Эволюция x86. Защищенный режим. Intel 64.

Иван Викторович Михайлов

ИТМО, КТ

imihajlow@gmail.com

25.02.2015

8086 (1978 год)

- 16 бит данные;
- 20 бит адрес;
- Регистры AX, BX, CX, DX, SI, DI, BP, SP, CS, DS, ES, SS, FLAGS, IP.

```
address = 16 * segment + offset
```

Сегменты по умолчанию:

- CS:IP
- DS:BX
- DS:SI, DS:DI
- SS:BP

Прерывания

1. - Периферийное устройство устанавливает 1 на INTR и номер прерывания на шине данных (8 бит) или
   - Программа вызывает `INT n`.
2. Процессор помещает на стек FLAGS, CS и IP.
3. CS:IP $\leftarrow$ [n * 4].

`IRET` – возврат из прерывания;
`SEI/CLI` – разрешение/запрет прерываний.

Немаскируемое прерывание (NMI) не может быть запрещено.

80286 (1982 год)

Шина адреса 24 бита.
Защищенный режим.

80386 (1985 год)

32 бита.
Новые сегментные регистры: FS, GS.
Расширение старых регистров до 32 бит.
Защищенный режим.
Режим Virtual-8086.
System-management mode (SMM).
Специальные регистры:

- GDTR (Global Descriptor Table Register)
- IDTR (Interrupt Descriptor Table Register)
- LDTR (Local Descriptor Table Register)
- TR (Task Register)
- CR0-CR3 (Control Registers)
- DR0-DR7 (Debug Registers)

Pentium Pro (1995 год)

PAE (physical address extension).
PSE (page size extension).

Pentium II Xeon (1998 год)

PSE-36.

Защищенный режим

Защищенный режим

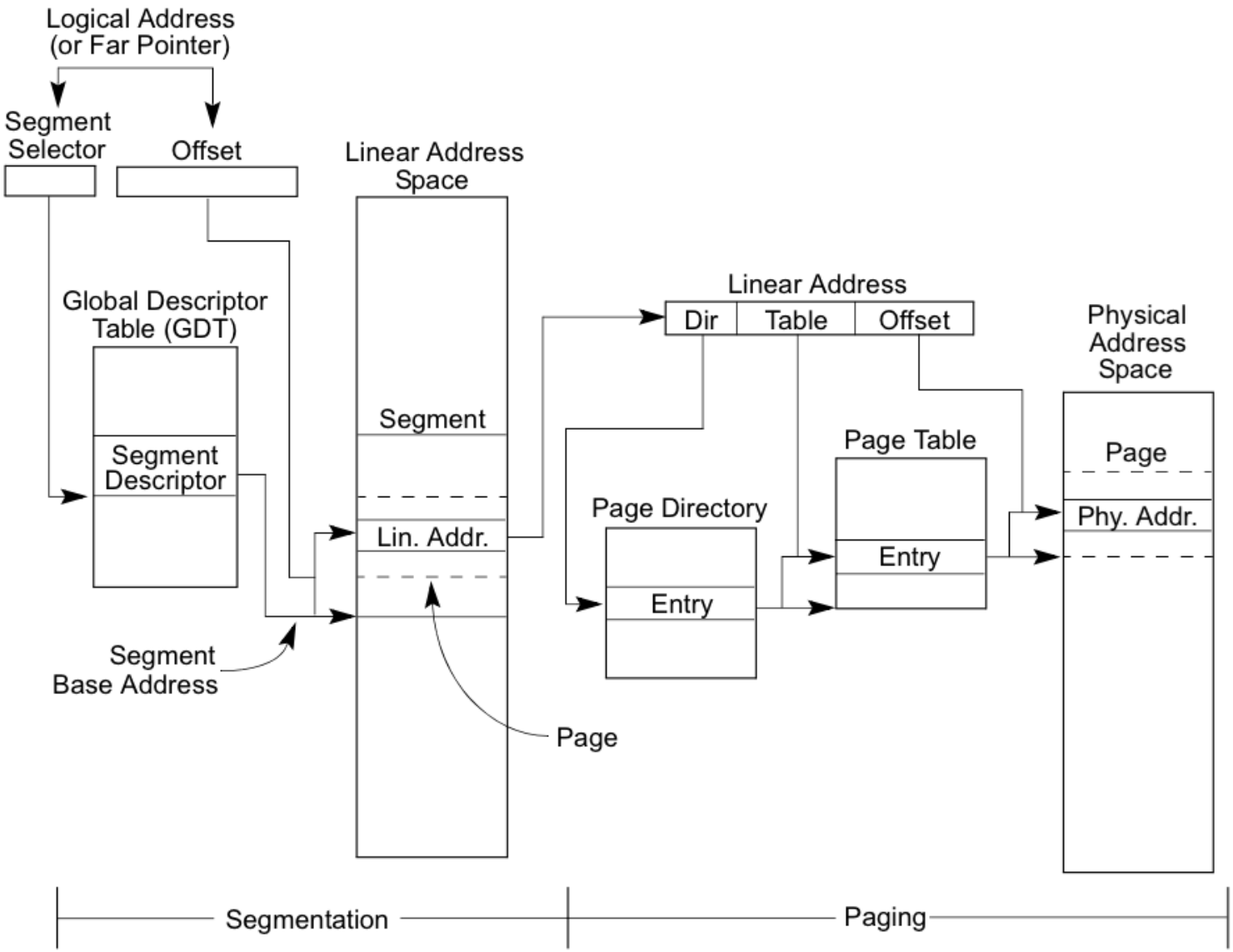
- Преобразование адреса;
- Защита;
- Управление задачами;
- Прерывания.

Преобразование адреса

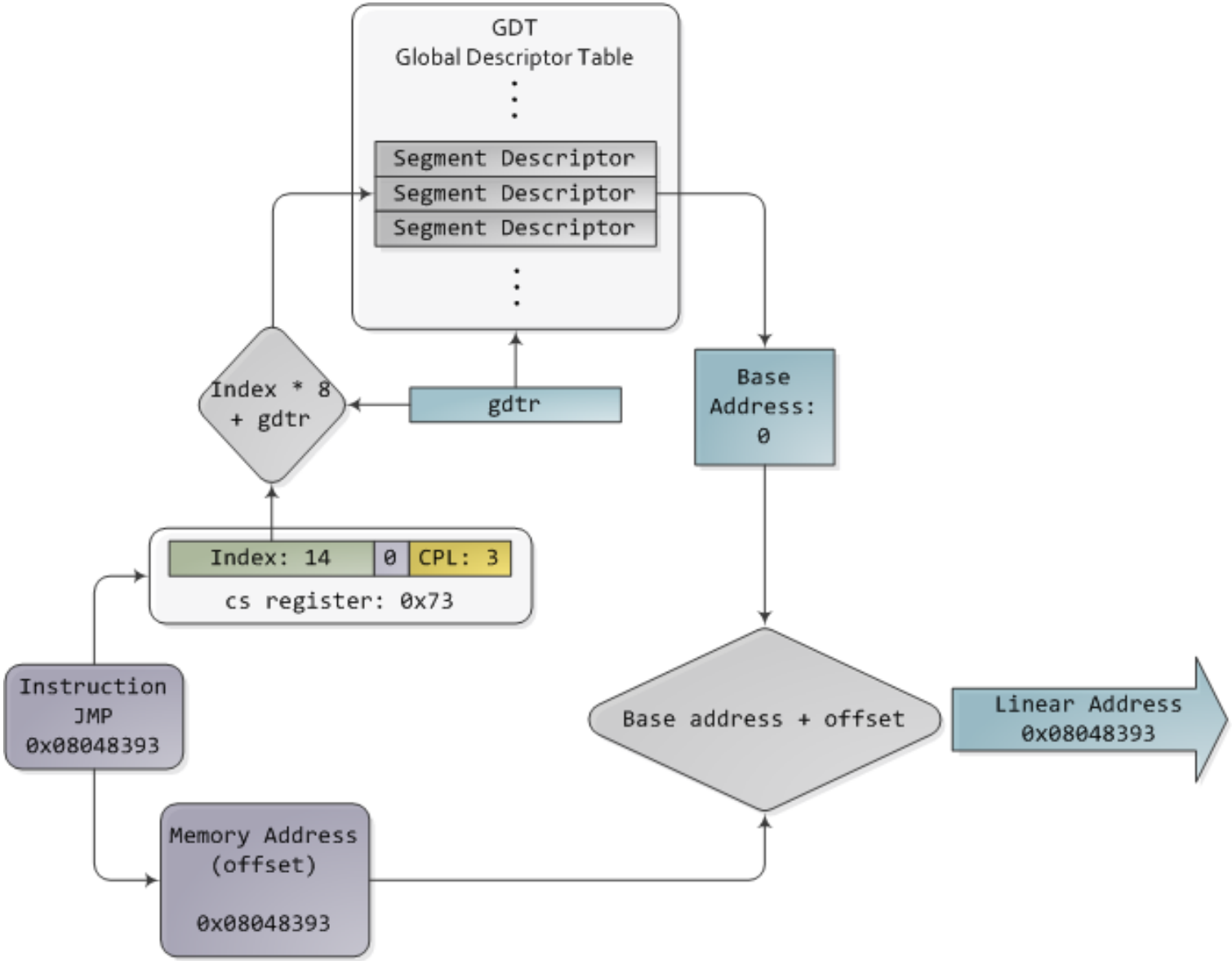
Преобразование адреса

Эффективный адрес ([ebp + 4 * esi + n]).
Логический адрес – селектор:эффективный.
↓ сегментация
Линейный адрес.
↓ страничное преобразование
Физический адрес.

