ЗАЩИТА ПО ПРИВИЛЕГИЯМ

Специфической особенностью защищенного режима является защита по привилегиям. В процессоре имеется два уровня защиты: защита на уровне сегментов и защита на уровне страниц.

Защита на уровне сегментов

Защита на уровне сегментов представлена четырьмя уровнями привилегий. Наиболее привилегирован нулевой уровень, наименее привилегирован - третий. В реальных операционных системах обычно не используются все четыре уровня. UNIX и Windows, например, используют только два уровня привилегий - 0 (для ядра системы) и 3 (для всего остального).

Для описания механизма защиты пользуются следующими понятиями:

Уровень привилегий дескриптора (Descriptor Privilege Level: DPL) - уровень привилегий, на который помещен описываемый дескриптором объект. Поле DPL хранится в байте прав доступа дескриптора.

Текущий уровень привилегий (Current Privilege Level: CPL) - уровень привилегий выполняемого сегмента кода. Это значение соответствует DPL сегмента кода (кроме подчиняемых сегментов кода). Значение CPL хранится в поле RPL селектора сегмента кода, который помещен в регистр CS.

Запрашиваемый уровень привилегий (Requested Privilege Level: RPL) - используется для временного понижения своего уровня привилегий при обращении к памяти. RPL заносится в младшие биты селектора.

Уровень привилегий ввода-вывода (Input/Output Privilege Level: IOPL) - указывает какой уровень привилегирован для работы с портами ввода-вывода. Это значение хранится в регистре EFLAGS и может быть различным для разных задач.

Защита на уровне сегментов состоит в защите от выполнения привилегированных команд, защите доступа к данным и защите сегментов кода.

В процессоре есть команды, которые могут кардинально изменить состояние всей системы. Такие команды выполняются только на нулевом уровне привилегий, а на всех других уровнях вызывают нарушение общей защиты (исключение #13). К этим командам относятся:

HLT - останов процессора;
CLTS - сброс флажка Task Switched (исп. при управлении мультизадачностью);
LIDT, LGDT, LLDT - загрузка регистров дескрипторных таблиц;
LTR - загрузка регистра задачи;
LMSW - загрузка младшего слова регистра CR0;
MOV CRx,reg32 - работа с управляющими регистрами;
MOV DRx,reg32 - работа с регистрами отладки;

