Операційні системи

Лекція 8

Керування оперативною пам'яттю у процесорах архітектури x86

План лекції

- Системні таблиці і регістри системних адрес
- Селектор і дескриптор сегмента
- Захист сегментів
- Завантаження селектора у сегментний регістр
- Звернення до пам’яті
- Сторінковий механізм керування пам'яттю

Керування пам'яттю у процесорах архітектури x86

- У захищеному режимі підтримується сегментний і сегментно-сторінковий розподіл пам'яті
- Сторінкове перетворення включається або відключається окремим прапорцем у регістрі керування процесором **cr0**
- Максимальний розмір сегмента – 4 ГБ
- У 16-розрядні сегментні регістри процесора завантажуються *селектори*, які вказують на *дескриптори* сегментів
  - **cs** – сегмент коду
  - **ss** – сегмент стека
  - **ds**, **es**, **fs**, **gs** – сегменти даних
- Дескриптори містяться у спеціальних системних таблицях, на які вказують *регістри системних адрес*
- У дескрипторах містяться:
  - базова адреса сегмента
  - межа (розмір) сегмента
  - правила доступу до сегмента

Системні таблиці і регістри системних адрес

- *Регістри системних адрес* містять покажчики на системні таблиці, призначені для керування пам'яттю та диспетчеризації процесів
- Доступ до сегментів у пам'яті здійснюється через *дескриптори*
- Дескриптори містяться у двох таблицях, доступних процесу
  - *Глобальна таблиця дескрипторів* (*Global Descriptor Table*, GDT)
    - Містить дескриптори, що описують програмний код і дані, спільні для усіх процесів (наприклад, бібліотеки, драйвери пристроїв, тощо), а також численні системні об'єкти
    - На цю таблицю вказує регістр **gdtr**
  - *Локальна таблиця дескрипторів* (*Local Descriptor Table*, LDT)
    - Доступна лише тому процесу, який виконується в даний момент
    - Кожний процес має свою власну локальну таблицю
    - На цю таблицю вказує регістр **ldtr**

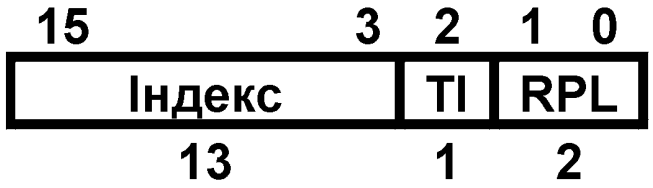
Системні таблиці і регістри системних адрес

- Обробники переривань доступні через *таблицю дескрипторів переривань* (*Interrupt Descriptor Table*, IDT)
  - На неї вказує регістр **idtr**
- *Контекст процесу* знаходиться у спеціальному системному об'єкті, що називається *сегментом стану задачі* (*Task Status Segment*, TSS)
  - Цей об'єкт описується дескриптором, на який вказує регістр **ts**
- Сегменти, що вказують на глобальні системні об'єкти (**gdtr** та **idtr**) містять базові лінійні адреси цих об'єктів, а також задають розмір об'єктів
- Сегменти, що вказують на локальні для кожного процесу об'єкти (**ldtr** та **tr**) містять лише селектори, які адресують відповідні дескриптори у глобальній таблиці
  - Таким чином, таблиця GDT містить дескриптори, що описують сегменти стану задач і локальні таблиці дескрипторів усіх процесів, що виконуються в системі
  - Для переходу до виконання іншого процесу необхідно просто завантажити у регістри **ldtr** та **tr** відповідні дескриптори

