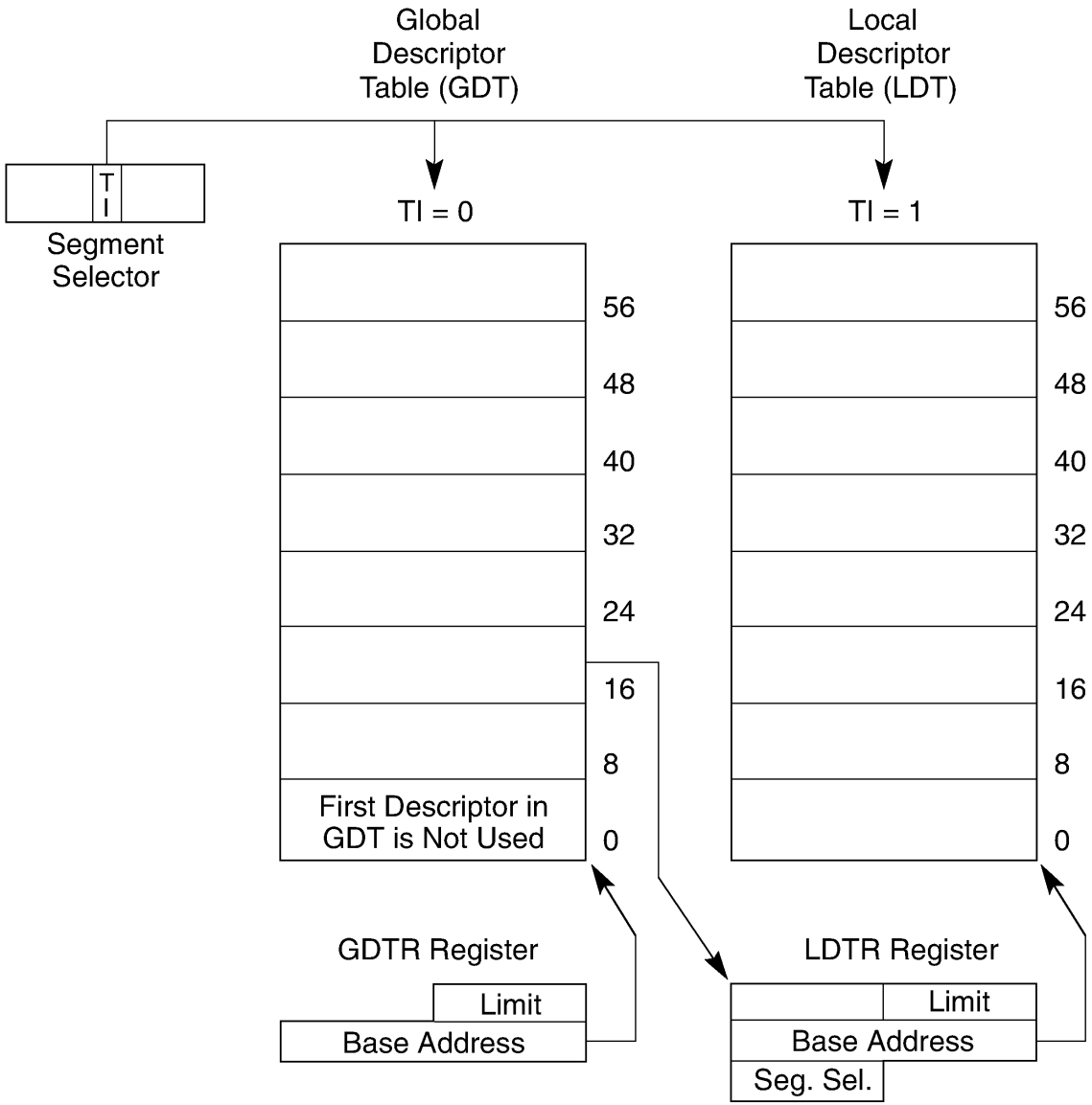
	47 — 16	15 — 0
GDTR	32-bit Linear Base Address	16-Bit Table Limit
IDTR	32-bit Linear Base Address	16-Bit Table Limit

	System Segment Registers 15 — 0	**Segment Descriptor Registers (Automatically Loaded)**				
						Attributes
Task Register	Seg. Sel.	32-bit Linear Base Address	Segment Limit			
LDTR	Seg. Sel.	32-bit Linear Base Address	Segment Limit			

Регистры **LDTR** и **TR** описывают расположение своих структур косвенно, через элементы GDT.