Сегментация и страничное преобразование

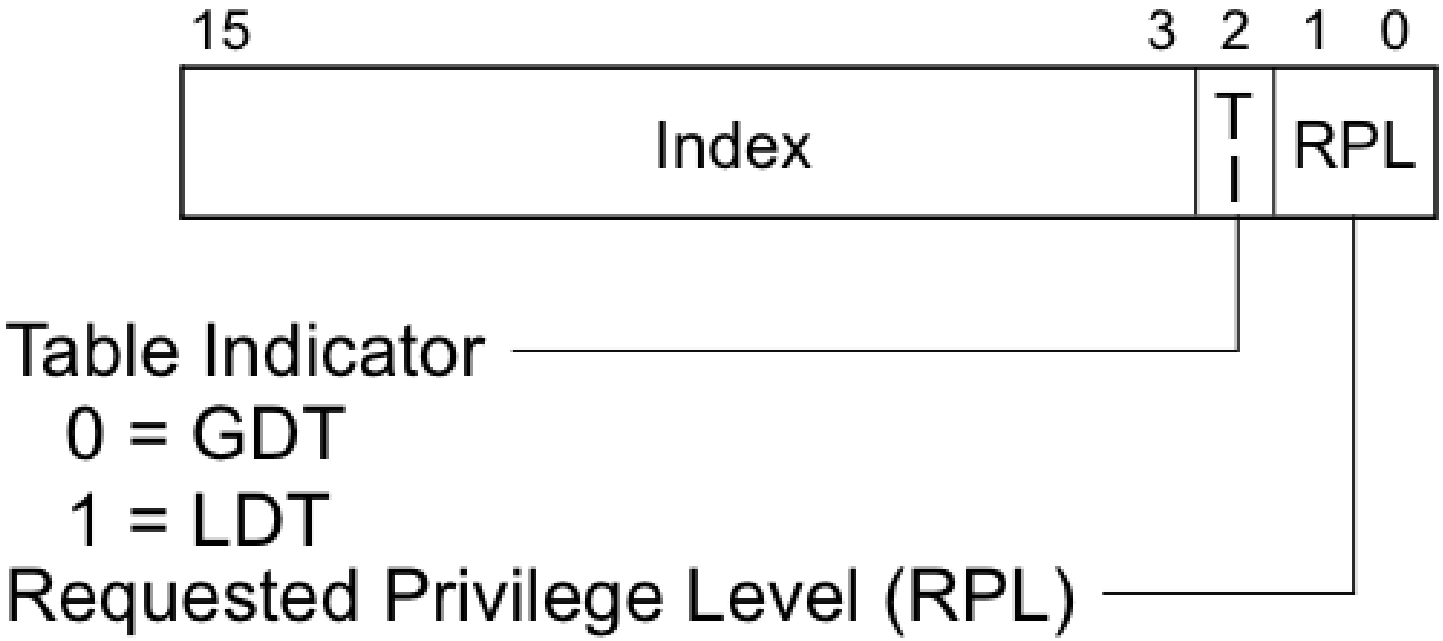


Сегментация

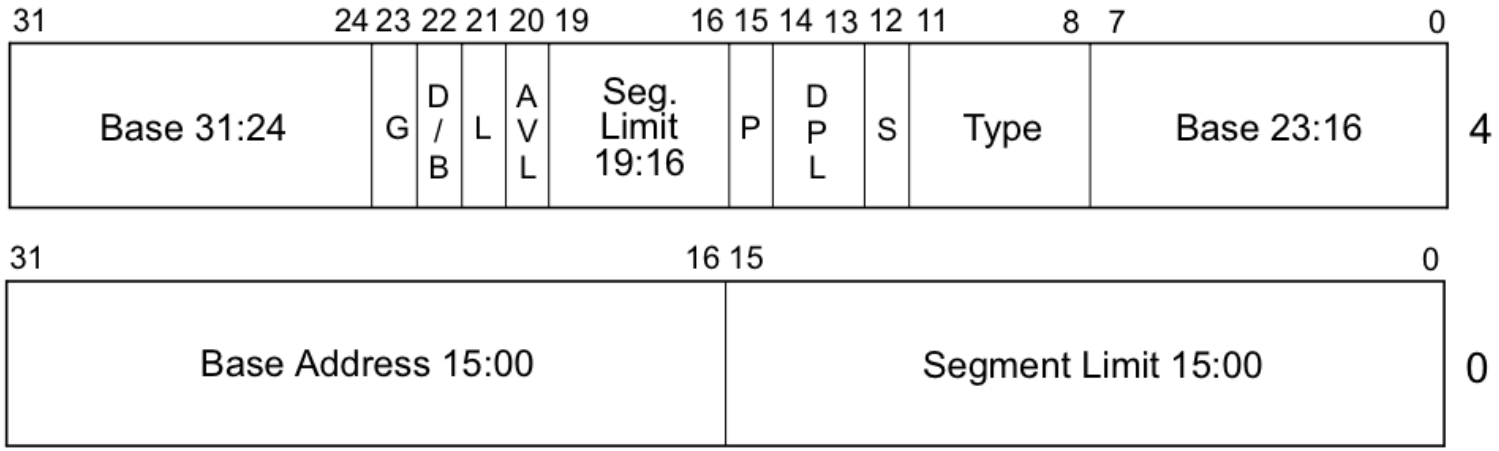


flat model

Селектор сегмента

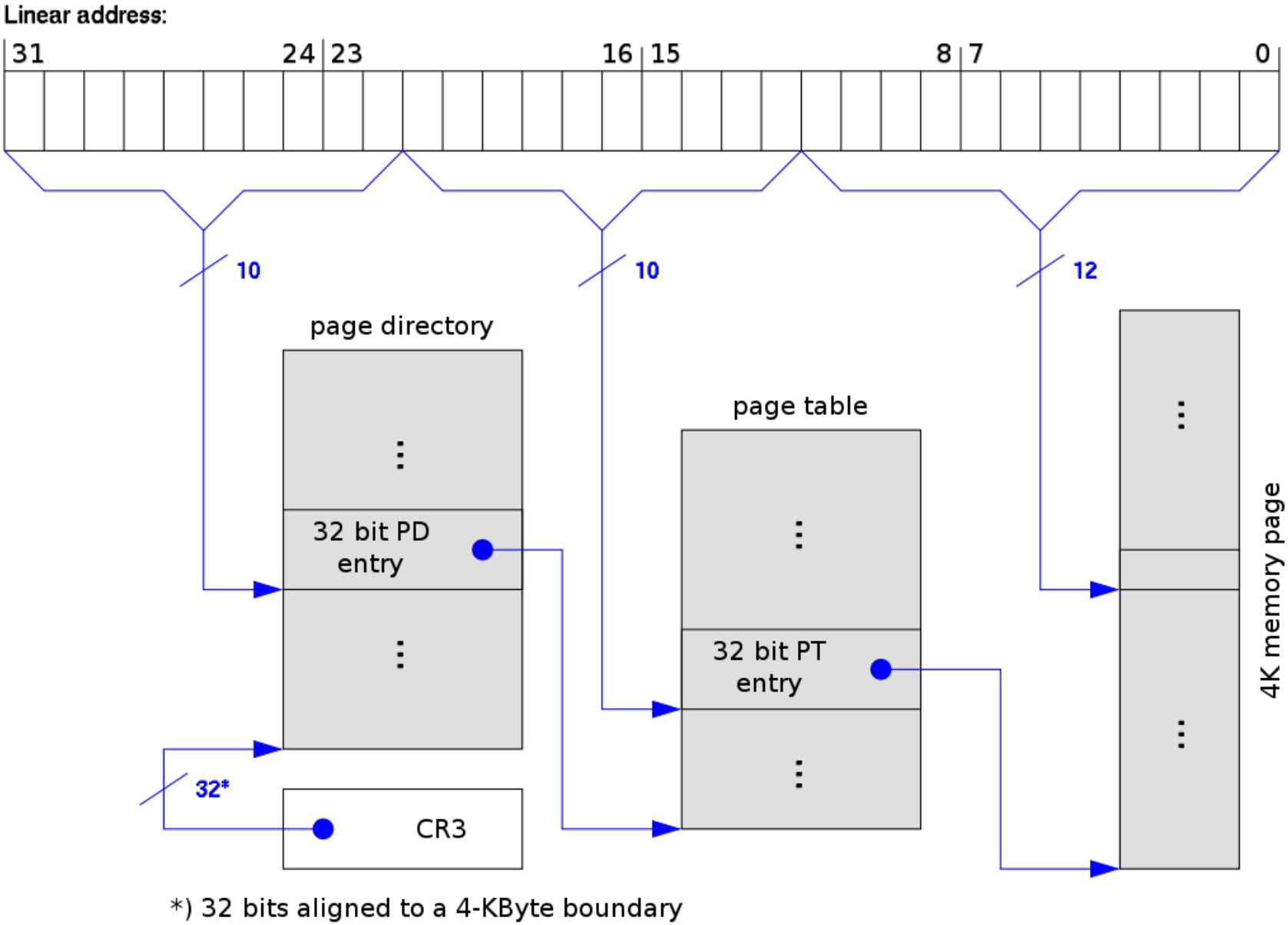


Дескриптор сегмента

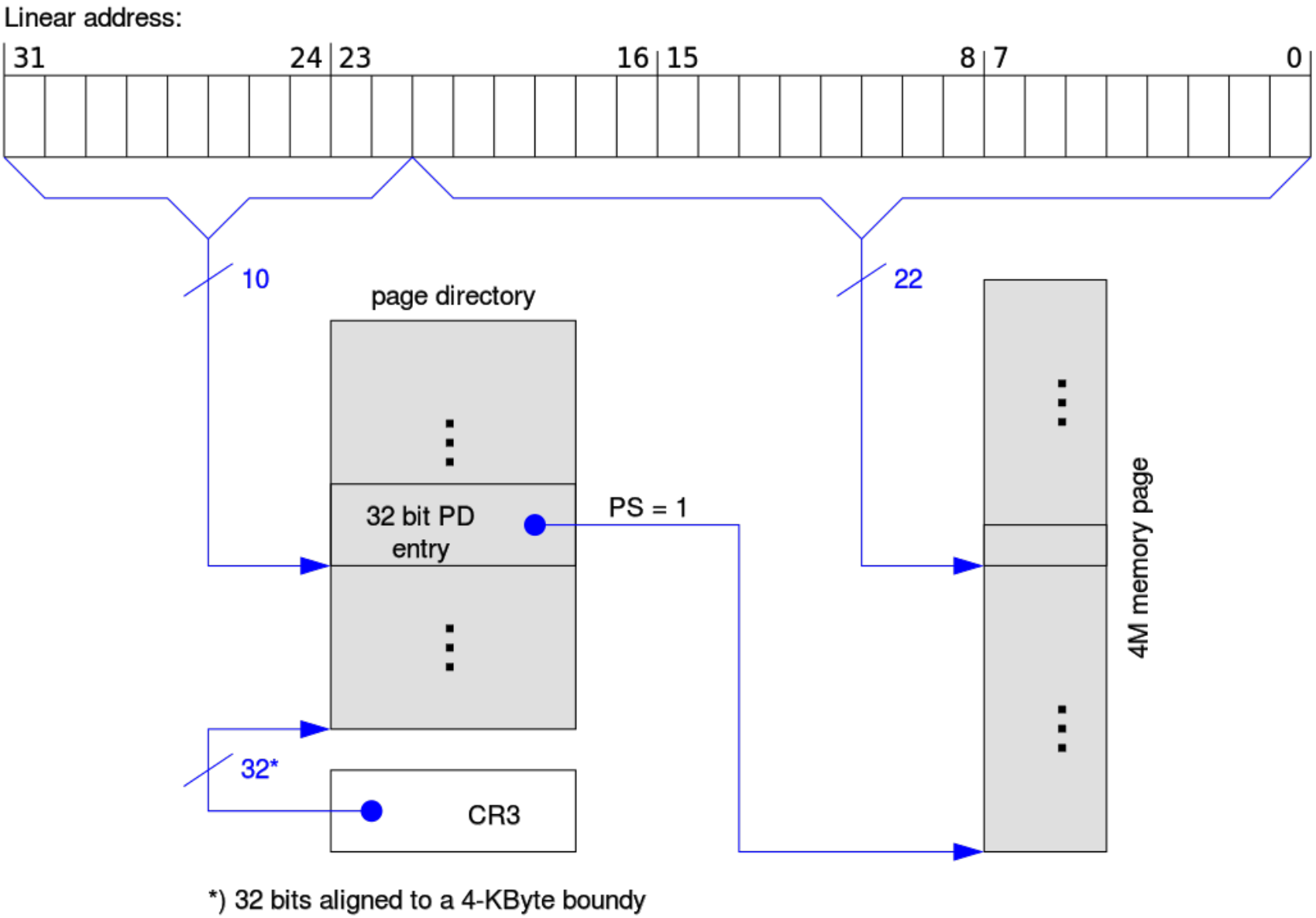


L — 64-bit code segment (IA-32e mode only)
AVL — Available for use by system software
BASE — Segment base address
D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
DPL — Descriptor privilege level
G — Granularity
LIMIT — Segment Limit
P — Segment present
S — Descriptor type (0 = system; 1 = code or data)
TYPE — Segment type