Системні таблиці і регістри системних адрес

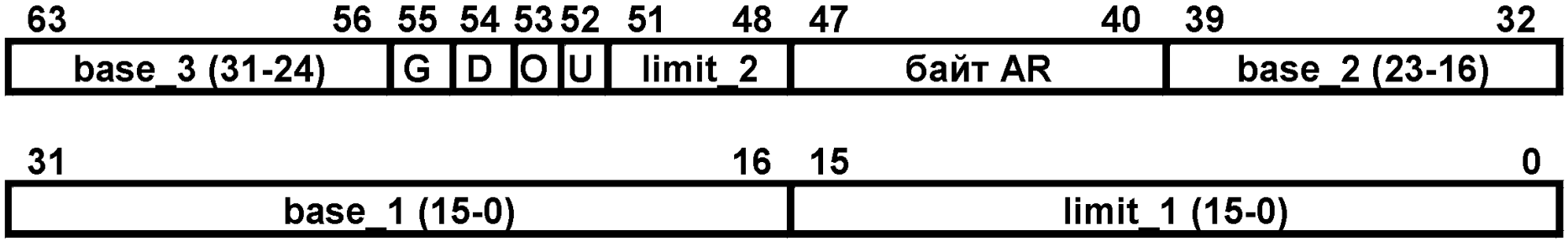
Регістр	Таблиця (об'єкт)		Формат регістру	
gdtr	GDT	Global Descriptor Table	48	Містить базову адресу і межу таблиці
ldtr	LDT	Local Descriptor Table	16	Селектор у таблиці GDT
idtr	IDT	Interrupt Descriptor Table	48	Містить базову адресу і межу таблиці
tr	TSS	Task Status Segment	16	Селектор у таблиці GDT

Селектор сегмента



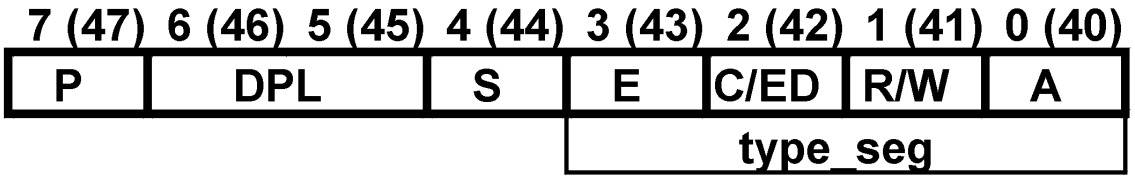
- *Селектор* – це 16-розрядна структура, яка завантажується в сегментні регістри
- Селектор адресує не сам сегмент, а його дескриптор
- 13 старших розрядів селектора є індексом в таблиці дескрипторів
  - Таким чином, кожна таблиця може містити 2^{13}=8192 дескрипторів
- Один розряд селектора (біт 2), який позначається як прапорець **TI**, вказує в якій з таблиць знаходиться дескриптор
  - **TI==0 □ GDT**
  - **TI==1 □ LDT**
- Останні 2 розряди селектора (біти 1,0) відведені для задавання рівня привілеїв (*Requested Privilege Level*, RPL), який використовується механізмом захисту

Дескриптор сегмента



- Дескриптор сегмента є 8-байтовою структурою
- Головні поля дескриптора:
  - 32-розрядна *базова адреса* (*base*)
  - 20-розрядна *межа сегмента* (*limit*) – визначає розмір сегмента в залежності від *прапорця гранулярності* **G**:
    - **G==0** → розмір у байтах (максимальний розмір сегмента 1 МБ)
    - **G==1** → розмір у 4-кБ сторінках (максимальний розмір сегмента – 4 ГБ)
  - *Байт захисту* (**AR**)
  - *Прапорець розрядності* (**D**):
    - **D==0** → 16-розрядні операнди і режими 16-розрядної адресації
    - **D==1** → 32-розрядні операнди і режими 32-розрядної адресації

Байт захисту



- Розряд 7 (**P** – *Present*) показує наявність сегмента у пам'яті
- Розряди 5 і 6 (**DPL** – *Descriptor Privilege Level*) визначають рівень привілеїв дескриптора
- Розряд 4 (**S** – *System / Segment*) визначає, чи є об’єкт, який описує цей дескриптор, сегментом у пам’яті чи спеціальним системним об’єктом
- Розряди 1 – 3 визначають тип сегмента і права доступу до нього
  - Розряд 3 (**E** – *Executable*)
  - Розряд 2
    - Для сегментів коду – біт підпорядкованості (**C** – *Conforming*)
    - Для сегментів даних – біт розширення вниз (**ED** – *Expand Down*)
  - Розряд 1
    - Для сегментів коду – дозвіл на зчитування (**R** – *Readable*)
    - Для сегментів даних – дозвіл на записування (**W** – *Writable*)
- Розряд 0 (**A** – *Accessed*) встановлюється при доступі до сегмента

Значення поля типу сегмента

Біт S	Поле type_seg	Тип сегмента
0	0100	Таблиця локальних дескрипторів (LDT)
0	0001 1000 1101 1101	Сегмент стану задачі (TSS)
1	000x	Сегмент даних, тільки для зчитування
1	001x	Сегмент даних, зчитування і записування
1	010x	Не визначено
1	011x	Сегмент стека, зчитування і записування
1	100x	Сегмент коду, тільки виконання
1	101x	Сегмент коду, зчитування і виконання
1	110x	Підпорядкований сегмент коду, тільки виконання
1	111x	Підпорядкований сегмент коду, дозволені зчитування і виконання