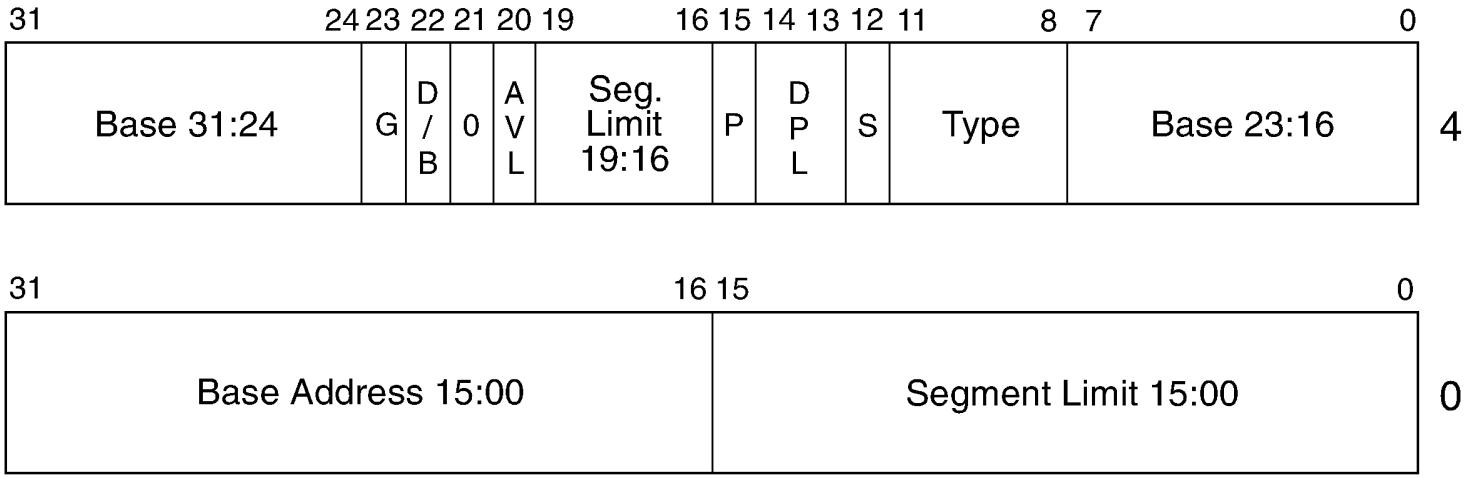
Адресация в защищённом режиме работы

Таблицы дескрипторов



Адресация в защищённом режиме работы

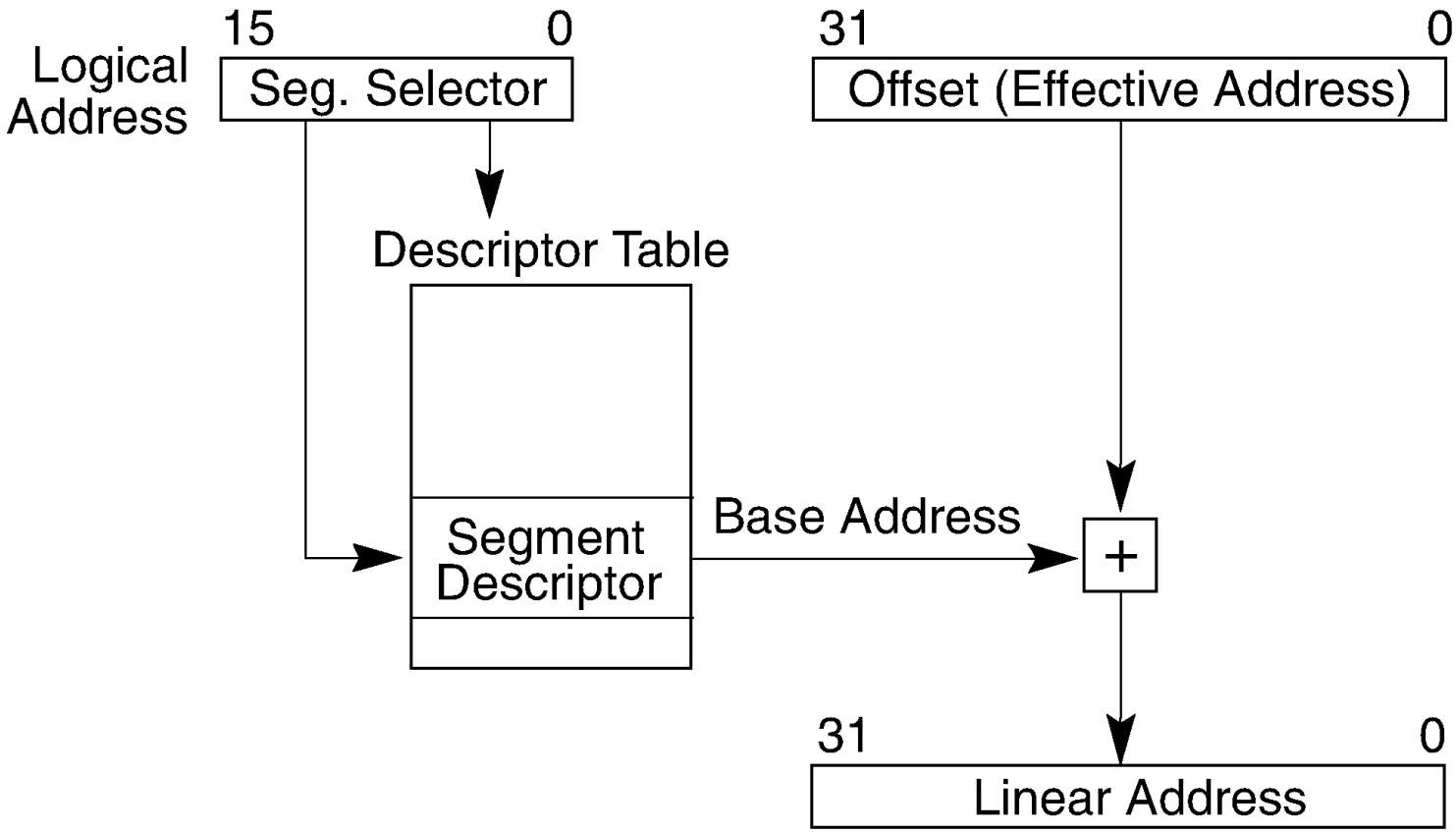
Описание сегментов



AVL — Available for use by system software
BASE — Segment base address
D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
DPL — Descriptor privilege level
G — Granularity
LIMIT — Segment Limit
P — Segment present
S — Descriptor type (0 = system; 1 = code or data)
TYPE — Segment type