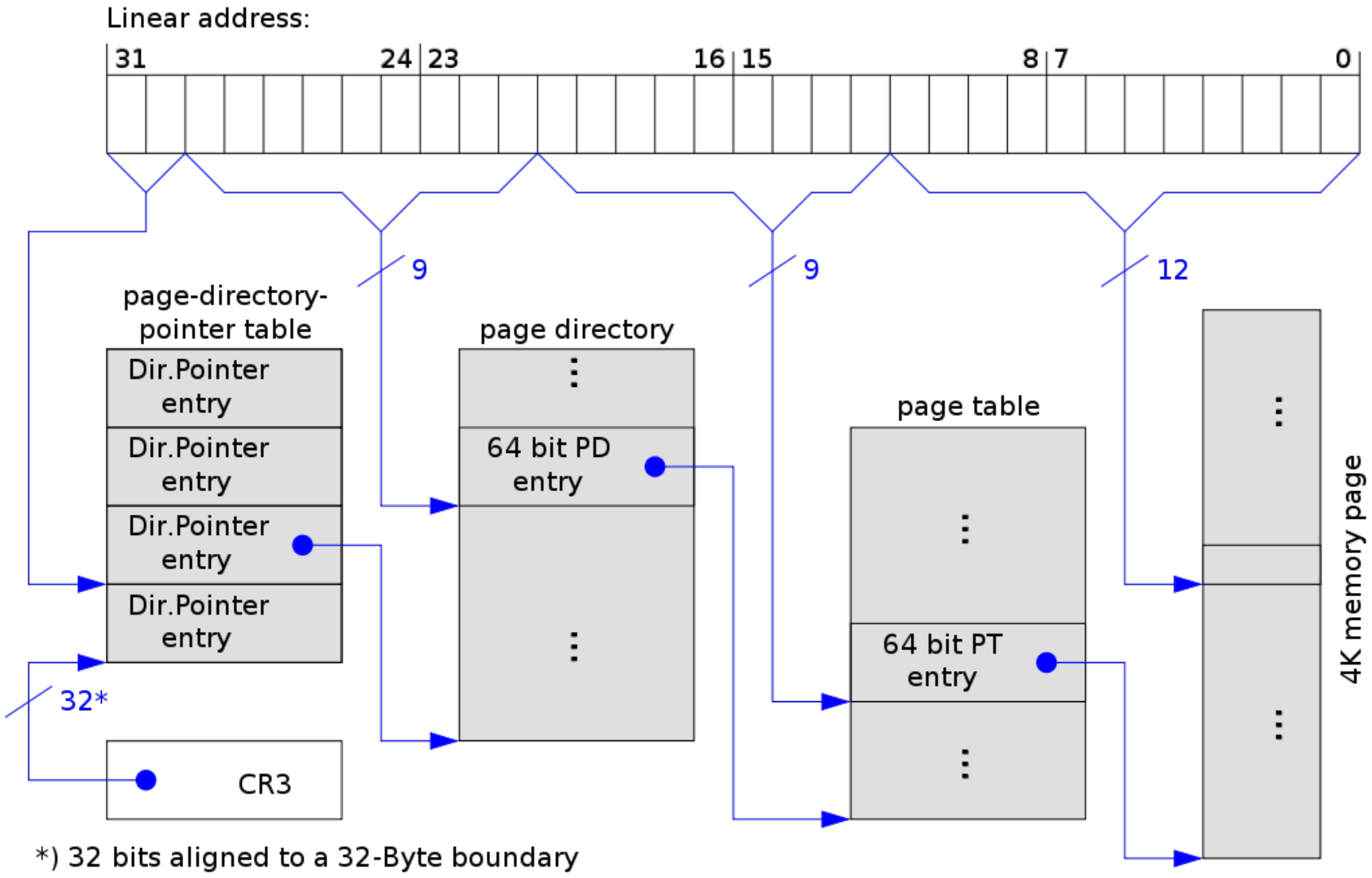
Страничное преобразование



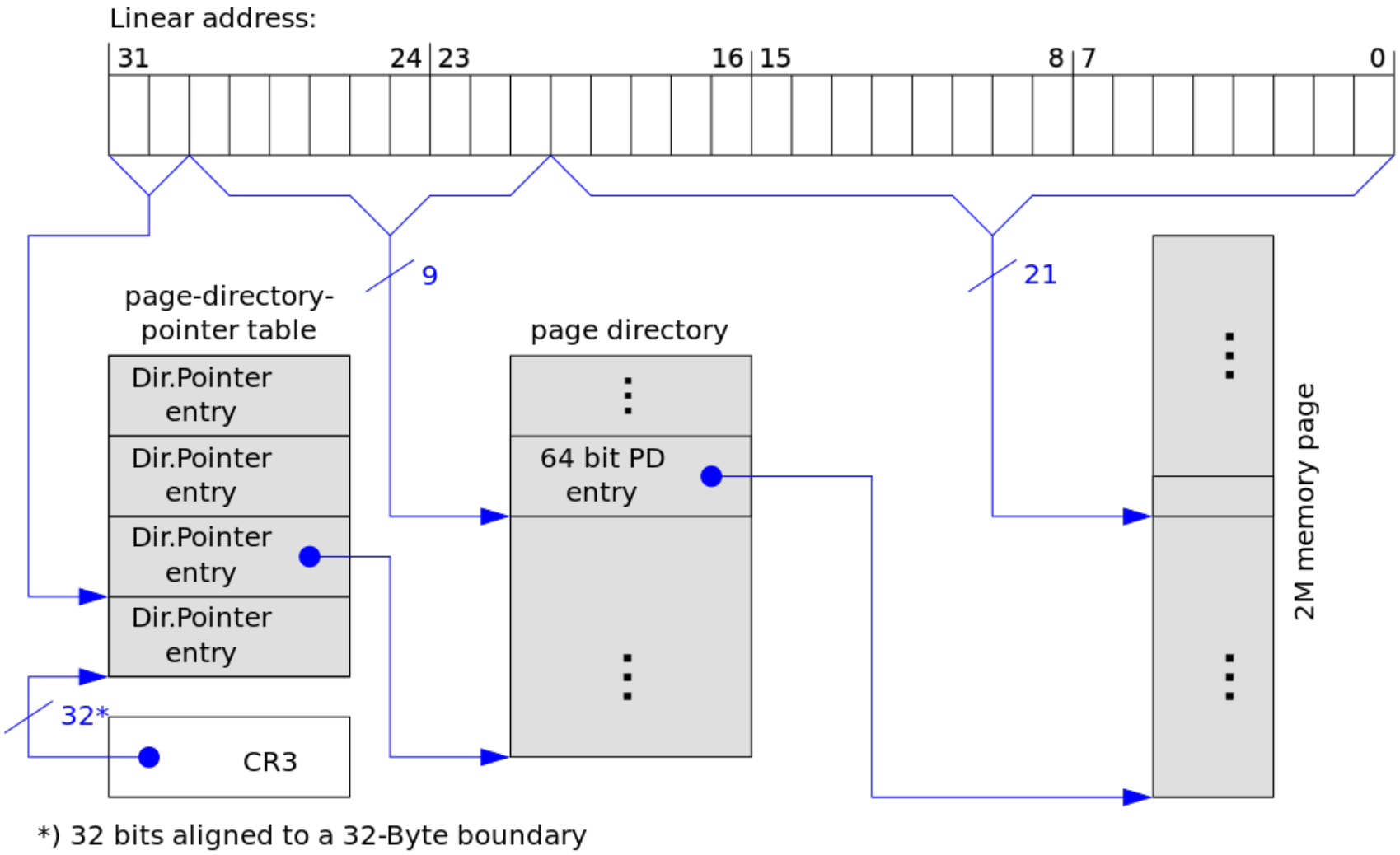
Страничное преобразование (PSE (4Мб))



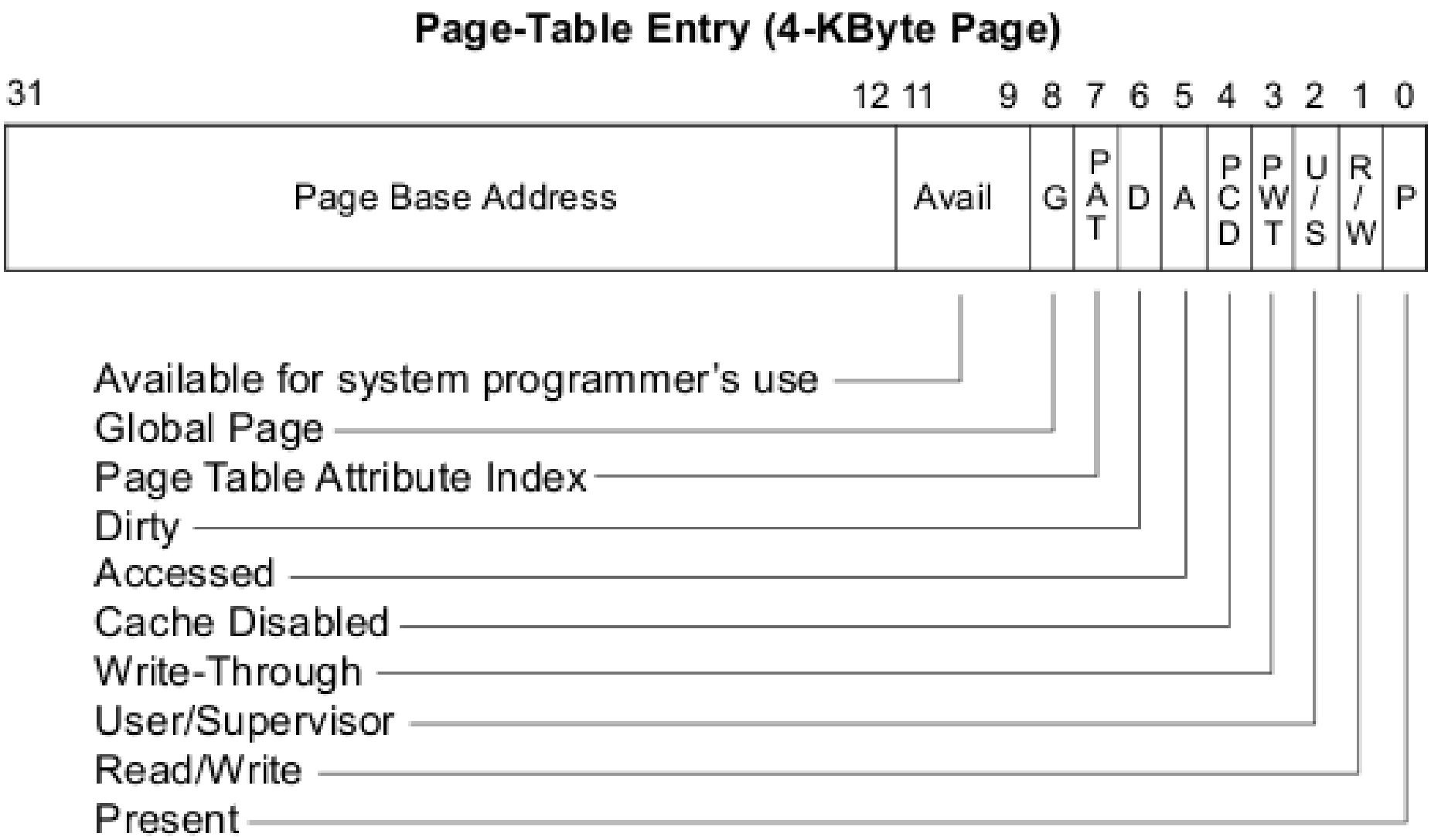
Страничное преобразование (PAE)