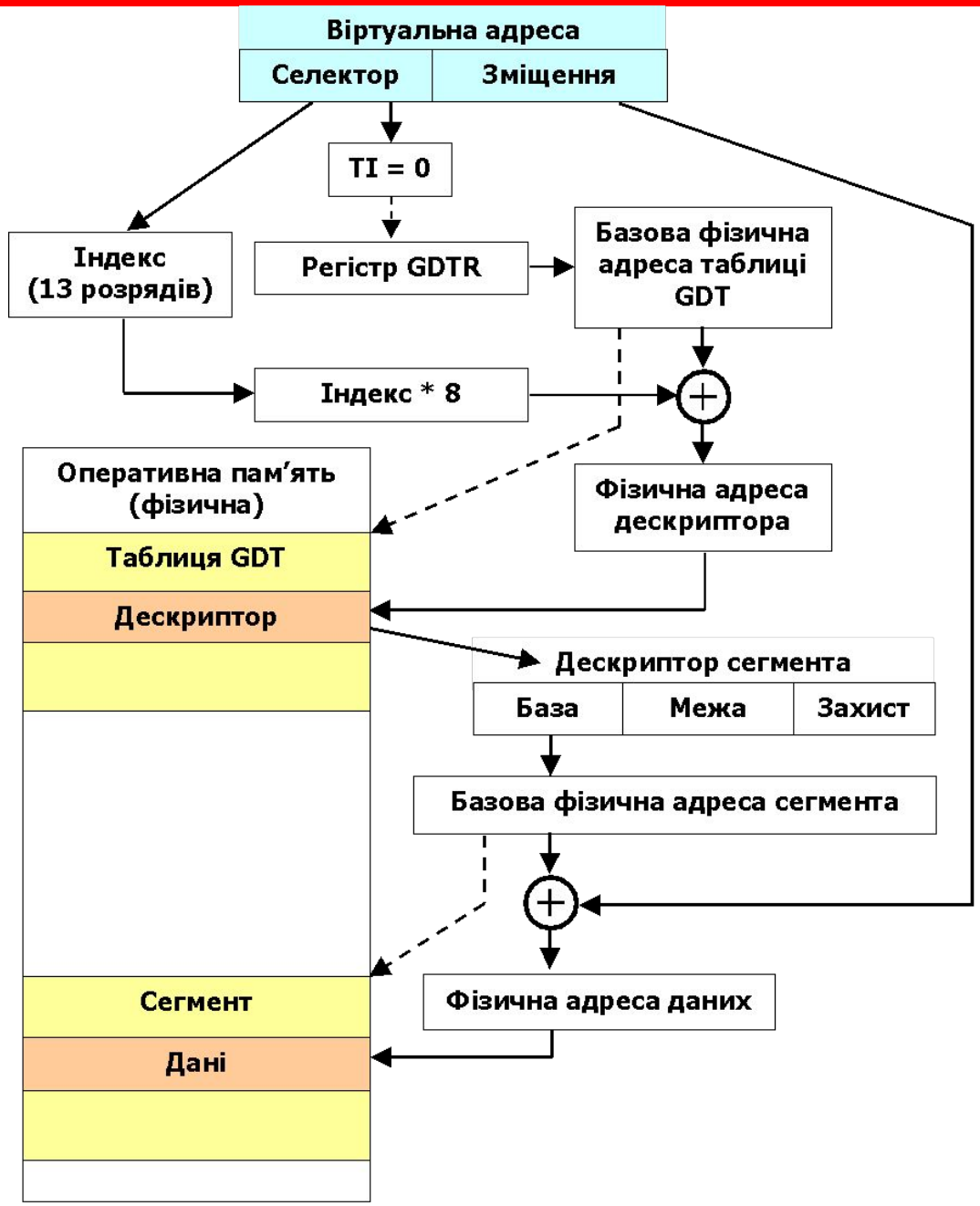
Завантаження селектора у сегментний регістр

