ЛЕКЦІЯ №6. АПАРАТНІ ПЕРЕРИВАННЯ В ЗАХИЩЕНОМУ РЕЖИМІ

Перехід до захищеного режиму та повернення до реального.

Перед переходом у захищений режим_програмісту потрібно повідомити процесору фізичну адресу таблиці глобальних дескрипторів і її розмір (межу). Для цього завантажити в регістр процесору GDTR інформацію про таблицю глобальних дескрипторів: лінійну базову адресу таблиці та її межу.

Ця інформація розміщується в перших 6 байтах поля даних (псевдодескрипторі). Для завантаження GDTR передбачена спеціальна привілейована команда lgdt (load global descriptor table), яка вимагає вказання в якості операнду імені псевдодескриптору.

7	6	5	4	3	2	1	0	Байти
-		Лінійна базова адреса				Межа		Опис

Рис. 1 Формат псевдодескриптору

Якщо таблиця глобальних дескрипторів розташована на початку сегменту даних і її базова адреса збігається з базовою адресою всього сегменту, то заповнення псевдодескриптору спрощується.

Заборонити всі апаратні переривання. У захищеному режимі процесор виконує процедури переривань не так, як у реальному. При надходженні сигналу переривання процесор не звертається до таблиці векторів переривань у першому кілобайті пам'яті, як у реальному режимі, а витягає адресу програми обробки переривання з таблиці дескрипторів переривань.

Переведення процесора в захищений режим можна виконати командами SMSW (store machine status word, збереження слова стану машини) і LMSW (load machine status word). Потрібно установити біт 0 слова стану машини (регістру CR0) Це біт PE - біт дозволу захисту.

Оскільки інші біти цього слова нам не відомі, потрібно прочитати в регістр AX слово стану машини, установити в ньому біт 0 і записати модифіковане слово стану назад, у процесор.

Усі наступні команди будуть виконуватися в захищеному режимі.

Програміст має справу з селекторами, тобто номерами дескрипторів, а процесор – із самими дескрипторами, що зберігаються в тіньових регістрах, які існують для кожного з сегментних регістрів.

Тіньові регістри недоступні програмісту, вони автоматично завантажуються процесором з таблиці дескрипторів щоразу, коли процесор завантажує відповідний сегментний регістр.

Саме зміст тіньового регістру (у першу чергу, лінійна адреса сегменту) визначає область пам'яті, до якої звертається процесор при виконанні конкретної команди. Після переходу в захищений режим потрібно завантажити у сегментні регістри селектори відповідних сегментів. Це дозволить процесору правильно заповнити всі поля тіньових регістрів з таблиці дескрипторів. Доки ця операція не виконана, деякі поля тіньових регістрів (межі сегментів) можуть містити невірну інформацію.

У сучасних процесорах з метою підвищення швидкості виконання програм використовується конвейєрна обробка команд. Одночасно з виконанням поточної команди здійснюється вибірка операндів наступної, дешифрація третьої і вибірка з пам'яті четвертої. Отже, у момент переходу в захищений режим уже можуть бути розшифровані кілька наступних команд і обрані з пам'яті їхні операнди. Але ці дії виконувалися за правилами реального режиму. Тому потрібно очистити чергу передвиборки перед завантаженням селектору в регістр CS (команда далекого переходу).

Але, якщо програма починала роботу під керуванням системи, то і закінчити її так само потрібно звичайним способом, щоб не зашкодити працездатності системи. У захищеному режимі не можна звертатися до функцій системи і BIOS, тому що це програми реального режиму. У них використовується завантаження в сегментні регістри сегментних адрес.

У захищеному режимі в ці регістри завантажуються селектори. Крім того, використання в захищеному режимі команд із визначеними номерами, призведе до інших результатів. Тобто програму, що працює в захищеному режимі, не можна завершувати засобами системи. Спочатку її потрібно повернути в реальний режим.

Повернення в реальний режим можна здійснити скиданням процесора. Дії процесора після скидання визначаються однією з чарунок КМОП–мікросхеми –

байтом стану відключення, розташованим за адресою Fh. Якщо в цьому байті записаний код Ah, після скидання процесору керування передається за адресою, що витягається з двослівної чарунки 40h:67h, розташованої в області даних BIOS.