•

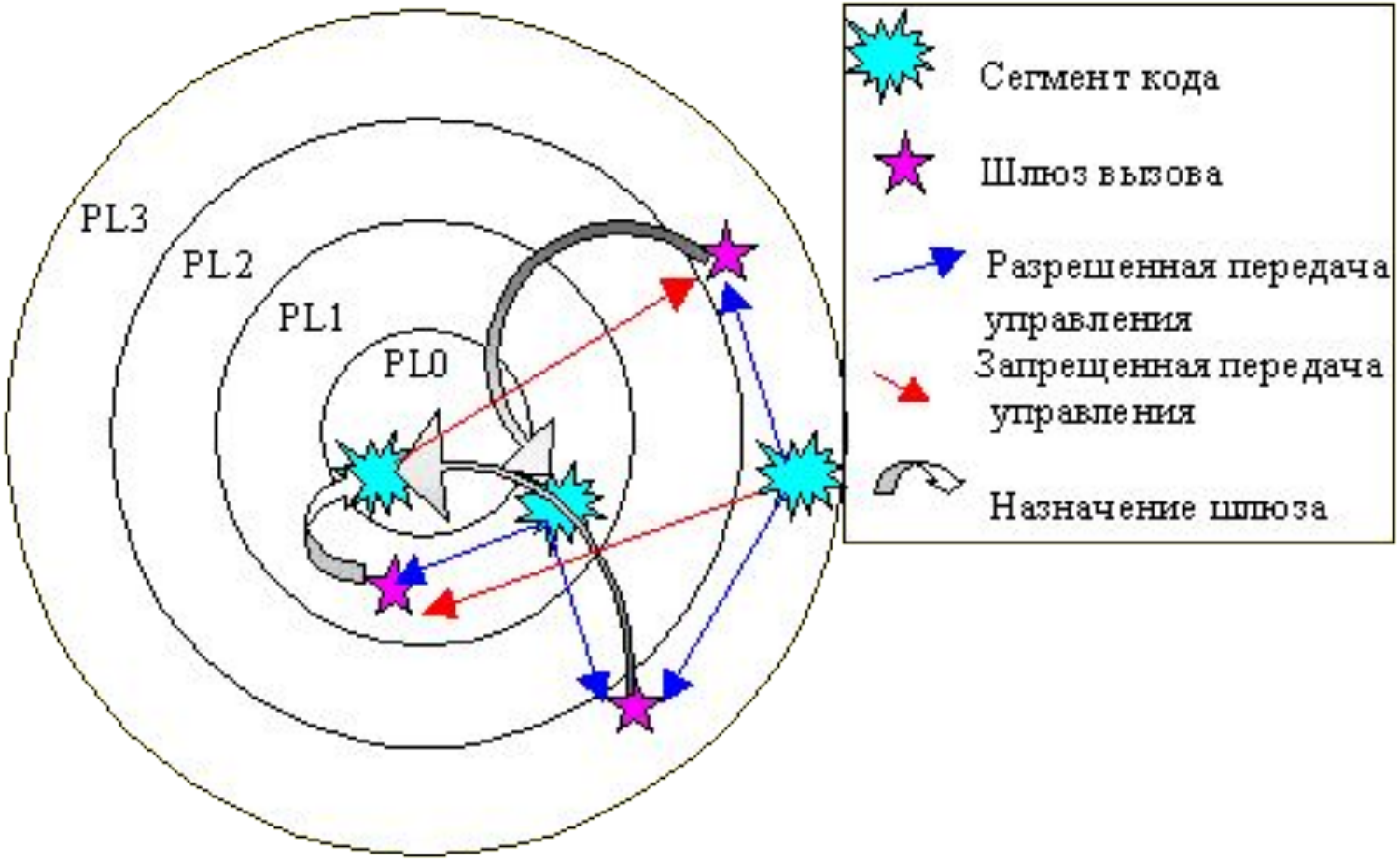
- команды работы со специфическими регистрами (TRx - для 386,486; MCRs - для Pentium и P6; MTRRs - для P6). Следует отметить, что команда **POPFD** также чувствительна к уровню привилегий. Она не изменяет состояние управляющих флажков IOPL, IF и др., если выполняется на уровне привилегий, отличном от нулевого.

Кроме безусловно привилегированных команд есть команды чувствительные к уровню привилегий ввода-вывода. Это команды работы с портами (**IN, INS, OUT, OUTS**) и команды сброса/установки флажка разрешения прерываний (**CLI, STI**). Эти команды выполняются только в том случае, если **CPL<=IOPL**. Если это условие не выполняется, то для команд ввода-вывода производится дополнительная сверка с картой разрешения портов ввода-вывода.

Второй аспект защиты - защита доступа к данным. Код имеет право обратиться к данным, которые находятся на том же или на более низком уровне привилегий. При этом учитывается не только CPL, но и RPL. Данные доступны, если дескриптор сегмента данных имеет **DPL>=max(CPL,RPL)**.

Такой контроль производится при загрузке селекторов в сегментные регистры (DS,ES,FS,GS). В сегментный регистр можно загрузить только селектор доступного с текущего уровня привилегий сегмента данных или, если сегментный регистр не будет использоваться, пустой селектор. Попытка нарушить правило привилегий или загрузить селектор системного дескриптора или дескриптора сегмента кода только для выполнения в сегментный регистр данных приведет к нарушению общей защиты (исключение #13). Кроме того, в командах изменения данных в памяти производится проверка на возможность записи в сегмент.