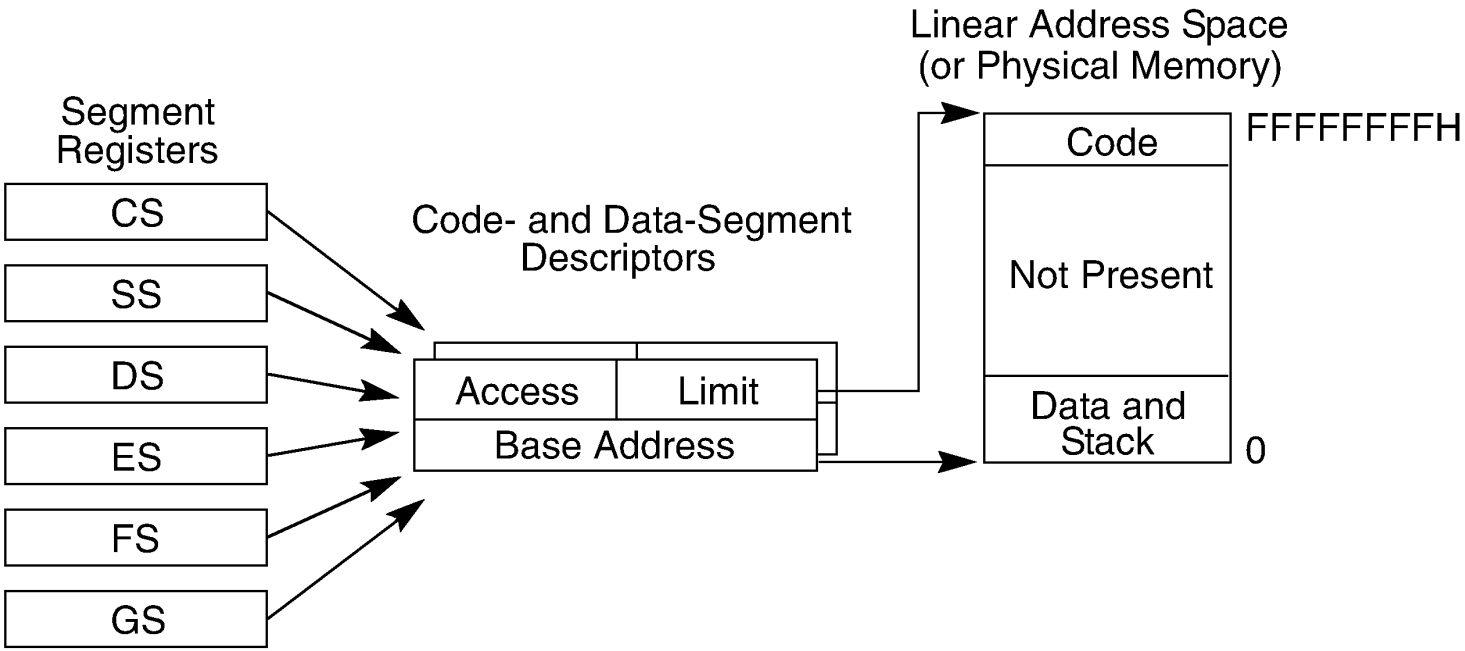
Адресация в защищённом режиме работы

Получение линейного адреса



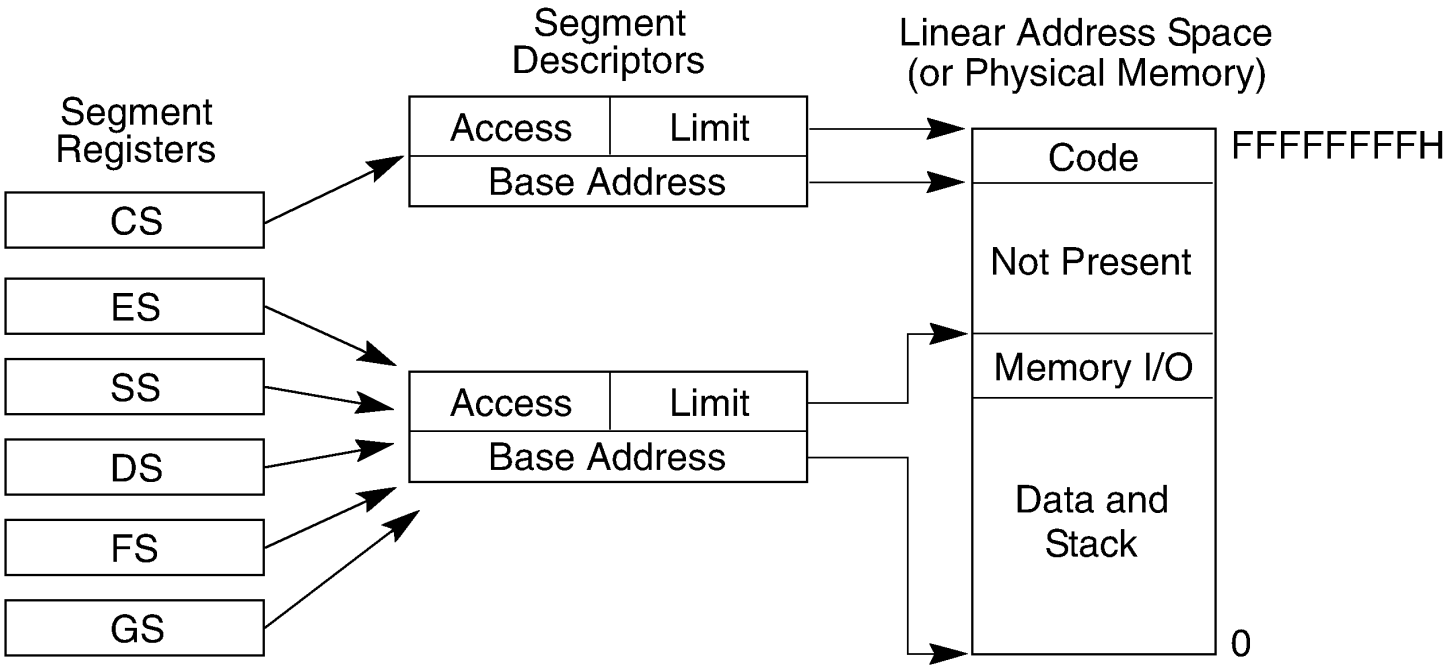
Примеры организации сегментной модели

Модель плоской (flat) памяти



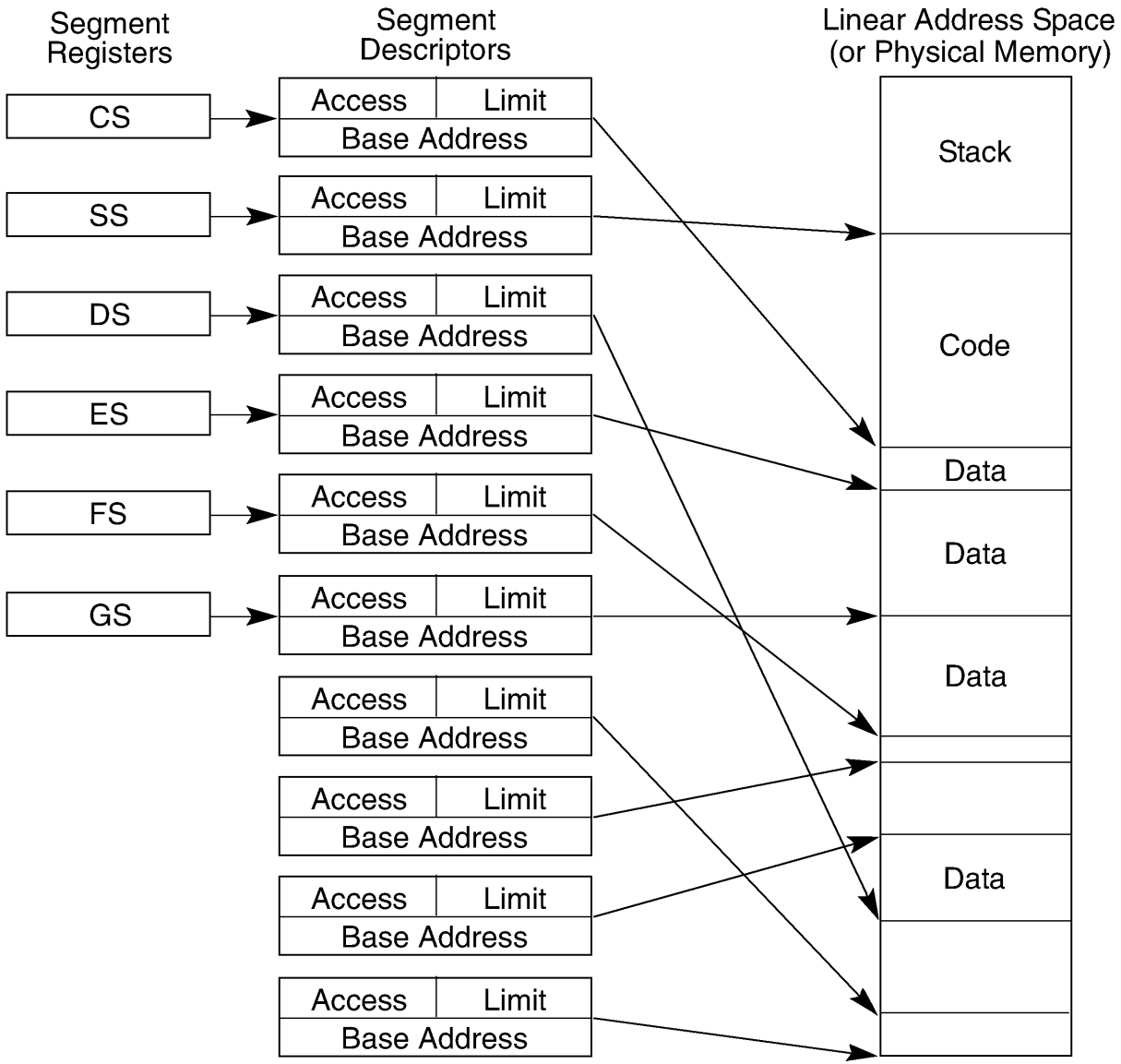
Примеры организации сегментной модели

Модель плоской памяти с защитой