Таким чином, для підготовки повернення в реальний режим ми повинні в чарунку 40h:67h записати адресу повернення, а в байт Fh КМОП–мікросхеми занести код Ah. Точка повернення може розташовуватися в будь-якому місці програми.

Інший метод переходу у захищений режим полягає у відкритті лінії A20. Це адресна лінія, на якій встановлюється одиничний рівень сигналу, якщо відбувається звертання до мегабайтів адресного простору з номерами 1,3,5 і т.д. (перший мегабайт має номер 0).

В реальному режимі лінія A20 заблокована і якщо значення адреси виходить за межі FFFFFh, виконується його циклічне обертання (лінійна адреса 1.000000h перетворюється на 00000h і т.д.).

Відкриття лінії A20 вимикає механізм циклічного обертання адрес. Управління блокуванням лінії A20 здійснюється через порт 64h, куди спочатку потрібно відправити команду D1h управління лінією A20, потім - код відкриття (DFh).

Перед переходом в реальний режим потрібно закрити лінію A20. Для цього в порт 64h засилається команда D1h і потім - DDh (закриття лінії).

Перехід у реальний режим.

Скидання процесора здійснюється засиланням команди FFh у порт 64h контролеру клавіатури. Ця команда збуджує сигнал на одному з виведень контролеру клавіатури, що призводить до появи сигналу скидання на виведенні RESET мікропроцесору.

Перед виконанням скидання потрібно зберегти поточний вміст SP, щоб після переходу в реальний режим можна було відновити стан стеку.

Після скидання процесор працює в реальному режимі. Керування передається програмам BIOS. BIOS аналізує стан байту відключення КМОП–мікросхеми і здійснює перехід за адресою, що зберігається в чарунці 40h:67h області даних BIOS.

Скидання процесора виконується не миттєво. Тому, щоб процесор не виконав до своєї зупинки ще декілька команд, потрібно організувати чекання скидання процесора (команди hlt чи нескінченного циклу). Далі потрібно дозволити переривання.

Внутрішні переривання процесора в захищеному режимі.

Виключення.

Обробка переривань у захищеному режимі дуже відрізняється від обробки переривань у реальному режимі. Якщо перейти в захищений режим, не виконавши заборони апаратних переривань, перше ж таке переривання (наприклад, від таймера) призвело б до відключення процесора.

У реальному режимі 32-розрядних процесорів передбачені переривання 3-х видів:

- внутрішні, виникаючі в самому МП;

- зовнішні, що надходять у процесор від зовнішніх пристроїв комп'ютера через контролери переривань;

- програмні, ініційовані командою int.

У захищеному режимі можливі всі ці типи переривань, але функції внутрішніх переривань і їх кількість істотно розширені. Внутрішні переривання називаються тут виключеннями, виключними ситуаціями чи особливими випадками (exception).

Усі переривання захищеного режиму, як і реального, мають свої номери, причому їх загальна кількість не повинна перевищувати 256. Під виключення віддані перші 32 номери 0...31. Реально виникаючі виключення мають номери 0...17, а номери 18...31 зарезервовані для майбутніх моделей процесорів.

У захищеному режимі аналогом таблиці векторів переривань є таблиця дескрипторів переривань (IDT – Interrupt Descriptor Table), яка розташовується в операційній системі чи в програмі користувача. Таблиця IDT містить дескриптори обробників переривань, до яких входять їх адреси. Для того, щоб процесор зміг звертатися до цієї таблиці, її адресу потрібно завантажити в регістр IDTR (командою lidt).

Таблиця дескрипторів переривань складається з дескрипторів, які називаються шлюзами (вентилями). Через шлюзи здійснюється доступ до обробників переривань і виключень. Процесор, зареєструвавши переривання чи виключення, за його номером витягає з IDT шлюз, визначає адресу обробника і передає йому керування.