- Якщо у селекторі TI == 0, то дескриптор міститься в GDT
  - З регістру gdtr визначають базову адресу і розмір таблиці GDT
  - За індексом вибирають з GDT потрібний дескриптор
  - Перевіряють, чи не виходить дескриптор (з урахуванням його розміру – 8 байтів) за встановлену межу таблиці GDT. В разі виходу за межу – виняткова ситуація 11
  - Перевіряють сумісність типу дескриптора і достатність привілеїв для доступу до нього. Якщо щось не так – виняткова ситуація 13
- Якщо у селекторі TI == 1, то дескриптор міститься в LDT
  - З регістру gdtr визначають базову адресу і розмір таблиці GDT
  - З таблиці GDT вибирають дескриптор, що описує таблицю LDT, для чого в якості селектора використовують вміст регістру ldtr
  - Перевіряють, чи відповідає тип дескриптора таблиці дескрипторів і чи присутня таблиця у фізичній пам’яті
  - З дескриптора таблиці LDT визначають базову адресу таблиці та її межу
  - Далі здійснюють операцію вибору з таблиці необхідного дескриптора, яка аналогічна операції вибору дескриптора з таблиці GDT, з тими ж перевірками.

Дозволені комбінації бітів байту захисту дескриптора при завантаженні селектора у сегментний регістр

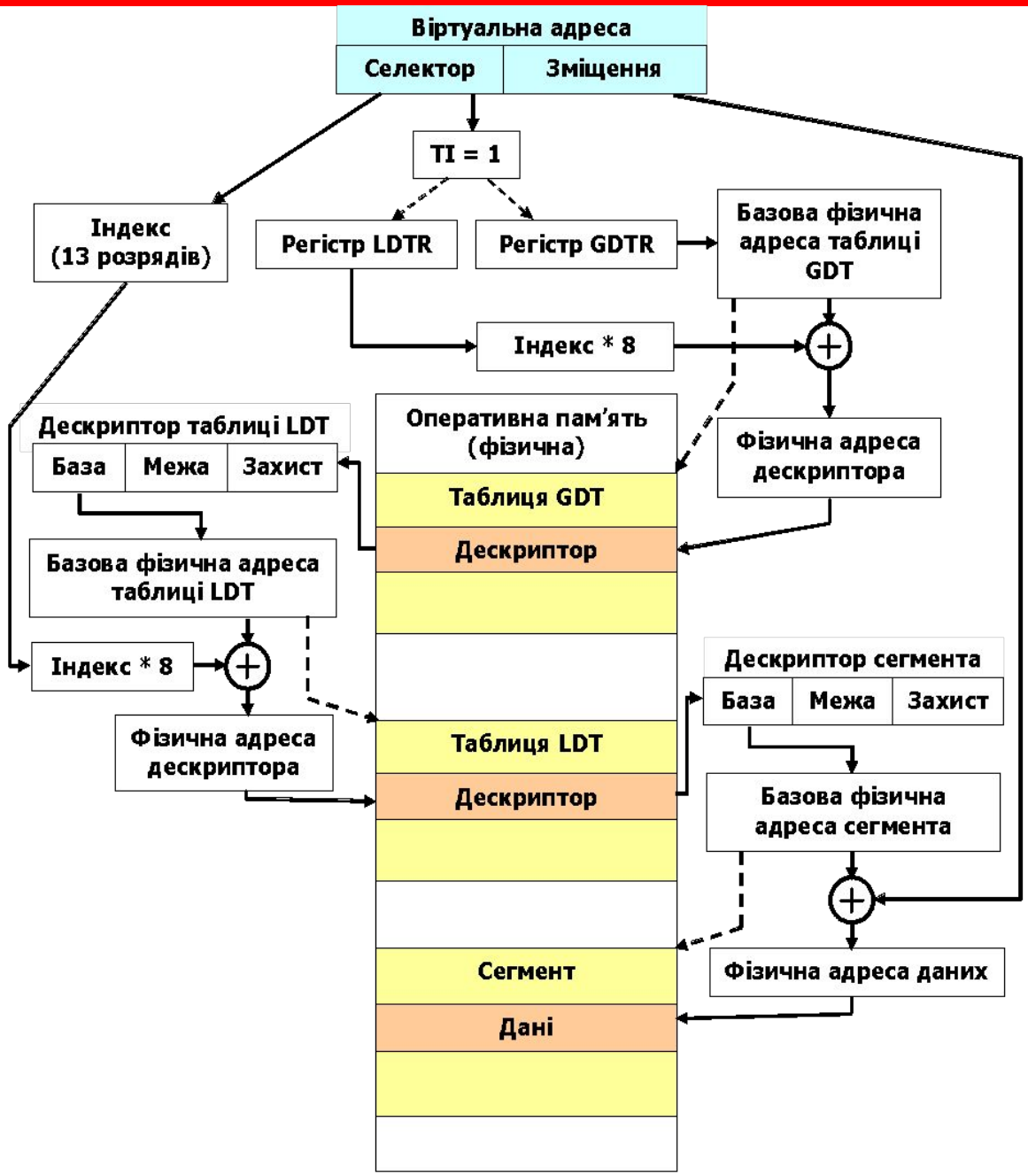
Регістр	DPL	S	E	C/ED	R/W	Примітка
cs	≥ RPL ≥ CPL	1	1	x	x	сегмент коду
ss	=RPL =CPL	1	0	1	1	сегмент стека
ds, es, fs, gs	≥RPL ≥CPL	1	1	x	1	сегмент коду
			0	x	x	сегмент даних або стека