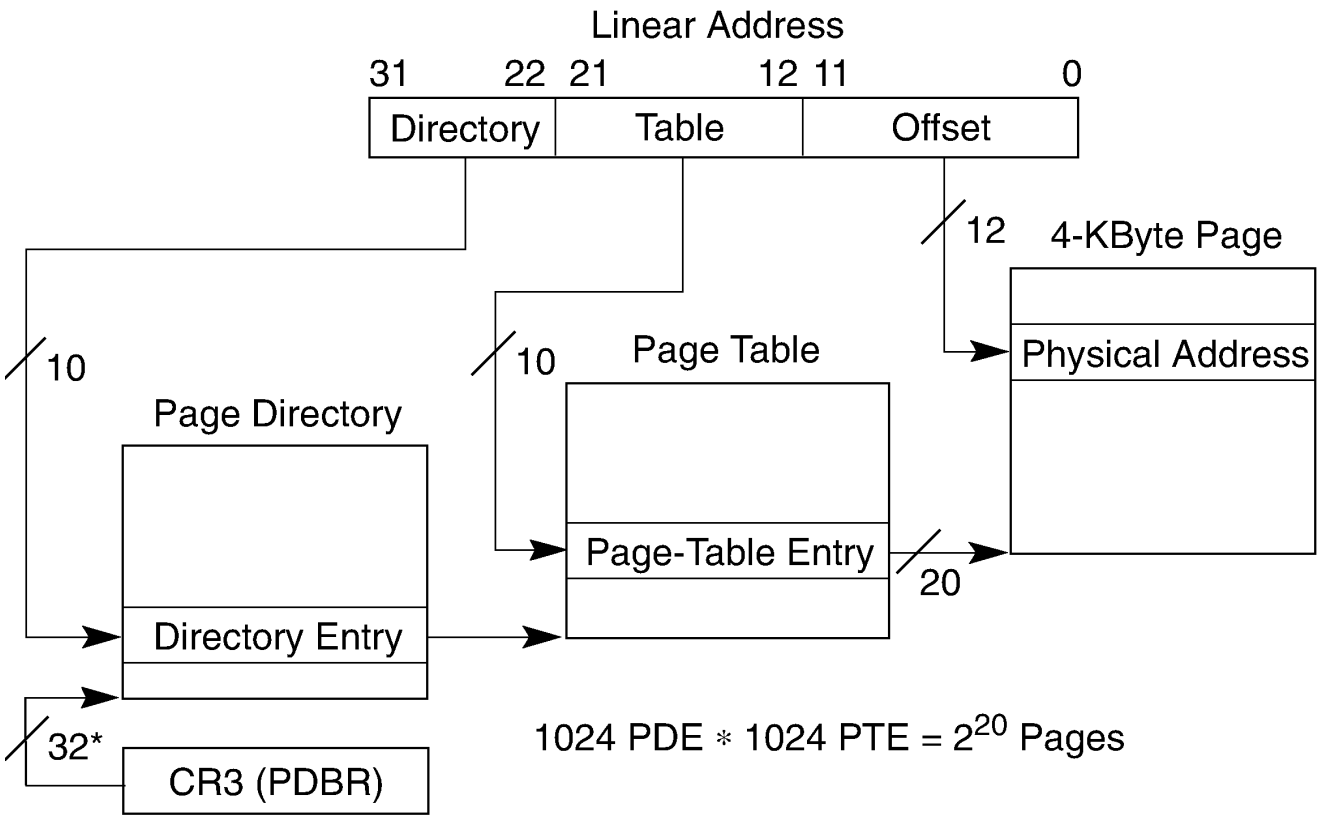
Примеры организации сегментной модели

Множество сегментов



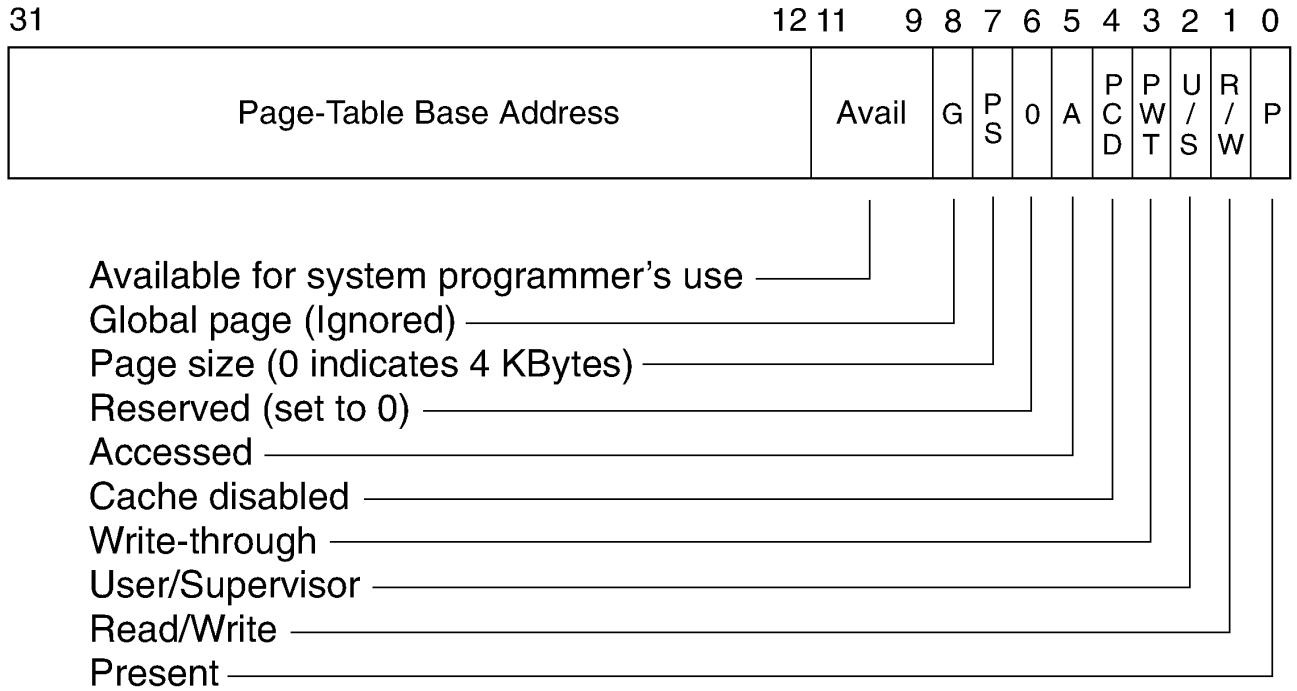
Организация страничной памяти

При сброшенном бите **PG** из **CR0** линейный адрес является физическим. При установленном — включается механизм **страничного** преобразования (paging).