Обробник повинен закінчуватися командою iret, що повертає керування в перервану програму. Тобто, у програмі, яка обслуговує виключення, треба:

1. Сформувати таблицю дескрипторів переривань IDT;
2. Помістити до неї адреси обробників виключень, передбачених у програмі;
3. Завантажити адресу IDT у системний регістр процесору IDTR;
4. Перейти в захищений режим.

Доки процесор знаходиться в захищеному режимі, він при виникненні переривань буде використовувати IDT. Після повернення в реальний режим (до дозволу переривань) її треба замінити таблицею векторів реального режиму.

У таблицю IDT можуть входити шлюзи наступних типів:

1. Шлюз виклику МП286 (тип 4);
2. Шлюз задачі (тип 5);
3. Шлюз переривання МП286 (тип 6);
4. Шлюз пастки МП286 (тип 7);
5. Шлюз виклику МП386, 486, і Pentium (тип Ch);
6. Шлюз переривання МП386, 486 і Pentium (тип Eh);
7. Шлюз пастки МП386, 486 і Pentium (тип Fh).

Через шлюз задачі (task) здійснюється перемикання на інші задачі в багатозадачному режимі.

Через шлюзи переривань (interrupt) обслуговуються апаратні переривання.

Через шлюзи пасток (trap) здійснюється обробка виключень і програмних переривань.

Формат шлюзів.

Розглянемо формат дескрипторів (рис. 2), з яких будується таблиця переривань IDT (шлюзів). У IDT можуть входити шлюзи трьох типів: пасток, переривань і задач.

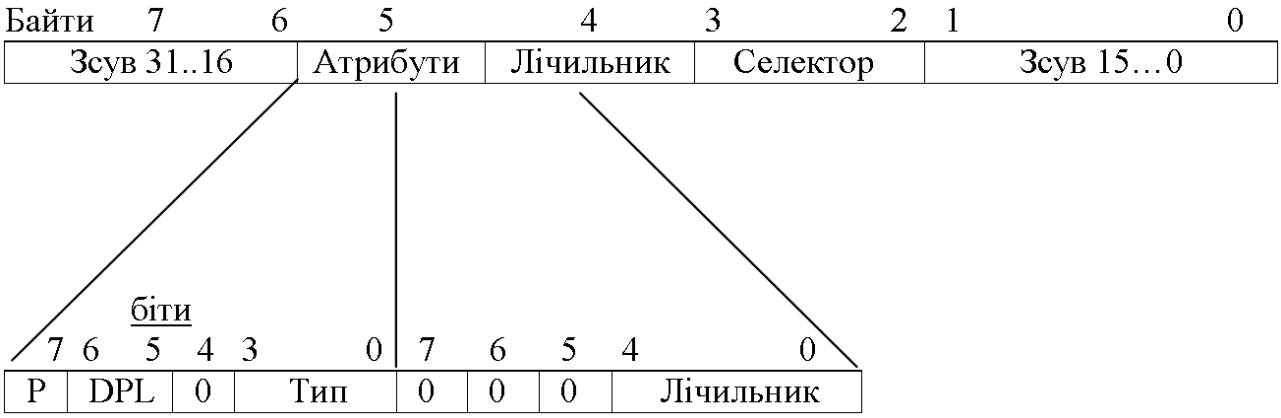

Рис. 2. Формат шлюзів

Якщо основною частиною вмісту сегменту пам'яті була його лінійна адреса, то в шлюзі вказується повна трьохслівна адреса обробника, що складається з селектора та зсуву.

Зсув має 32 біти і займає в шлюзі 2 поля - байти 0-2 і 6-7. Селектор має 16 бітів і займає байти 2,3. Байт 4 не використовується.

Поле лічильника призначене для зберігання числа параметрів, які копіюються з одного стеку на інший в тих випадках, коли здійснюється перехід з одного рівня привілеїв на інший. Кожен параметр – це подвійне слово.

Байт атрибутів має таку ж саму структуру, як і в дескрипторах пам'яті і включає тип, ідентифікатор дескриптору (біт 4), рівень привілеїв дескриптору DPL і біт присутності P. Тип дескриптору може приймати 16 значень, але в IDT припустимо описувати тільки шлюзи задач, переривань і пасток.