Особое правило привилегий для сегментов стека. Стек должен находиться строго на том же уровне привилегий, что и код программы (**DPL=CPL**). При этом сегмент стека обязательно должен быть присутствующим (P=1) и для него должны быть доступны операции и чтения, и записи. Для защиты сегментов кода используется жесткое правило привилегий: **DPL=CPL**.
Т.е. межсегментные команды **FAR JMP** и **FAR CALL** могут передавать управление сегментам кода в пределах того же уровня привилегий. Исключением являются подчиненные сегменты кода. При передаче управления подчиненному сегменту действует правило: **DPL>=max(CPL,RPL)**.Однако при этом подчиненный код будет выполняться на том же уровне привилегий, что и вызвавший его код (CPL не изменится).
Для передачи управления между уровнями привилегий используются системные дескрипторы, называемые *шлюзами вызова*. Дескриптор шлюза вызова содержит точку входа в привилегированную процедуру (селектор: смещение) и число передаваемых ей через стек параметров. Для передачи управления привилегированной процедуре адресуется не сама процедура, а шлюз к ней. Шлюзы можно адресовать только в команде **FAR CALL**, т.е. "насовсем" сменить уровень привилегий таким способом нельзя, всегда предполагается возврат на более низкий уровень привилегий. Правило разрешения вызова через шлюз выглядит так:
DPLцели<=max(CPL,RPL)<=DPLшлюза.
Примеры разрешенных переходов показаны на картинке.


PL3
PL2
PL1
PL0
Сегмент кода
Шлюз вызова
Разрешенная передача управления
Запрещенная передача управления
Назначение шлюза

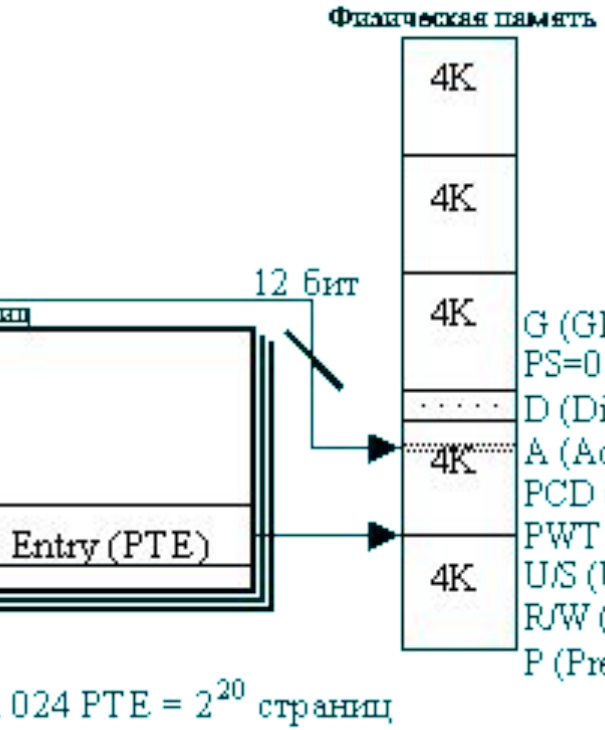
При переключении уровней привилегий происходит переключение стека. При этом из внешнего (менее привилегированного) стека происходит копирование указанного в шлюзе числа параметров во внутренний стек. Перед этим во внутреннем стеке сохраняется указатель вершины внешнего стека. После копирования параметров во внутренний стек заносится адрес возврата. Привилегированная процедура должна заканчиваться инструкцией **RETF n**, где n - число байт, занимаемых параметрами в стеке.

Смена уровня привилегий, происходящая при передаче управления, автоматически вызывает переопределение стека. Начальное значение указателя стека *SS:SP* для уровня привилегий 0, 1, 2 содержится в *TSS.* При передаче управления по командам *JMP* или *CALL* в *CS:SP* загружается новое значение указателя стека, а старые значения помещаются в новый стек. При возврате на прежний уровень привилегий его стек восстанавливается (как часть инструкции *RET или IRET).* Для вызовов подпрограмм с передачей параметров через стек и сменой уровня привилегий из предыдущего стека в новый копируется фиксированное число слов, заданное в вентиле. Команда межсегментного возврата *RET c* выравниванием указателя стека при возврате корректно восстановит значение предыдущего указателя.

и уровни привилегий сегментов, а уровню пользователя - 3-й. Для каждой страницы указывается (в PDE/PTE) Защита страниц обеспечивается двумя уровнями привилегий: супервизор и пользователь. Уровню супервизора соответствуют 0-й, 1-й, 2-й уровни привилегий сегментов, а уровню пользователя - 3-й. Для каждой страницы указывается (в PDE/PTE), с какого уровня привилегий она доступна. Правило привилегий таково: с уровня супервизора доступны все страницы, а с уровня пользователя - только страницы с битом U/S=1. Кроме того, в PDE/PTE указывается тип доступа к странице (доступна ли страница для записи): R/W=0 - только чтение, R/W=1 - доступны чтение и запись. Следует отметить действие бита 16 (Write Protect) в регистре CR0. Когда этот бит выставлен, страницы уровня пользователя с пометкой "read-only" защищены от записи при обращениях с уровня супервизора.

