

4.25.Програмування чисел підвищеної розрядності

Поняття "підвищена розрядність" тут означає таку кількість біт, яку процесор не може обробити одною командою. Наприклад, у процесорах 32-бітової архітектури арифметичні команди виконуються для 32-бітових чисел і таку саму розрядність мають робочі регістри (EAX, EBX, ECX, EDX тощо). Процесор 64-бітовий здатен одною командою додавати вже 64-бітові числа. Проте, може виникнути потреба у роботі з числовими даними, розрядність яких може суттєво – у десятки разів перевищувати довжину машинного слова. Підвищена розрядність означає підвищену точність та (або) більший діапазон числових даних. Реалізувати арифметику підвищеної розрядності можна власноруч програмним шляхом. Мова асемблера дозволяє це реалізувати достатньо просто та ефективно.

Ми використовуємо пам'ять, а в регістрах тримаємо адресу на це число, зсуваючи адресу пам'яті (наприклад по адресі 00000000h знаходиться молодші 32 біти числа 800000028000000180000000h тобто якщо ми додамо до адреси одиницю, ми перейдемо до 80000001h наступні 32 біти числа і т.д.)

Це можна реалізувати беручи адресу та число з попереднього прикладу

`mov ebx, 00000000h` ; адреса молодших 32 бітів числа

`mov eax, dword ptr[ebx]` ; в eax запишиться 80000000h

`mov eax, dword ptr[ebx + 4]` ; в eax запишиться 80000001h

`mov eax, dword ptr[ebx + 8]` ; в eax запишиться 80000002h

Ви спитаєте чому саме 4 та 8, я вам відповім, зсув відбувається байтами, 32 біти – 4 байти.

4.26. Використання функцій API Windows. Приклади програмування

Для забезпечення можливостей створювати програми, які працюватимуть у середовищі Windows, програмістам наданий інтерфейс API (Application Program Interface) у вигляді прототипів системних функцій та інших бібліотечних файлів. При розробці власних програм програмісти можуть писати у вихідних текстах програмного коду виклики функцій API Win32. У такий спосіб можна створювати програми на різноманітних мовах програмування, у тому числі на асемблері. Написання на асемблері програм з викликами системних функцій було відоме ще давно, наприклад, у MS-DOS – там виклики системних функцій були оформлені як переривання INTnn. Виклик функцій API Win32 у програмах на асемблері зручно програмувати за допомогою директиви INVOKE. Формат виклику `invoke Ім'яФункції, параметри`

Приклад запису у файл

```
.data
    hFile dd 0
    pRes dd 0
    szFileName db "tmp.txt",0
    szTextBuf db "Рядок тексту, записаний у файл",0

.code
    . . .
    invoke CreateFile, ADDR szFileName,
        GENERIC_WRITE,
        FILE_SHARE_WRITE,
        0, CREATE_ALWAYS,
        FILE_ATTRIBUTE_NORMAL,
        0

    cmp eax, INVALID_HANDLE_VALUE
    je @exit ;доступ до файлу неможливий
    mov hFile, eax
    invoke strlen, ADDR szTextBuf
    invoke WriteFile, hFile, ADDR szTextBuf, eax, ADDR pRes, 0
    invoke CloseHandle, hFile

@exit:
    . . .
```