Значення поля типу для системних дескрипторів і шлюзів:

- 0 Не визначено;
- 1 Вільний сегмент стану задачі TSS 80286;
- 2 LDT;
- 3 Зайнятий сегмент стану задачі TSS 80286;
- 4 Шлюз виклику 80286;
- 5 Шлюз задачі;
- 6 Шлюз переривання 80286;
- 7 Шлюз пастки 80286;
- 8 Не визначено;

- 9 Вільний сегмент стану задачі;
- Ah Не визначено;
- Bh Зайнятий сегмент стану задачі TSS 386, 486, Pentium;
- Ch Шлюз виклику 80386, i486, Pentium;
- Dh Не визначено;
- Eh Шлюз переривань 80386, i486, Pentium;
- Fh Шлюз пастки 80386, i486, Pentium.

Обробка виключень.

При виникненні виключення процесор множить його номер на 8 і отриманий добуток використовує, як індекс у таблиці дескрипторів переривань IDT. Перші 18 дескрипторів IDT повинні завжди описувати програми обробників виключень.

Виключення підрозділяються на три класи: порушення, пастки й аварії.

Порушення чи відмова (fault) - це виключення, що фіксується ще до виконання команди чи в процесі її виконання. Приклади порушень: адресація за невстановленою межею сегменту чи звертання до відсутнього дескриптору. При обробці порушення процесор залишає в стеку адресу тієї команди, виконання якої призвело до виключення. При цьому передбачається, що в обробнику порушення його причина буде ліквідована, після чого команда iret поверне керування на ту ж саму, ще невиконану команду. Тобто, сам механізм обробки порушень припускає відновлення цього програмного збою.

Пастка (trap) обробляється процесором після виконання команди, що викликала це виключення й у стеку зберігається адреса не цієї, а наступної команди. Тобто, після повернення з обробника пастки виконується не команда, що ініціювала виключення, а наступна команда програми. До пасток відносяться всі команди програмних переривань int.

Аварія чи вихід із процесу (abort) є наслідком серйозних помилок що не відновлюються, наприклад, виявлення в системних таблицях недозволених чи несумісних значень. Адреса, що зберігається в стеку, не дозволяє локалізувати команду, що викликала виключення, і відновлення програми не передбачається. Зазвичай аварії вимагають перезавантаження системи.

Процесор, зареєструвавши те чи інше виключення, зберігає в стеку вміст розширеного регістру прапорів EFLAGS, селектор сегменту команд, зсув точки повернення і, у деяких випадках, 32-бітовий код помилки.

Стан стеку при виключенні.

Навіть якщо програма виконується в режимі 16-бітових адрес і операндів, як зсув повернення виступає 32-бітовий вміст покажчика команд EIP, у якому старша половина дорівнює нулю.

Двослівні дані розташовуються в стеку в такому порядку: у слові стеку з меншою адресою – молодша частина 32-бітного даного, а в слові стеку з більшою на 2 адресою – старша частина.

Для вирівнювання стеку під вміст CS також виділяється подвійне слово в молодшій половині якого розташовується CS, а старша половина нічим не заповнюється. У режимі 16-розрядних операндів використовується тільки молодша половина ESP (SP).

Перед завершенням обробника (командою iret) код помилки потрібно зняти зі стеку. Тобто, для правильної обробки виключення необхідно знати причину його виникнення і вид, а також куди повернеться керування після завершення обробника, і чи включений у стек код помилки.

При переході на обробник виключення процесор заносить у стек адресу повернення. В обробнику виключення цю адресу можна витягти і вивести на екран з номером виключення, що виникло.

При витягненні зі стеку адреси повернення треба враховувати можливість наявності в стеку коду помилки.

При налагодженні програм захищеного режиму найчастіше виникає виключення 13 порушення загального захисту.

Наприклад, звертання до сегменту даних за відносною адресою, яка виходить за його межі:

mov AX, DS:[data_size - 1]; виникнення виключення 13.