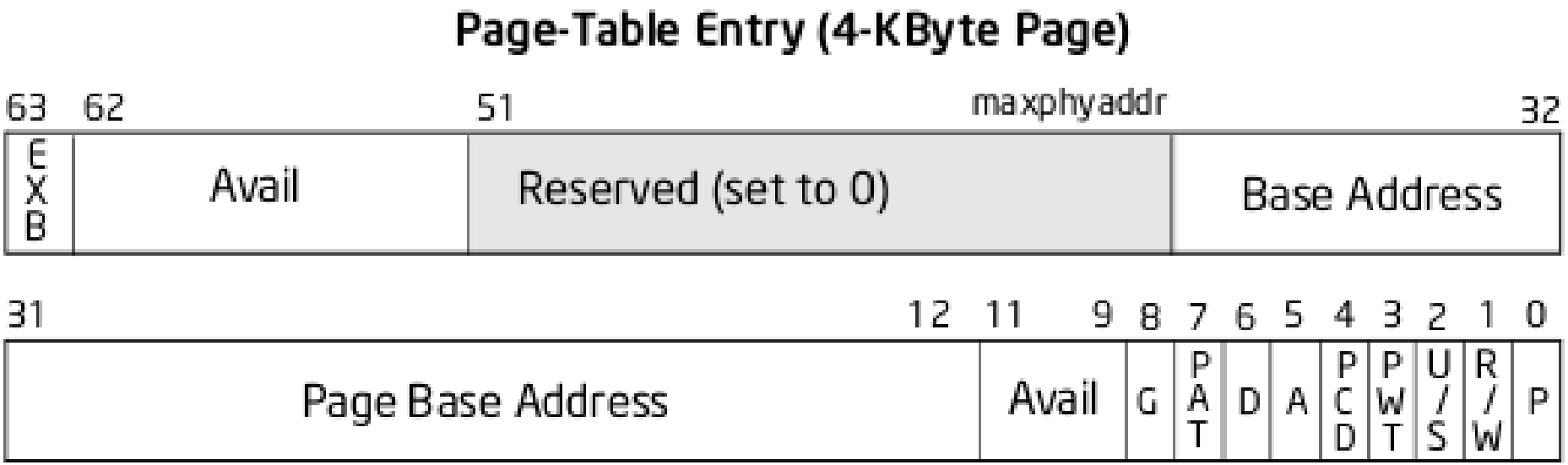
Страничное преобразование (PAE + PSE (2Мб))



Иерархические страничные структуры

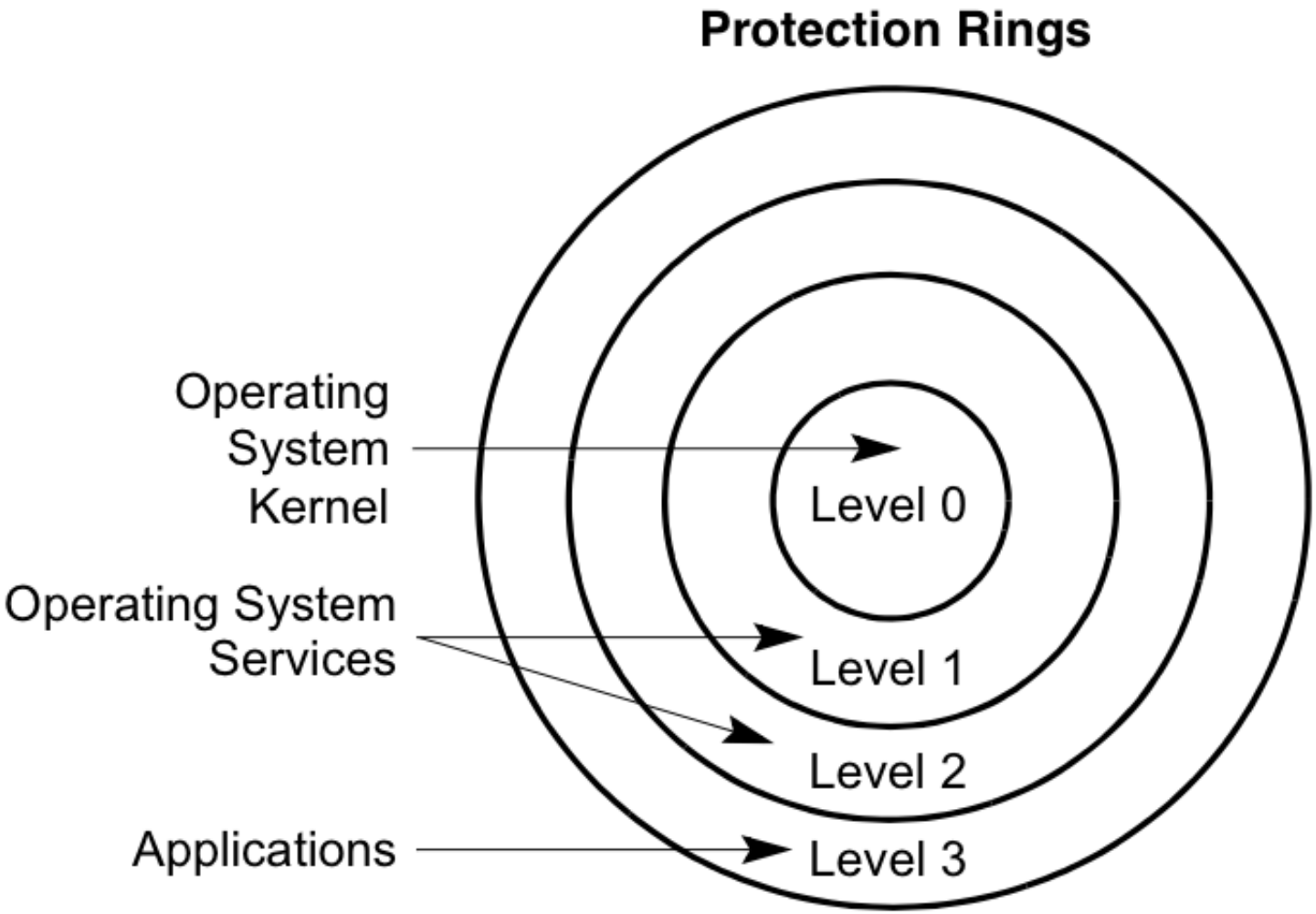


Иерархические страничные структуры (PAE)