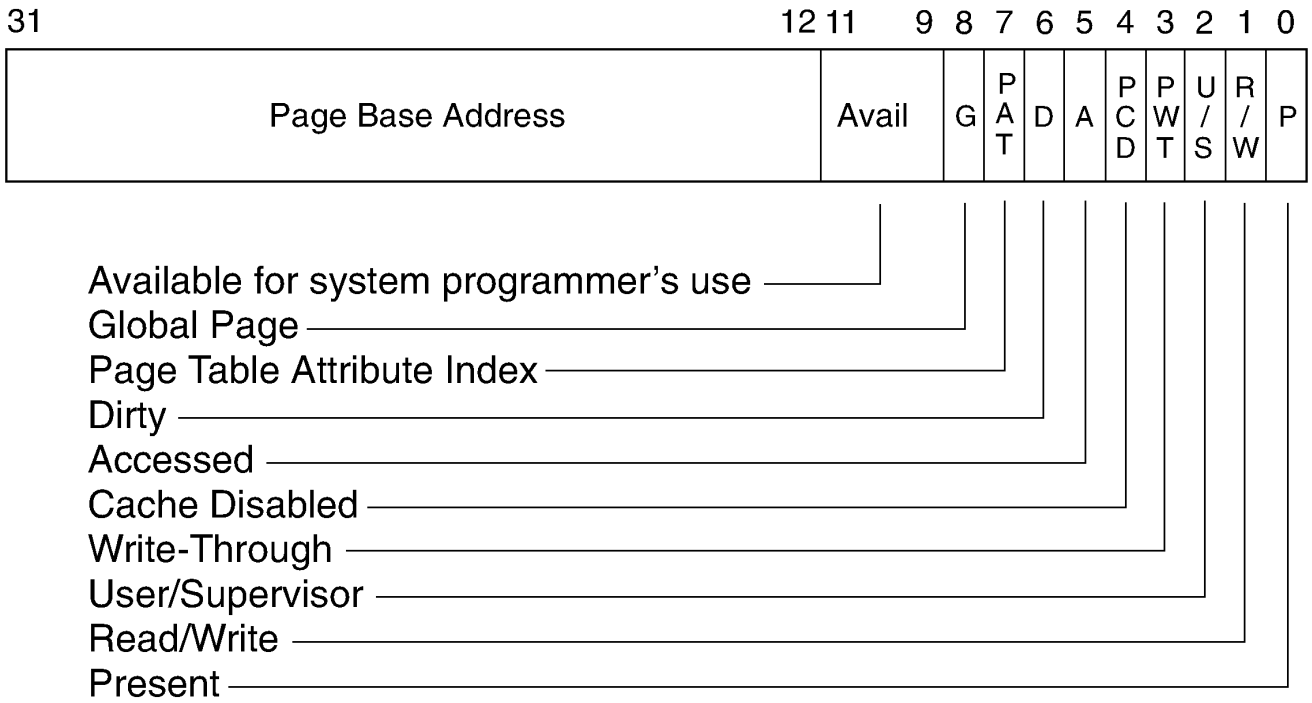
Организация страничной памяти
Элемент каталога страниц

Каталог страниц занимает 1 страницу. Его расположение указывается регистром **CR3**. Элементы определяют расположение и параметры таблиц страниц.



Организация страничной памяти
Элемент таблицы страниц

Таблица страниц занимает 1 страницу. Их расположение указывается записями каталога таблиц страниц. Элементы определяют расположение и свойства страниц памяти.



Бит **P** (Present) установлен, когда описываемая страница в самом деле присутствует в памяти.

Установленный бит **R/W** (Read/Write) разрешает запись в данную страницу.

Бит **A** (Access) устанавливается при первом доступе к странице.

Бит **D** (Access) устанавливается при записи в страницу.

При попытке доступа к отсутствующей странице, записи в страницу “только для чтения”, доступе к системной странице из процесса пользователя происходит **страничное исключение** (INT 0Eh), причём адрес и причина ошибки помещается в **CR2**.

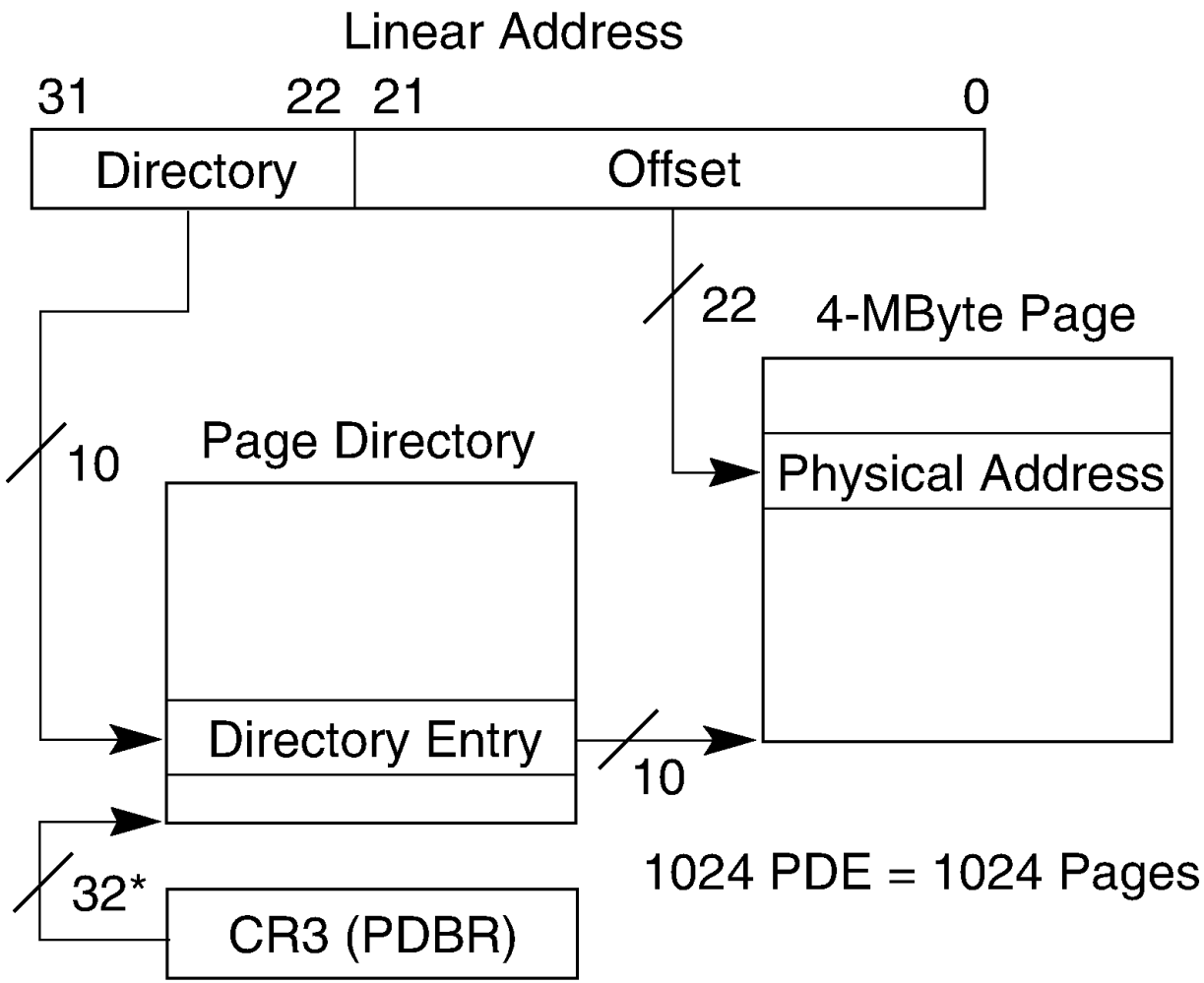
Виртуальная память

Использование страничного механизма позволяет организовать виртуальную память.
Возможности и преимущества:

- Логическое адресное пространство — непрерывное, физическое расположение — произвольное.
- Процессу выделяются только те страницы памяти, которые реально используются.
- При нехватке физической памяти можно организовать страничный обмен с внешними устройствами (разделами или файлами подкачки — swap).
- Одинаковые сегменты только для чтения в разных процессах могут состоять из одних и тех же физических страниц.
- Простая реализация механизма COW — Copy On Write.
- ...