Перетворення адрес за сегментного розподілу пам'яті



дескриптор сегмента знаходиться у таблиці GDT

Перетворення адрес за сегментного розподілу пам'яті



дескриптор сегмента знаходиться у таблиці LDT

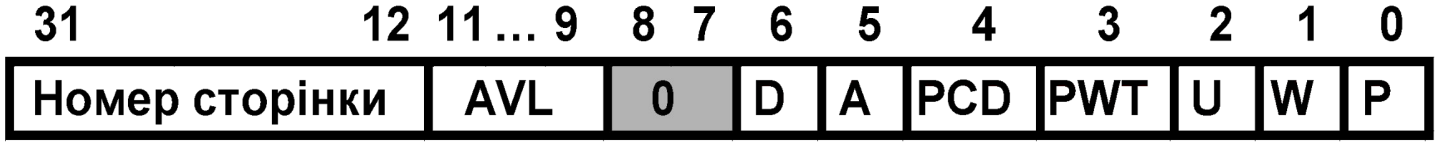
Звернення до пам’яті

- Перевіряють дозвіл на операцію
- Перевіряють коректність доступу (відсутність виходу за межу сегмента)
  - Сегмент стека зростає в бік молодших адрес, і обчислена адреса повинна бути не меншою за межу сегмента
  - Сегменти коду і даних зростають у бік старших адрес, тому до обчисленої адреси додається розмір даних, і отримана адреса повинна бути не більшою за межу сегмента
- Дозволені комбінації бітів байту захисту:

Операція	**S**	**E**	**C/ED**	**R/W**	**Примітка**
Зчитування з пам’яті	1	0	x	x	сегмент даних або стека
	1	1	x	1	сегмент коду
Записування у пам’ять	1	0	x	1	сегмент даних або стека

Сторінковий механізм керування пам'яттю

- Лінійна адреса є 32-розрядною, старші 20 розрядів інтерпретуються як номер сторінки, а молодші 12 – як зміщення в сторінці
- Номер сторінки інтерпретується як індекс дескриптора сторінки, а з дескриптора визначається номер сторінки у фізичній пам’яті



P (Present) Прапорець присутності сторінки у фізичній пам’яті.
W (Writable) Прапорець дозволу записування у сторінку
U (User mode) Прапорець користувач/супервізор
PWTКерують механізмом кешування сторінок (введені, починаючи з процесора i486)
PCD
A (Accessed) Ознака, що мав місце доступ до сторінки
D Ознака модифікації вмісту сторінки
0 Зарезервовані
AVL (Available) Зарезервовані для потреб операційної системи

Регістри, що забезпечують сторінковий механізм

cr0	містить прапорці, які суттєво впливають на роботу процесора і відображають глобальні (незалежні від конкретної задачі) ознаки його функціонування. Деякі важливі системні прапорці з цього регістру: **pe** (*Protect Enable*), біт 0 – вмикає захищений режим роботи процесора; **cd** (*Cache Disable*), біт 30 – вмикає використання внутрішньої кеш-пам'яті (кеш першого рівня); **pg** (*Paging*), біт 31 – вмикає сторінкову трансляцію адрес.
cr2	Містить лінійну віртуальну адресу команди, яка викликала виняткову ситуацію 14 – відсутність сторінки у пам'яті. Обробник цієї виняткової ситуації після завантаження необхідної сторінки у пам'ять має змогу відновити нормальну роботу програми, передавши керування на адресу з cr2
cr3	Містить фізичну базову адресу каталогу сторінок