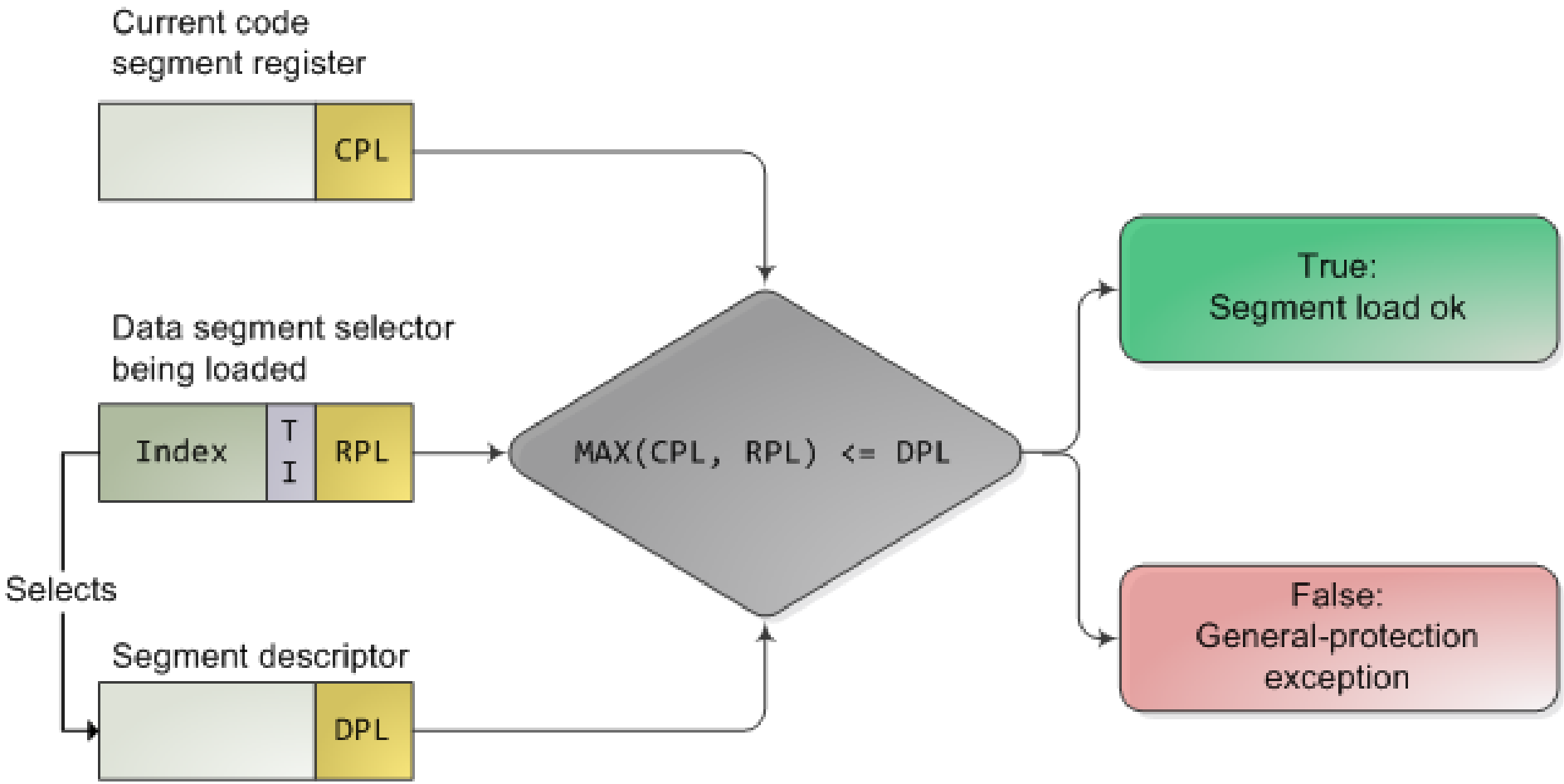


Защита

Уровни привелегий



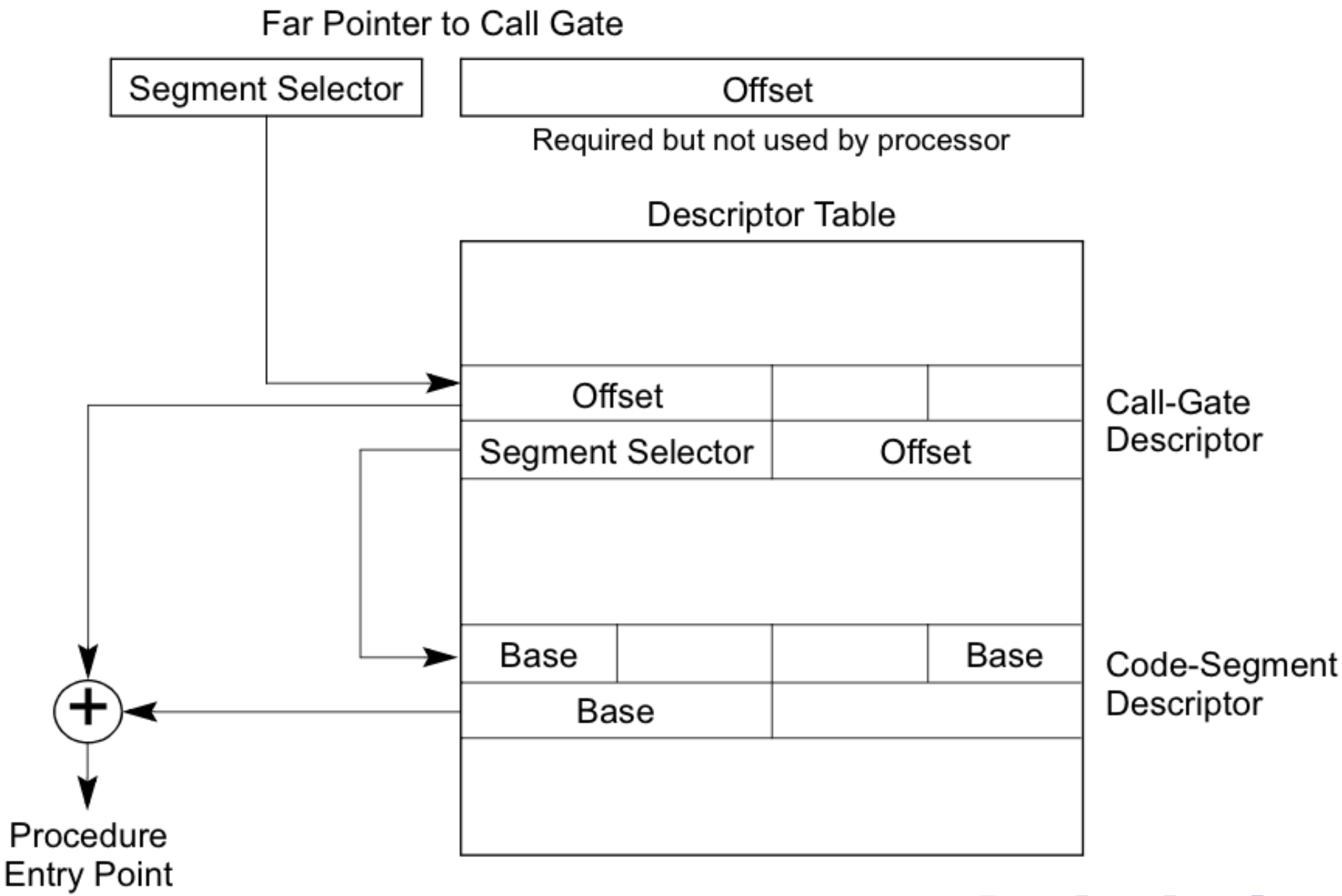
Проверка привелегий



Привилегии для страниц

- Supervisor flag – страница недоступна из кольца 3.
- Read/write flag.