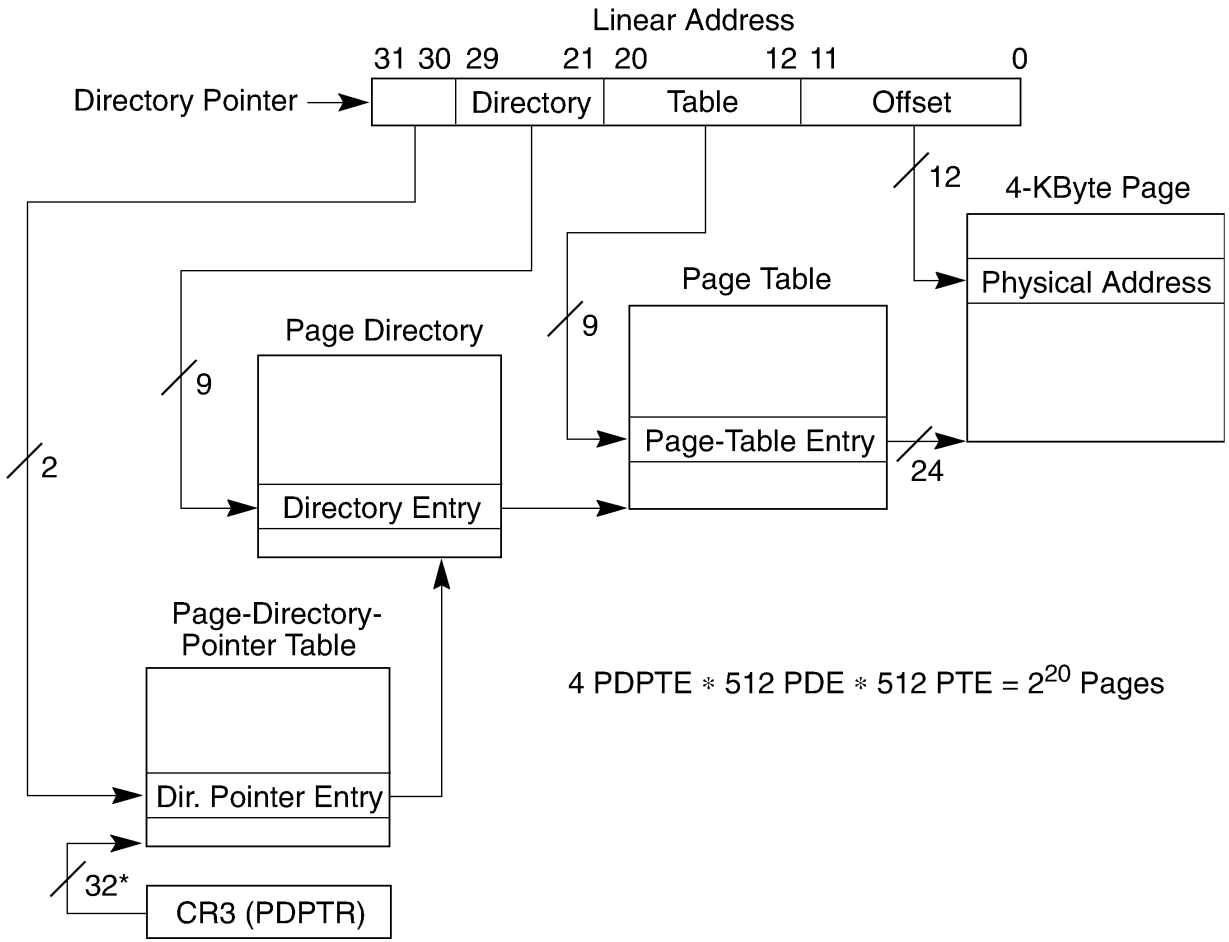
Организация страничной памяти
Использование страниц в 4MB

Возможно использование больших страниц в 4MB.
Способ адресации немного отличается:

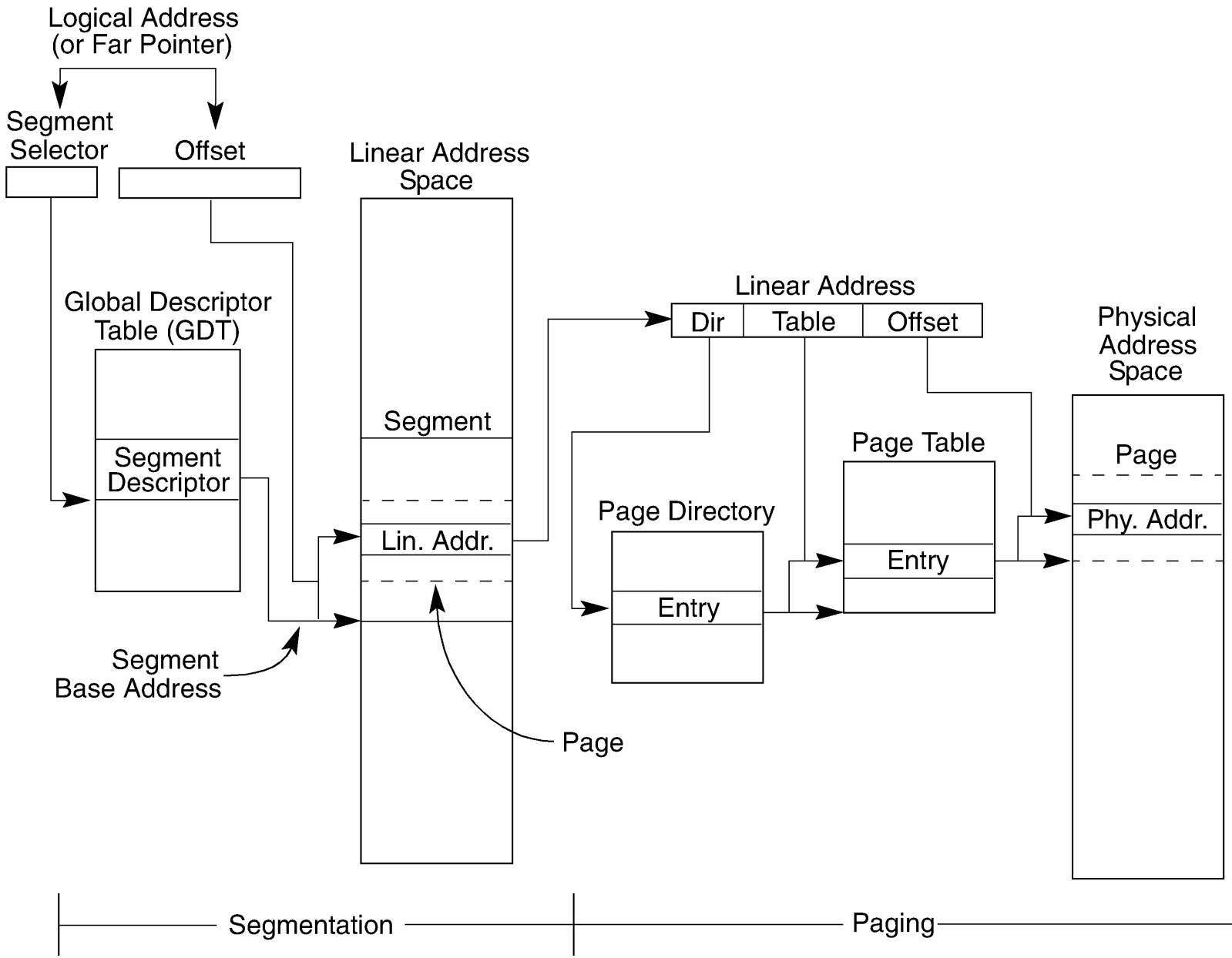


Организация страничной памяти
Использование расширение PAE

Начиная с Pentium Pro, проявилась возможность расширить физическое адресное пространство до 64GB при использовании PAE:



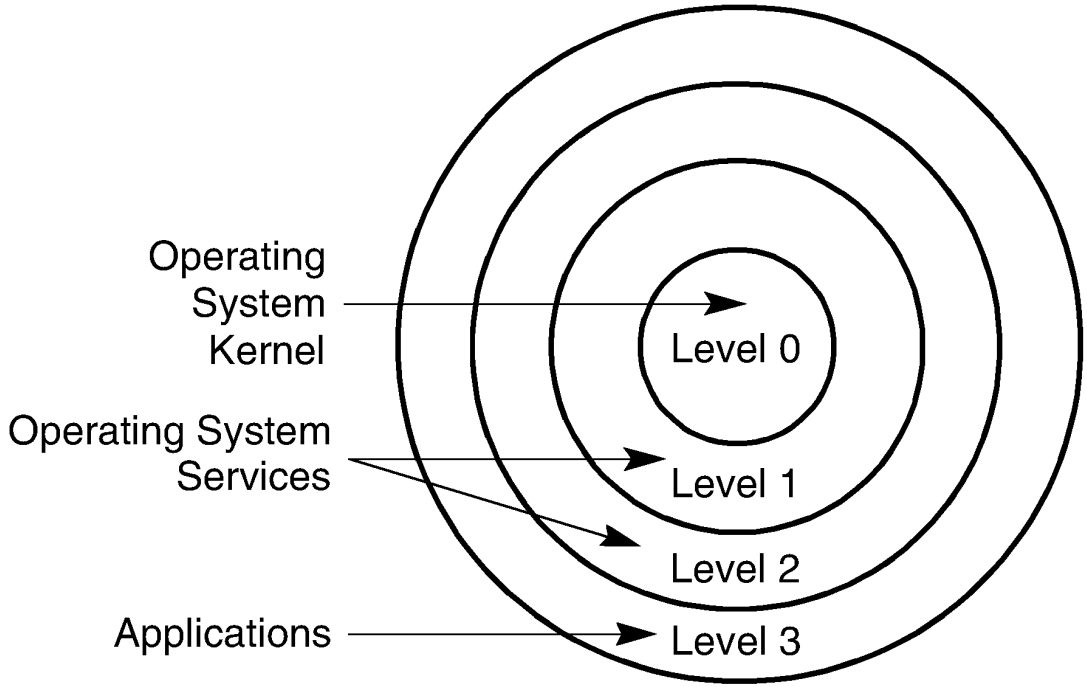
Сегментный и страничный механизм



Режимы доступа в защищённом режиме

4 кольца защиты

Поле DPL текущего сегмента кода определяет текущий уровень привилегий (CPL).



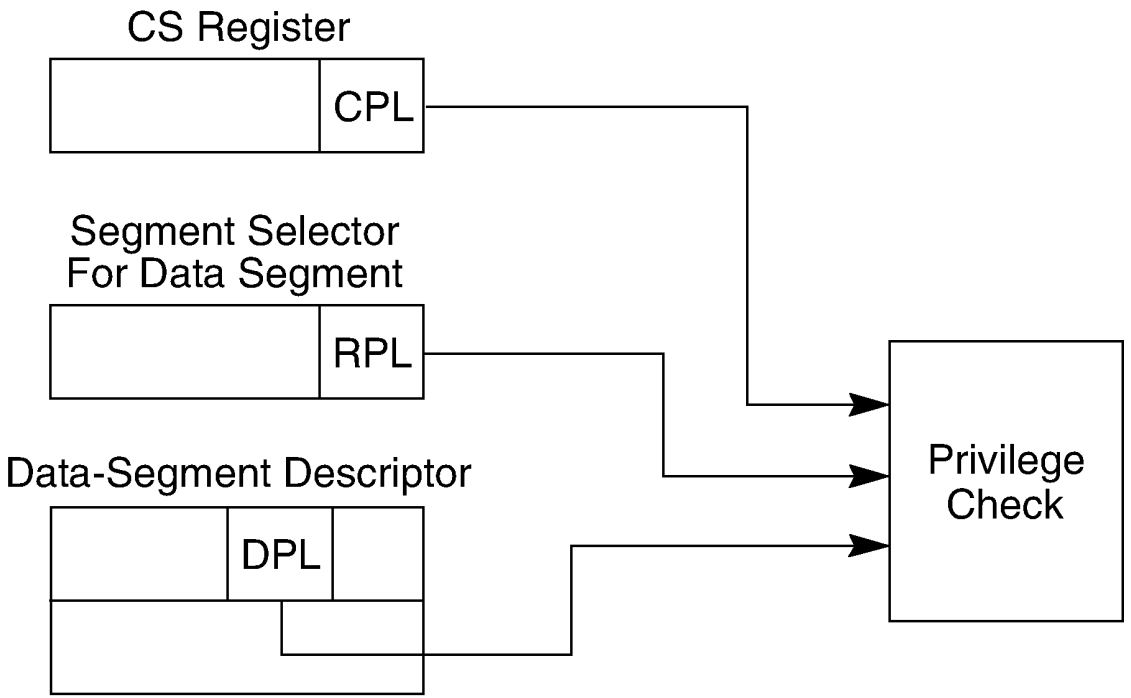
Изменение системных регистров, запрещение/разрешение прерываний и другие опасные действия разрешены только на уровне 0.

