

## ЛЕКЦІЯ №4. РОБОТА З ПАМ'ЯТТЮ В РЕАЛЬНОМУ РЕЖИМІ ТА ОСОБЛИВОСТІ ЗАХИЩЕНОГО РЕЖИМУ

### Розподіл пам'яті у реальному режимі x86 процесора.

Кожному блоку пам'яті передуює 16 байтовий блок керування пам'яттю (МСВ), що визначає власника блоку, розмір блоку, і, якщо блок містить програму, ім'я програми. Блок, що не має власника, вільний і може бути призначений програмі. МСВ утворює ланцюжок, що визначає структуру пам'яті. Якщо встановлене місцезнаходження головного МСВ, можна знайти блок, розташований у більш старших адресах пам'яті, додавши до сегментної адреси цього МСВ довжину блоку пам'яті, що йде за МСВ і ще 1. (1 додається тому, що довжини блоків пам'яті виражаються в параграфах і сам МСВ займає 1 параграф).

Кожний МСВ містить 4 поля. Байт із зсувом 0 (байт сигнатури) визначає, чи є цей МСВ останнім у ланцюжку. Розмір  $4Dh(N)$  означає, що за цим МСВ впливає інший;  $5Ah(Z)$  - блок останній у ланцюжку. Слово зі зсувом 1 визначає власника блока пам'яті, що впливає за МСВ. Воно містить сегментну адресу програми, для якого виділений цей блок пам'яті. Розмір 0 означає, що блок на даний момент не виділений. Слово зі зсувом 3 встановлює в байтах довжину останнього блока пам'яті. Якщо блок пам'яті містить програму, її ім'я з'являється в МСВ із зсувом 8. Ім'я зберігається у форматі ASCII і закінчується нулем, якщо його довжина менше 8 символів. Коли в Config.sys включена команда DOS=UMB, система створює ланцюжок МСВ у верхній пам'яті.

За замовчуванням цей ланцюжок не пов'язаний із ланцюжком МСВ у звичайній пам'яті. Блоки пам'яті, що резервуються для операційної системи, мають значення  $08h$  у полі ідентифікатора власника. Байти 8 і 9 у відповідних МСВ містять літери SC і SD, що означають системний код і системні дані. Блоки, позначені SD, підрозділяються на менші блоки. Цим меншим блокам передують свої Суб-МСВ, формат яких ідентичний формату дійсних МСВ, але замість M і Z байт сигнатури містить B, D, F, L, S, X. Кожна з цих літер ідентифікує зміст блока пам'яті, що впливає за Суб-МСВ. Літера B встановлює, що блок містить дискові буфери, пов'язані з командою BUFFERS; D - указує, що блок містить драйвер пристрою; F -

блок містить системну таблицю файлів; L - блок береже дані логічного типу, S - блок містить внутрішні стеки системи; X - кешує дані MCB.

Програмам майже не припадає управляти блоками пам'яті на такому низькому рівні. Щоб полегшити для програм роботу з пам'яттю, система надає такі системні функції:

1. 48h виділяє блок пам'яті.
2. 49h звільняє виділений блок пам'яті.
3. 4Ah змінює розмір виділеного блока пам'яті.
4. 5800h і 5801h одержують і встановлюють стратегію виділення пам'яті.
5. 5802h і 5803h одержують і встановлюють стан "прив'язки" верхньої пам'яті.

### ***Виділення блоків верхньої пам'яті.***

За замовчуванням всі блоки пам'яті, що виділяються функцією 48h, беруться зі звичайної пам'яті. Якщо система містить вільні блоки верхньої пам'яті, то використовуючи функцію 48h можна виділити UMB. Спочатку програма повинна зв'язати ланцюжка MCB у верхній і звичайній пам'яті. Цей процес називається прив'язкою верхньої пам'яті. Потім програма повинна змінити стратегію виділення пам'яті. Ці дві дії можуть бути виконані за допомогою функцій 5803h і 5801h.

Функція 5803h. Управляє зв'язком між звичайною і верхньою пам'яттю.

Вхід:

AX - код функції

BX - код зв'язку: 1 - верхня і звичайна пам'ять зв'язуються,

0 - зв'язок скасовується.

Вихід: Прапор переносу скинуто - виклик завершений успішно; прапор переносу встановлений - відбулася помилка. Функція завершується невдало, якщо відбувається спроба встановити зв'язок із верхньою пам'яттю, у той час як UMB немає. Наприклад, якщо EMM386. EXE не завантажений або завантажений без ключів RAM або NOEMS. Вона також зазнає невдачі, якщо UMB є, у Config. sys відсутня команда DOS=UMB.

Функція 5801h. Встановлює стратегію, що система повинна використовувати при пошуку вільних блоків у процесі задоволення запиту виділення пам'яті.

Вхід:

- АХ - код функції;
- ВХ - код стратегії виділення пам'яті (табл. 1).