В процессорах, начиная с Pentium, страницы могут иметь размер 4Кбайт или 4Мбайт (эта возможность называется *расширением размера страниц*), а в процессорах с архитектурой P6 при включенном *расширении физического адреса* - 4Кбайт или 2Мбайт. (Расширение физического адреса заключается в использовании 36-битного физического адреса вместо 32-битного.) Расширение размера страниц включается установкой бита 4 (Page Size Extension) в регистре CR4 - 4Кбайт или 2Мбайт. (Расширение физического адреса заключается в использовании 36-битного физического адреса вместо 32-битного.) Расширение размера страниц включается установкой бита 4 (Page Size Extension) в регистре CR4, а расширение физического адреса - установкой бита 5 (Physical Address Extension) в регистре CR4. Обе возможности работают только в

PG (CR0)	PAE (CR4)	PSE (CR4)	PS (PDE)	Размер страницы	Разрядность физического адреса
0	x	x	x	-	32 бит
1	0	0	x	4K	32 бит
1	0	1	0	4K	32 бит
1	0	1	1	4M	32 бит
1	1	x	0	4K	36 бит
1	1	x	1	2M	36 бит



Формат PDE/PTE

31 … 12	11 … 9	8	7	6	5	4	3	2	1	0
Base Address	Avail.	G	PS	D	A	PCD	PWT	U/S	R/W	P

G (Global page) – PTE : управление кэшированием

PS=0 (Page Size) – PDE : управление размером страниц

D (Dirty) – PTE : управление кэшированием

A (Accessed) – PDE/PTE : бит обращения

PCD (Cache Disable) – PDE/PTE : управление кэшированием

PWT (Write-through) – PDE/PTE : управление кэшированием

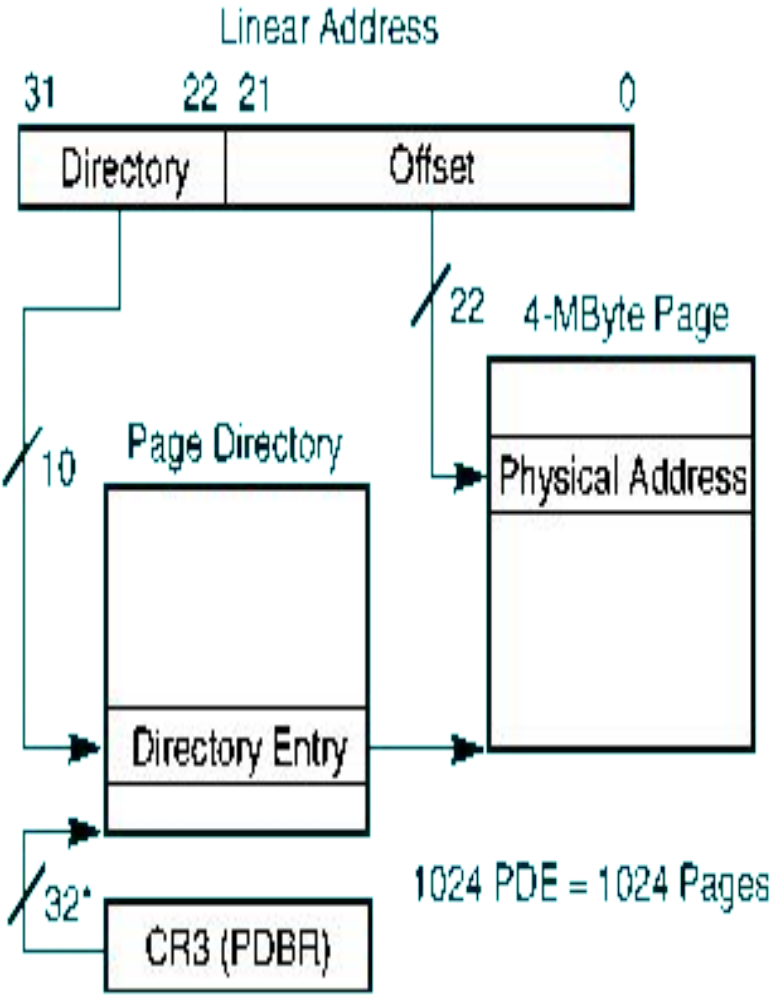
U/S (User/Supervisor) – PDE/PTE : права доступа