Режимы доступа в защищённом режиме

Проверка доступа

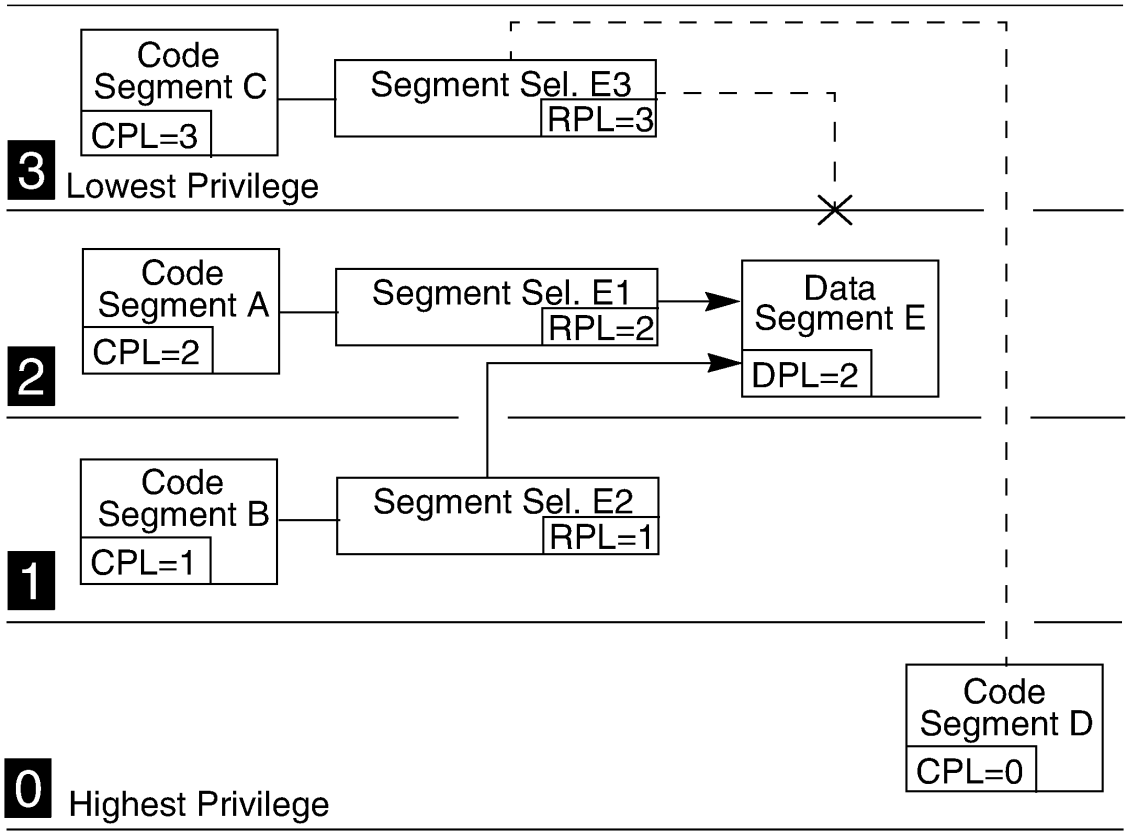
При загрузке значений в селекторные регистры производится проверка доступа. Код может загружать только селекторы с таким же или большим значением EPL.

$$CPL \leq EPL = \max(RPL, DPL)$$



Режимы доступа в защищённом режиме

Пример проверка доступа



Общая ошибка защиты

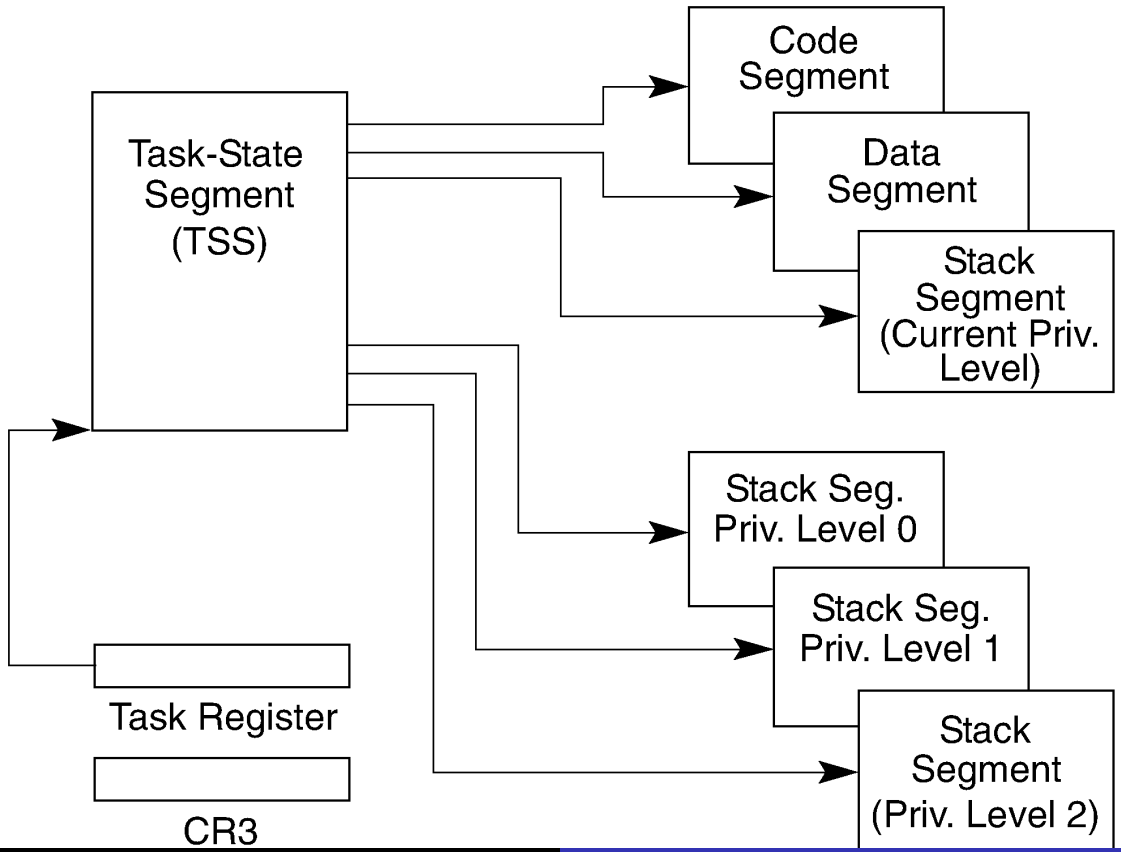
При попытке загрузить селектор, для которого не выполняется хотя бы 1 из 15 условие защиты, происходит общая ошибка защиты (GPF) = INT 0D.

Управление передаётся операционной системе, которая и принимает необходимые меры.

Для передачи управления управления от менее привилегированного кода к более (например, системный вызов) используется механизм **шлюзов** или **программных прерываний**.

Состояние задачи

Для поддержки переключения между задачами средствами ОС существуют специальные сегменты — **TSS** — Task State Segment. Эти сегменты хранят текущее состояние процесса при переключении задач. Специальный регистр TR определяет TSS текущего процесса.



Сегмент состояния задачи

В TSS сохраняются как регистры (автоматически), так и произвольная информация, требуемая операционной системой.

31 — 15	0	
I/O Map Base Address	T	100
LDT Segment Selector		96
GS		92
FS		88
DS		84
SS		80
CS		76
ES		72
EDI		68
ESI		64
EBP		60
ESP		56
EBX		52
EDX		48
ECX		44
EAX		40
EFLAGS		36
EIP		32
CR3 (PDBR)		28
SS2		24
ESP2		20
SS1		16
ESP1		12
SS0		8
ESP0		4
Previous Task Link		0