Табл. 1 Коди стратегій

| Код | Стратегія                               | Опис   |
|-----|---|--|
| 00h | Нижній-перший/<br>пошук від початку     | Пошук ведеться спочатку в звичайній пам'яті, потім у верхній. Повертається адреса першого знайденого блоку, обсяг котрого більше або дорівнює розміру переданого ф. 48h. |
| 01h | Нижній-перший/<br>оптимальний<br>пошук  | Пошук спочатку в звичайній пам'яті, потім у верхній. Повертається адреса блоку, обсяг якого найбільше близький до розміру, переданого ф. 48h.                            |
| 02h | Нижній-перший/<br>пошук від кінця.      | Пошук спочатку в звичайній пам'яті, потім у верхній. Повертається адреса останнього блоку, обсяг котрого більше або дорівнює розміру, переданого ф. 48h.                 |
| 40h | Тільки-верхній/<br>пошук від початку    | Пошук тільки у верхній пам'яті. Повертається адреса першого знайденого блоку, обсяг котрого більше або дорівнює розміру переданого ф. 48h.                               |
| 41h | Тільки верхній/<br>оптимальний<br>пошук | Пошук тільки у верхній пам'яті. Повертається адреса блоку, обсяг якого близький до розміру, переданого ф. 48h.   |
| 42h | Тільки верхній/<br>пошук від кінця      | Пошук у верхній пам'яті.   |
| 80h | Верхній-перший/<br>пошук від початку    | Пошук спочатку у верхній, потім у звичайній пам'яті.   |
| 81h | Верхній-перший/<br>оптимальний<br>пошук | Спочатку у верхній, потім у звичайній пам'яті.   |

Після того, як зв'язок із верхньою пам'яттю і стратегія виділення пам'яті встановлена, функція 48h може бути використана для призначення вільних UMB.

Звільнення блоків верхньої пам'яті.

Програма звільняє UMB тим же засобом, що і блок звичайної пам'яті: викликаючи функцію 49h з сегментною адресою блоку в реєстрі ES. Щоб UMB міг бути звільнений, повинен бути встановлений зв'язок із верхньою пам'яттю. У протилежному випадку функція 49h поверне встановлений прапор переносу і UMB залишиться виділеним.

### ***Зберігання і відновлення стратегії виділення пам'яті і стану зв'язку.***

Всякий раз, коли програма змінює стан зв'язку з верхньою пам'яттю, їй варто спочатку зберегти поточний стан зв'язку і відновити його по завершенні. Точно також, перед тим, як змінити стратегію системи виділення пам'яті, програмі варто зберегти поточну стратегію і потім відновити її перед завершенням. Для визначення всередині програми поточної стратегії виділення пам'яті і стану зв'язку системи надає функції 5800h і 5802h. Функція 5800h повертає код поточної стратегії в АХ.

Аналогічно, функція 5802h повертає 0 у AL, якщо зв'язок із верхньою пам'яттю не встановлений; у противному випадку вона повертає 1. Відновлення стана зв'язку з верхньою пам'яттю виконується функцією 5803

#### **Приклад призначення UMB:...**

```
mov ax,5800h ; зберігаємо цю
int 21h      ;стратегію
mov strategy,ax ; виділення
mov ax,5804h ;зберігаємо поточну
int 21h      ; прив'язку верхньої пам'яті
mov byte ptr linkstate,al
mov ax,5803h ; встановлення зв'язку з верхньою
mov bx,1     ; пам'яттю
int 21h
mov ax,5801h ;переключаємося
mov bx,40h ;на стратегію
int 21h ;"тільки верн./пошук від початку"
mov ah,48h ; виділяє блок пам'яті
mov bx,400h ; обсягом 16к
int 21h
jc error
mov addr,ax ; збер. сегм. адресу
mov ax,5803h ; відновлення стану
mov bx,linkstate ; зв'язку з верх. п.
int 21h
mov ax,5801h ;відновлення стратегії
mov bx,strategy ; виділення.
int 21h
...
```

Написана таким чином програма лишає недоторканими внутрішні прапори, що контролюють метод виділення пам'яті, яка використовується системою.

## ***Керування пам'яттю.***

Карти пам'яті. Пам'ять ПК складається з множини інтегральних мікросхем (чипів). Для програм неважливе фізичне розташування чипів, але важливо, у якому логічному просторі розташована пам'ять ПК. Цей логічний простір називається адресним простором пам'яті.

Карта пам'яті - це діаграма розподілу збереженої інформації. На ній також зазначені місця, що у принципі можуть використовуватися для збереження даних, хоча і не призначені для цього.

Засоби уявлення карти пам'яті:

- Найменша адреса - вгорі (за аналогією з книгою).
- Найменша адреса - знизу, тому що програми розміщуються від нижніх адрес пам'яті до верхніх.

## ***Пам'ять у реальному режимі.***

Під пам'яттю в реальному режимі розуміється область пам'яті з адресами від 0 до 1 Мбайта (1024 кб). IBM прийняла декілька рішень про використання першого мегабайту простору, що адресується: перші 5/8 першого мегабайту використовуються для операційної системи і для будь-якої іншої програми, що виконується. Це логічні сторінки від 0 до 9. Сторінки А і В зарезервовані для Відео-РАМ. Сторінки С і D були зарезервовані для подальшого застосування. Сторінки С і F - для системної ROM (постійної пам'яті). Ці рішення призвели до появи бар'єра в 640 Кбайт.

Реальний режим називається так тому, що є операційним режимом сімейства процесорів 80x86 у момент подачі на них напруги або після перезавантаження системи.

У реальному режимі немає розходження між лінійними і фізичними адресами. (Лінійна адреса = сегмент \* 16 + зсув). Максимальна фізична адреса = 1 Мб.