Call gate

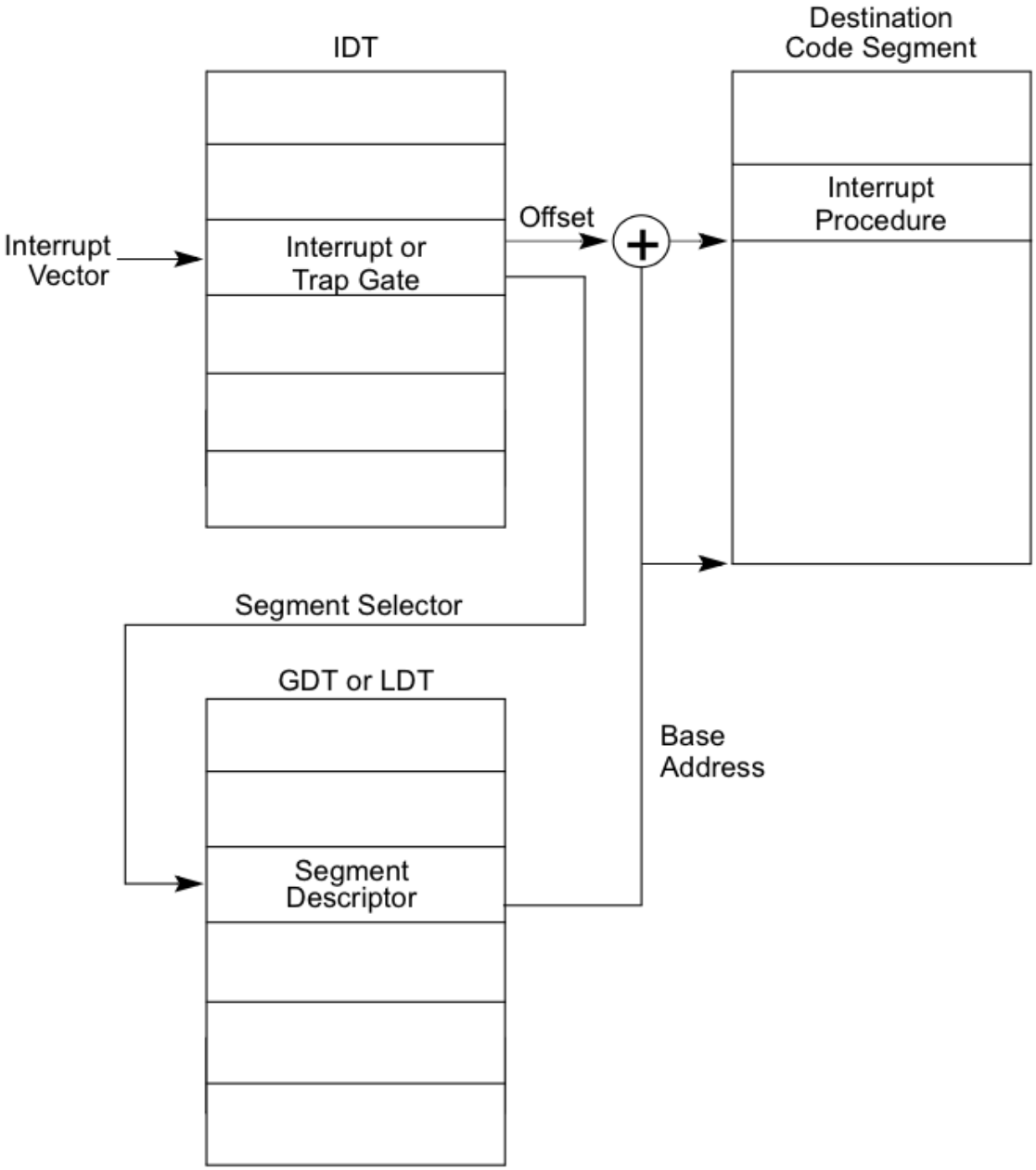

Far Pointer to Call Gate
Segment Selector
Offset
Required but not used by processor
Descriptor Table
Offset
Segment Selector
Offset
Call-Gate Descriptor
Base
Base
Base
Code-Segment Descriptor
Procedure Entry Point

Прерывания и исключения

Прерывания и исключения

256 прерываний.
Вектора 0...31 – внутренние исключения и прерывания процессора.
Вектора 32...255 – внешние прерывания или вызываются через `INT n`.

Interrupt gate/trap gate



SYSCALL

SYSENTER/SYSEXIT (Intel)
SYSCALL/SYSRET (AMD)

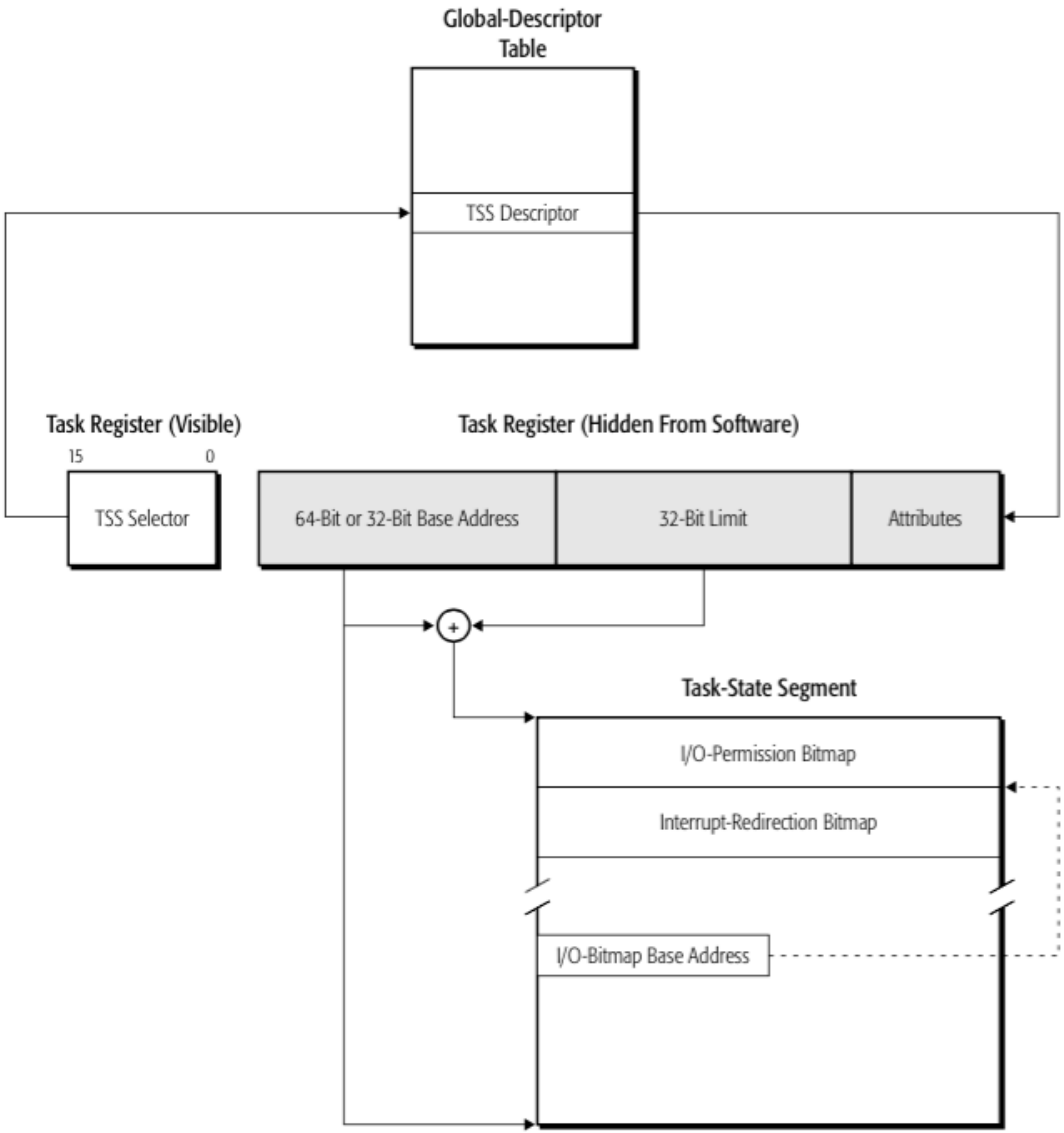
Управление задачами

Управление задачами

Задача – программа, которую можно приостановить и продолжить с той же точки. Состояние задачи:

- Регистры;
- Сегменты;
- LDT;
- TR;
- CR3 (указатель на страничные структуры);
- Маски доступа к прерываниям и портам;
- TSS предыдущей задачи.

TSS


Global-Descriptor Table
TSS Descriptor
Task Register (Visible)
15
0
TSS Selector
Task Register (Hidden From Software)
64-Bit or 32-Bit Base Address
32-Bit Limit
Attributes
+
Task-State Segment
I/O-Permission Bitmap
Interrupt-Redirection Bitmap
I/O-Bitmap Base Address

SMM

SMM

- Вход – по #SMI;
- Выход – инструкция RSM.