Байт із номером data_size – 1 – останній байт сегменту даних. Якби ми звернулися за цією адресою командою читання байту, усе було б нормально:

mov AL, DS:[data_size - 1]

При читанні цілого слова другий байт цього слова знаходиться за межами сегменту даних і виникає порушення загального захисту.

Табл. 2 Виключення процесора

Вектор	Назва виключення	Клас виключення	Код пом.	Команди, які викликають виключення
0	Помилка ділення	Порушення	Немає	div, idiv
1	Виключення налагодження	Порушення	Немає	Будь-яка команда
2	Немасковане переривання	Аварія	Немає	
3	Int3	Пастка	Немає	int 3
4	Переповнення	Пастка	Немає	Into
5	Порушення границь масиву	Порушення	Немає	Bound
6	Неприпустимий код команди	Порушення	Немає	Будь-яка команда
7	Співпроцесор недоступний	Порушення	Немає	esc, wait
8	Подвійне порушення	Аварія	так	Будь-яка команда
9	Вихід співпроцесора з сегменту	Аварія	Немає	Команда співпр. звертання до пам'яті
10	Неприпустимий сегмент стану задачі TSS	Порушення	так	Jmp, call, iret, переривання
11	Відсутність сегменту	Порушення	так	Команда завантаження сегм. регістру
12	Помилка звертання до стеку	Порушення	Так	Команда звертання до стеку
13	Загальний захист	Порушення	так	Команда звертання до пам'яті
14	Сторінкове порушення	Порушення	Так	Команда звертання до пам'яті
15	Зарезервовано			
16	Помилка спів процес.	Порушення	немає	esc, wait
17	Помилка вирівнювання	Порушення	Так	Команда звертання до пам'яті
18...31	Зарезервовані			
32...255	Надані користувачу для апаратних переривань і команд int.			

У такий спосіб у захищеному режимі процесор стежить за тим, щоб програма не зверталася до невиділених їй ділянок пам'яті. Команда mov AL, CS [0]; порушення загального захисту повинна прочитати в регістр AL перший байт сегменту команд. Але, якщо атрибут сегменту команд дорівнює 98h, команда виконана не буде, а збудить порушення загального захисту.

Таким чином, у захищеному режимі коди команд у сегменті команд з кодом 98h не можна ні модифікувати, ні навіть читати. (З кодом сегменту 9Ah це робити можна).

Порушення загального захисту виникає в тому випадку, коли програма звертається до пам'яті за межею таблиці дескрипторів.

Припустимо, в нашій програмі описана таблиця, що містить 5 дескрипторів. Рядки, що завантажують у сегментний регістр селектор, який відсутній у таблиці глобальних дескрипторів, викликають порушення загального захисту.

mov AX, 40

mov FS, AX; порушення загального захисту.

Друга команда цього фрагменту повинна завантажити в тіньовий регістр, зв'язаний із сегментним регістром FS, дескриптор, що відповідає селектору 40, тобто шостий за рахунком дескриптор.

Така ж ситуація виникає, якщо в програмі зустрічається команда переривання INT з номером, відсутнім у таблиці дескрипторів переривань.

Виключення порушення стеку з №Ch(12) виникає, наприклад, при спробі витягти слово з порожнього стеку.

Виключення з вектором 11 виникає при звертанні до сегменту, позначеного як відсутній. У дескрипторі будь-якого сегменту є біт присутності сегменту в пам'яті. Він знаходиться в розряді 7 байту атрибутів 1 і позначається P від Present.

Біт присутності призначений для організації віртуального режиму, у якому сумарний розмір усіх виконуваних одночасно програм може перевищувати фактичний обсяг оперативної пам'яті.

У цьому випадку операційна система зберігає частину сегментів програм на диску, завантажуючи їх у пам'ять у міру необхідності на місце тих сегментів, до яких тимчасово не відбувається звертань. Робота з бітом присутності – функція операційної системи.

Вивантаживши якийсь сегмент на диск, система скидає біт P в дескрипторі цього сегменту. Дескриптор знаходиться в пам'яті. Якщо програма робить спробу звертання до відсутнього сегменту, виникає виключення, що змушує систему завантажувати необхідний сегмент у пам'ять. У його дескрипторі встановлюється біт P, позначаючи його присутнім.