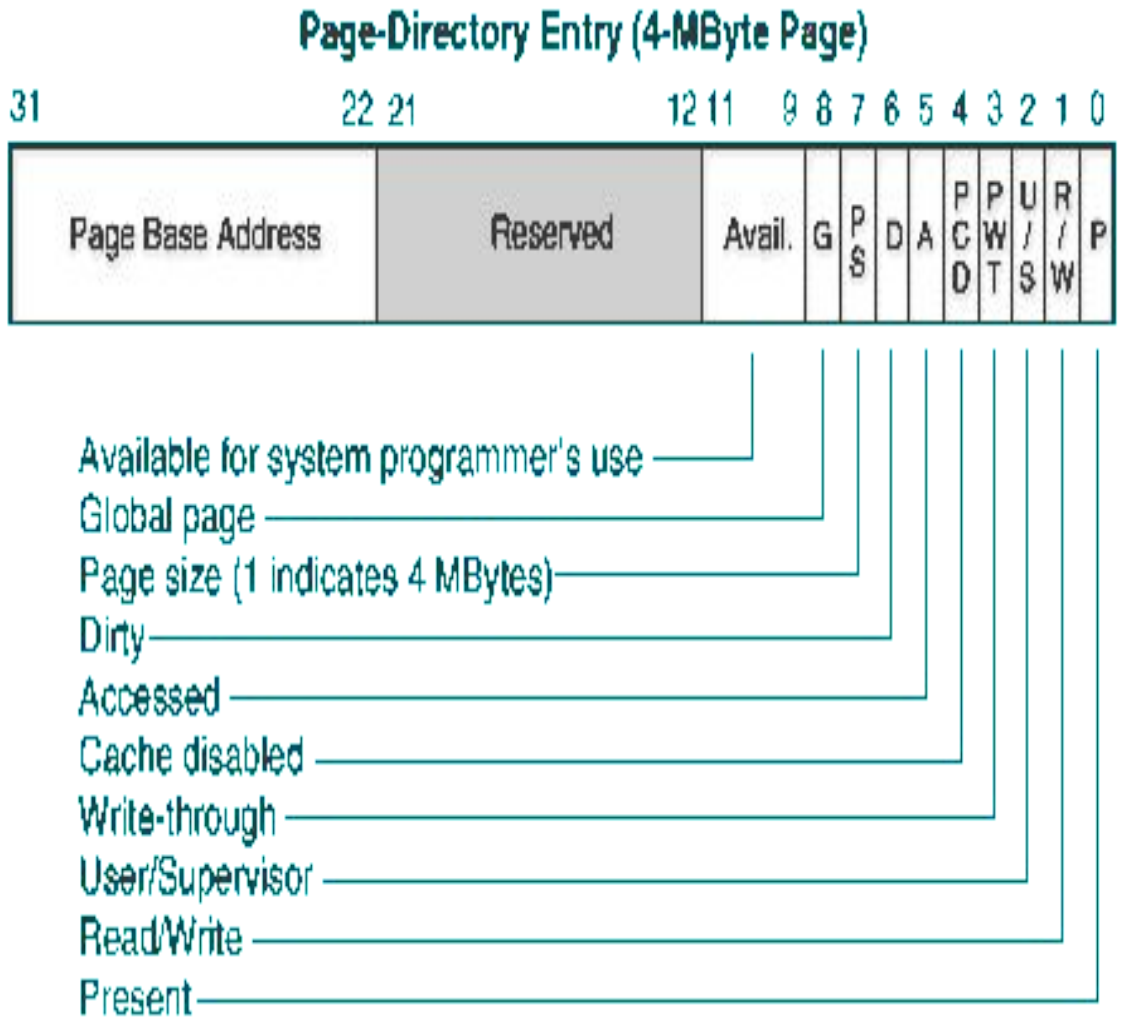
R/W (Read/Write) – PDE/PTE : тип доступа