Intel 64

AMD64, x86-64, x64, IA-32e

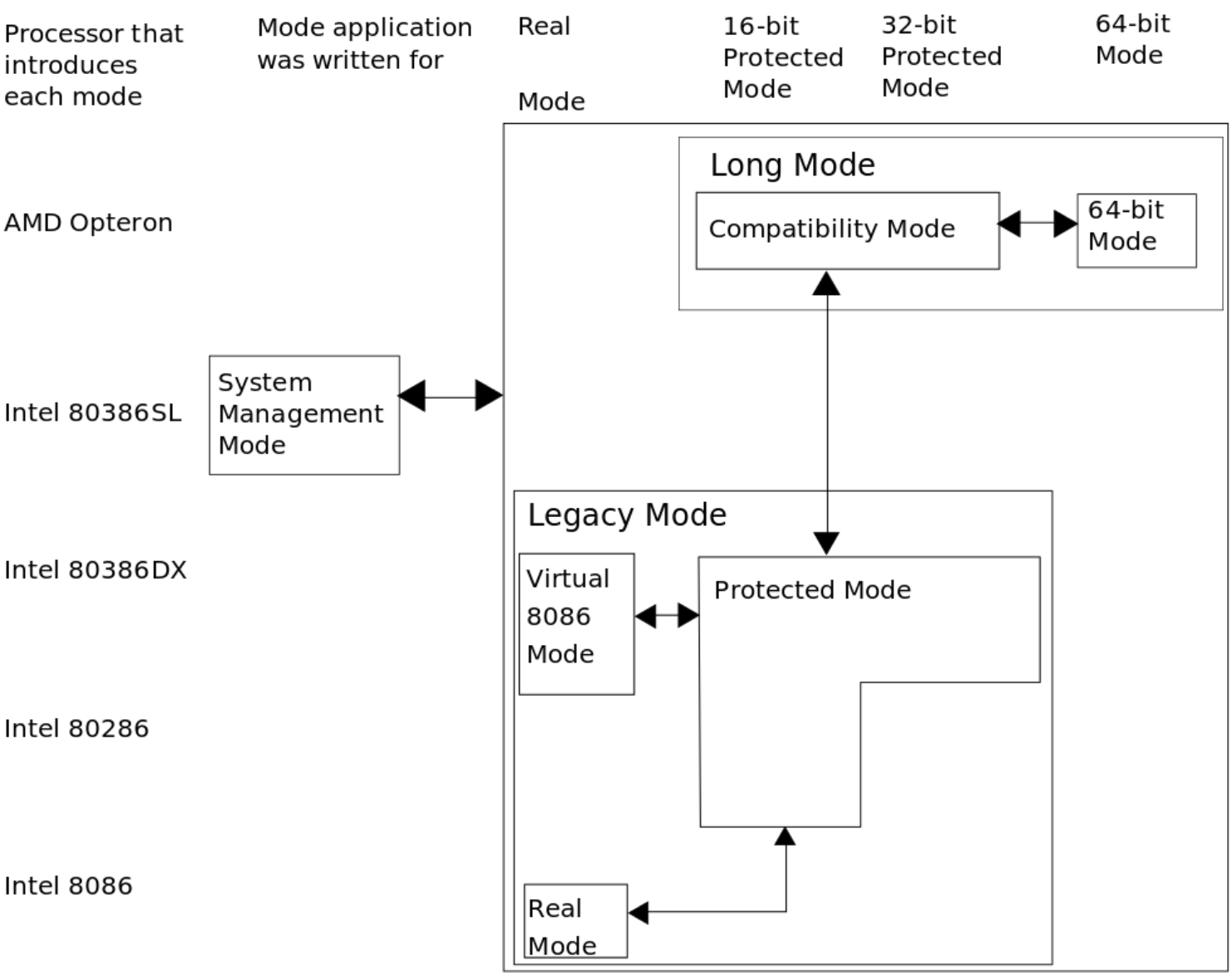
AMD K8, 2003 год

Режимы работы

Новый режим – IA-32e (Long Mode). Подрежимы:

- 64-битный режим;
- Режим совместимости (с защищенным режимом).

Режимы работы



Расширение регистров

- Префикс R (EAX $\leftarrow$ RAX) – 64-битные регистры;
- Новые регистры общего назначения (R8-R15);
- Новые XMM-регистры (XMM8-XMM15).

Изменения инструкций

Удалены

- PUSHA/POPA;
- Двоично-десятичная арифметика;
- И другие.

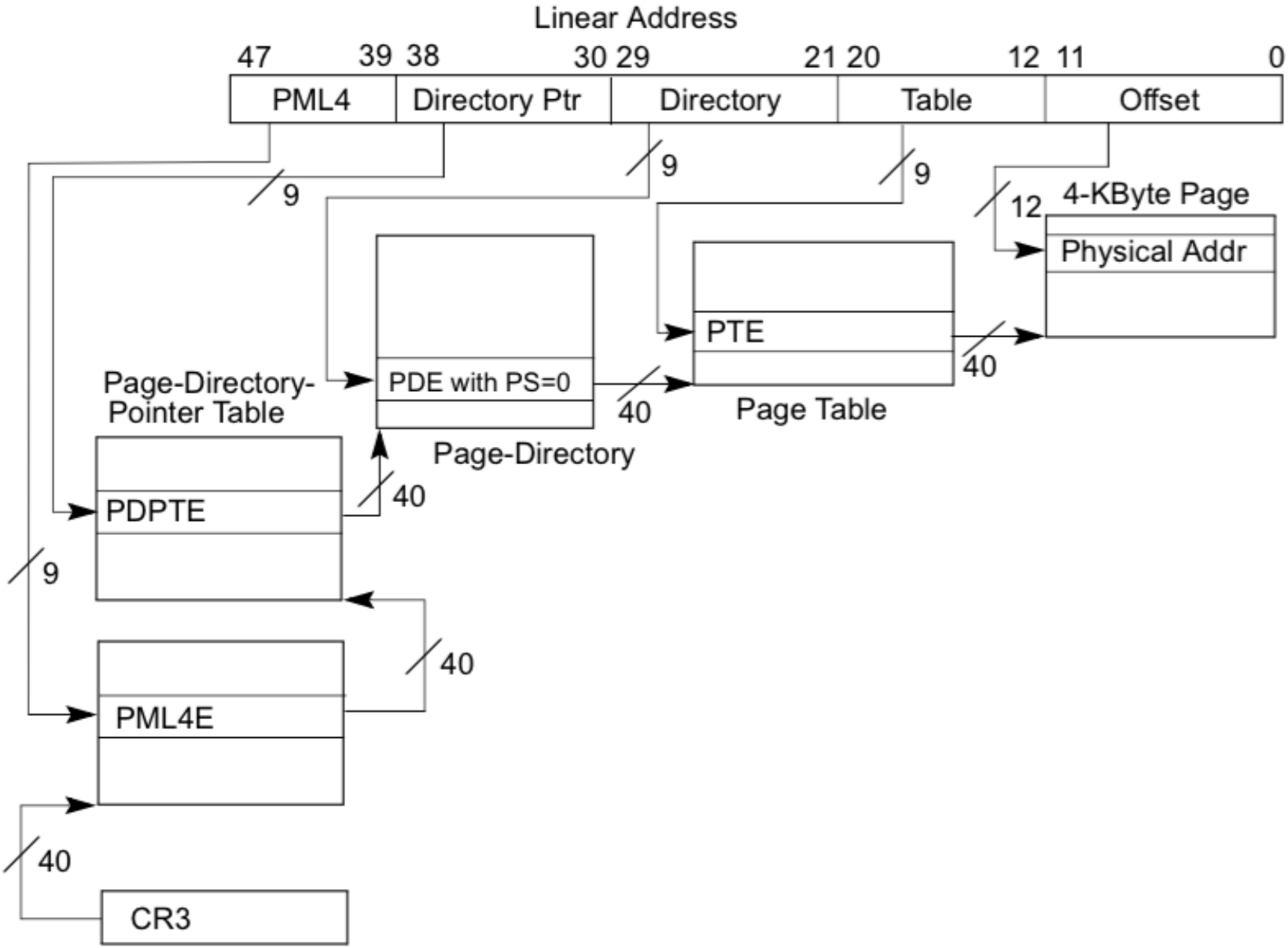
Не поддерживаются

- Virtual-8086;
- Аппаратное управление задачами.

Сегментация в 64-битном режиме

- База и лимит игнорируются;
- Новый флаг L для сегмента кода;
- FS и GS содержат базы для системных структур.

Страничное преобразование



Адресация на базе RIP

NASM:

```
mov [LABEL wrt rip], rax
```

YASM:

```
mov [rel LABEL], rax ; Адрес относительно RIP
mov [abs LABEL], rax ; Абсолютный адрес
```

Директива `default`:

```
default rel
mov [LABEL], rax ; Адрес относительно RIP
default abs
mov [LABEL], rax ; Абсолютный адрес
```

Конвенция вызова

System V AMD64 ABI – Linux, BSD, OS X.

- Стэк перед вызовом выравнен на 16 байт.
- „Красная зона” (red zone) – 128 байт стэка от [rsp-128] до [rsp-8].
- Целые числа: RDI, RSI, RDX, RCX, R8, R9.
- Вещественные числа: XMM0-XMM7.
- Максимум 14 параметров в регистрах.
- Вызываемый сохраняет RBP, RBX, R12-R15.
- Результат в RAX (целый) или в XMM0 (вещественный).

Конвенция вызова

Microsoft x64

- Стэк перед вызовом выравнен на 16 байт.
- Shadow space – пустые 32 байта на стэке ([rsp+8] - [rsp+32]).
- Параметры в регистрах: RCX/XMM0, RDX/XMM1, R8/XMM2, R9/XMM3 (всего 4 параметра).
- Вызываемый сохраняет RBX, RBP, RDI, RSI, R12-R15.
- Результат в RAX (целый) или в XMM0 (вещественный).

Что почитать

- Intel® 64 and IA-32 Architectures Software Developer's Manual, Vol. 3, Chapters 2-7
- AMD64 Architecture Programmer's Manual Volume 2: System Programming,
- http://duartes.org/gustavo/blog/post/cpu-rings-privilege-and-protection/
- http://duartes.org/gustavo/blog/post/memory-translation-and-segmentation/
- Calling conventions for different C++ compilers and operating systems by Agner Fog. Technical University of Denmark.
- http://habrahabr.ru/company/intel/blog/238091/
- http://prodebug.sourceforge.net/pmtut.html

Конец.