Якщо прикладна програма бере на себе частину функцій операційної системи, вона сама може встановлювати і скидати біт присутності.

Виключення 10, 11, 12, 13 відрізняються від інших тим, що перед передачею керування обробнику, процесор заносить у стек крім регістру прапорів і адреси повернення ще і код помилки (на верхівці стеку). Код помилки має формат зображений на рисунку 3.5.

31...16	15...4	3	2	1	0
Сміття	Індекс дескриптору		TI	IDT	EXT

Рис. 3 Формат коду помилки

Біт 0 – EXT (External, зовнішній) – дорівнює 1, якщо виключення виникло в результаті обробки іншого виключення чи зовнішнього переривання.

Біт 1 (IDT)=1, якщо виключення виникло в результаті читання елемента в таблиці дескрипторів переривань, що може мати місце тільки при обробці виключення чи переривання.

Біт 2 (TI – Table Indicator, індикатор таблиці) дорівнює 1, якщо дескриптор знаходиться в таблиці локальних дескрипторів і 0, якщо дескриптор глобальний.

Використавши інформацію, що знаходиться в двослівному коді помилки, програма може відновити початковий вміст регістрів команди, в процесі виконання якої виникло виключення. Встановивши біт присутності в потрібному дескрипторі, потрібно виконати цю ж саму команду.

Апаратні переривання в захищеному режимі.

Обробка апаратних переривань у захищеному режимі.

Апаратним перериванням потрібно призначати вектори переривань відмінні від 0...31. У машинах типу IBM PC контролери переривань у процесі завантаження завжди програмуються так, що базовий вектор ведучого контролера дорівнює 8, а підрядного – 70h. Отже, перед переходом у захищений режим потрібно перепрограмувати контролери переривань, призначивши ведучому контролеру, наприклад, базовий вектор 20h=32.

Очевидно, що перед поверненням у реальний режим контролери треба знову перепрограмувати, інакше не зможуть працювати обробники апаратних переривань BIOS. Але можна зробити інакше: перехід у реальний режим організувати за допомогою відключення процесору, попередньо завантаживши в область даних BIOS за адресою 40h:67h двослівну адресу повернення в програму. Якщо при цьому заслати в байт 0Fh КМОП – мікросхеми код 05h, то після скиду процесору BIOS перепрограмує контролери переривань і передасть керування в зазначену точку.

Друга особливість обробки переривань пов'язана з форматом дескриптору переривання (шлюзу). У дескрипторі тип шлюзу вказується в байті атрибутів 1.

Біти	7	6		5	4	3	2	1	0
	Назва	P	DPL	0	Тип шлюзу				

Рис. 4. Формат байту Атрибутів 1:

Поле типу шлюзу може приймати 16 різних значень, але в таблиці IDT можна описувати тільки 5, раніше розглянутих:

1. Тип задачі –5;
2. Тип перерив. 286 – 6;
3. Пастка – 7;
4. Тип перив. 386 Eh;
5. Пастка Fh.

Шлюз задачі використовується в тих випадках, коли обробник переривань розташований в іншій задачі. Шлюзи переривань і пасток припускають наявність обробників у поточній задачі.

Поле DPL визначає рівень привілеїв шлюзу. Це поле перевіряється процесором тільки при виконанні команд програмних переривань - пасток int та into, щоб запобігти звертанню програм користувача до системних програмних переривань. Для інших виключень і переривань поле DPL процесором ігнорується.

Біт P в дескрипторі шлюзу повинен бути встановлений у 1. В іншому разі, звертання до такого шлюзу викликає виключення.

Таким чином, атрибут дескриптору шлюзу типу переривання повинен дорівнювати 8Eh, на відміну від дескриптору пасток - 8Fh.

Якщо перехід на обробник здійснюється через шлюз переривання, процесор скидає при вході в обробник прапор IF у регістрі прапорів EFLAGS. Команда iret, завантажуючи зі стеку збережений там уміст EFLAGS, знову дозволяє переривання. При переході на обробник через шлюз пастки стан прапору IF не змінюється.