P (Present) – PDE/PTE : бит присутствия

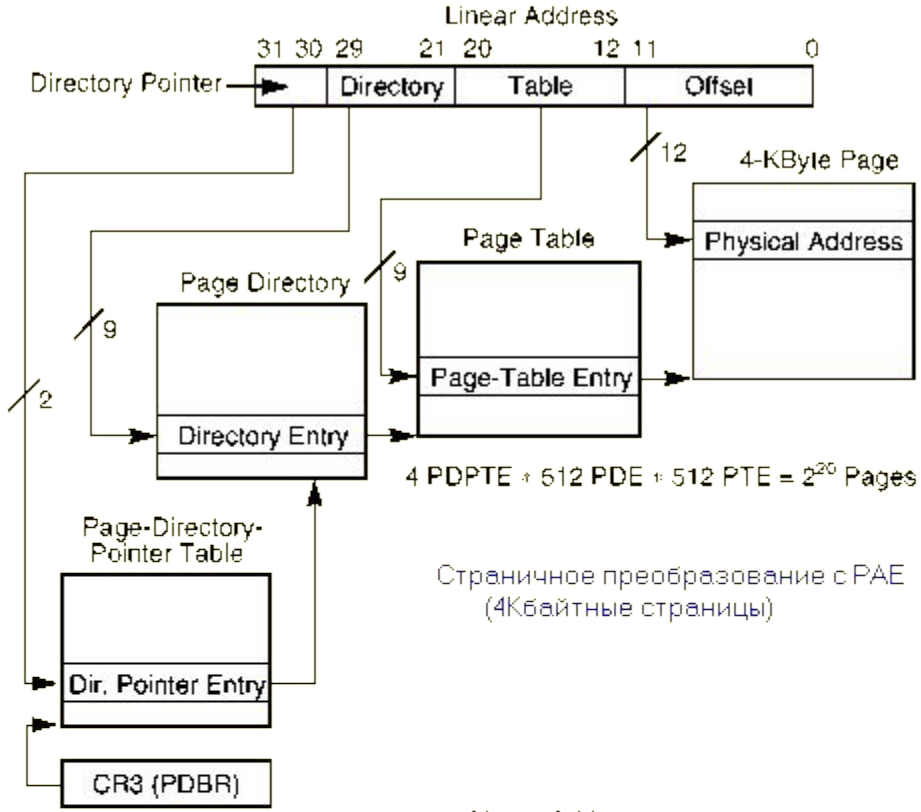
024 PTE = 2^{20} страниц

Для страниц размером 4Мбайт действует упрощенная (одноуровневая) схема формирования физического адреса. В этом случае физический адрес (старшие 10 бит) страницы хранится непосредственно в каталоге таблиц. Младшие 22 бита линейного адреса задают смещение от начала страницы. Конечно, страницы большого размера неудобны для подкачки при работе с маленькими приложениями, но тот факт, что при включенном PSE (или PAE) в системе можно использовать страницы обоих размеров позволяет повысить эффективность работы: на страницах большого размера можно разместить код операционной системы, к которому часто обращаются все приложения и который не следует выгружать из памяти, при этом экономится место - не нужны промежуточные таблицы страниц.

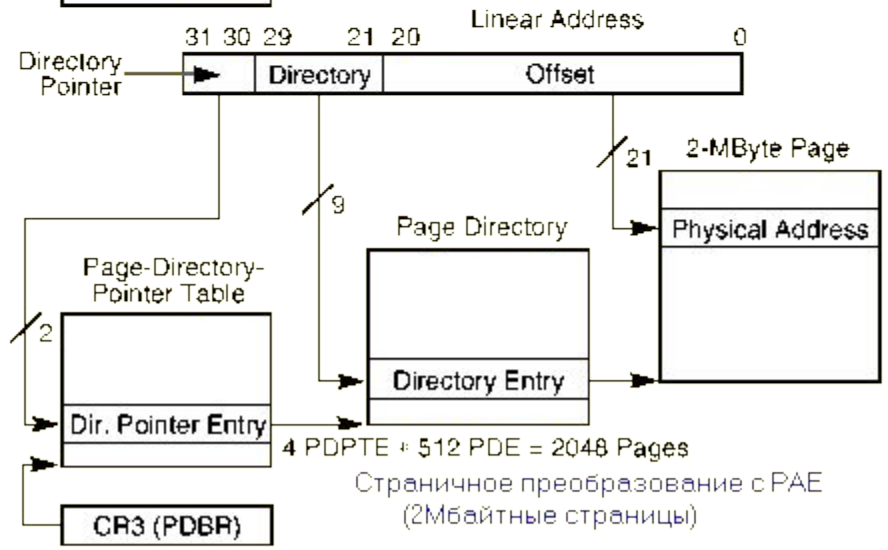

Linear Address
31
22 21
0
Directory
Offset
22
4-MByte Page
Physical Address
10
Page Directory
Directory Entry
32*
CR3 (PDBR)
1024 PDE = 1024 Pages


Page-Directory Entry (4-MByte Page)
31
22 21
12 11
9 8 7 6 5 4 3 2 1 0
Page Base Address
Reserved
Avail.
G
PS
D
A
PCD
PWT
U/S
R/W
P
Available for system programmer's use
Global page
Page size (1 indicates 4 MBytes)
Dirty
Accessed
Cache disabled
Write-through
User/Supervisor
Read/Write
Present

В процессорах с архитектурой P6 шина адреса 36-разрядная, однако для того, чтобы процессор генерировал 36-битные адреса необходимо включить PAE. Для того, чтобы хранить физический адрес большей разрядности размер элементов каталогов таблиц и таблиц страниц увеличен до 64 бит, а число элементов сокращено до 512 (чтобы каждая структура занимала ровно 4Кбайт - одну страницу памяти). Сокращение числа элементов привело к уменьшению размеров индексов, в результате высвободилось два старших бита в линейном адресе. Они используются для индексации еще одной структуры - *таблицы указателей на каталоги таблиц страниц (PDPT)*. Эта таблица содержит четыре 64-битных элемента, задающих физические адреса каталогов таблиц. Если при 32-битной адресации активен только один каталог таблиц, старшие 20 бит адреса которого хранятся в регистре CR3(PDBR), то при 36-битной адресации активных каталогов 4. Их адреса хранятся в PDPT, а регистр CR3 в этом случае хранит адрес PDPT (старшие 27 бит физического адреса в битах 5-31 регистра CR3) и называется Page-Directory-Pointer Table Register. Таким образом, формирование физического адреса для 4Кбайтных страниц трехуровневое, а для 2Мбайтных страниц - двухуровневое.

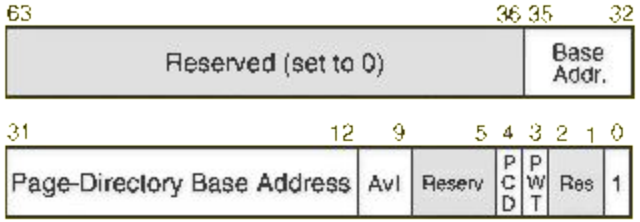


Страничное преобразование с PAE
(4Кбайтные страницы)

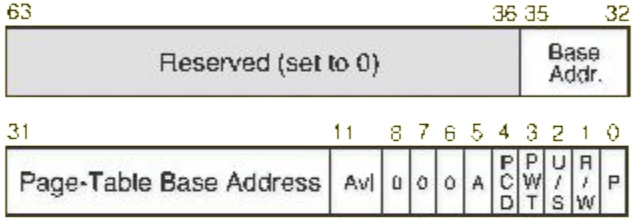


Страничное преобразование с PAE
(2Мбайтные страницы)

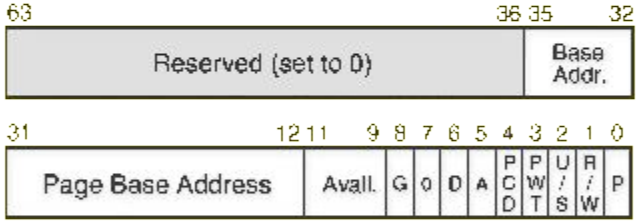
Page-Directory-Pointer-Table Entry



Page-Directory Entry (4-KByte Page Table)



Page-Table Entry (4-KByte Page)



Page-Directory Entry (2-MByte Page)