Бібліотека асемблерних підпрограм asmio16

Перелік підпрограм бібліотеки asmio16

Основні функції	
`Crlf`	переводить курсор на початок нового рядка
`WriteChar`	виводить символ на екран
`WriteString`	виводить рядок символів на екран
`WriteByteHex`	виводить однобайтове число на екран як шістнадцяткове
`WriteWordHex`	виводить двобайтове число на екран як шістнадцяткове
`WriteUnsigned`	виводить на екран двобайтове число як десяткове беззнакове
`WriteInt`	виводить на екран двобайтове число як десяткове із знаком
`ReadChar`	вводить один символ з клавіатури
`ReadString`	вводить рядок символів з клавіатури
`ReadInt`	вводить з клавіатури десяткове число із знаком і перетворює його в двобайтове двійкове число
`ReadHex`	вводить з клавіатури шістнадцяткове число і перетворює його в двобайтове двійкове число
Допоміжні функції	
`IsDigit`	перевіряє, чи є число десятковою цифрою
`IsHex`	перевіряє, чи є число шістнадцятковою цифрою
`atoh`	перетворює ASCII-рядок шістнадцяткових цифр в ціле число
`chtoh`	перетворює ASCII-код шістнадцяткової цифри в цифру

Детальний опис підпрограм бібліотеки asmio16

Основні функції

Crlf

Виклик цієї підпрограми переводить курсор на екрані монітора на початок наступного рядка. Для цього на стандартний пристрій виведення посилаються коди керуючих символів 13, 10 (0Dh, 0Ah), які, відповідно, повертають курсор на початок рядка (Carriage Return або CR) і переміщують його на наступний рядок (Line Feed або LF).
Виклик підпрограми:

```
call Crlf
```

WriteChar

Підпрограма виводить на екран один символ, заданий його ASCII-кодом в регістрі AL.
Приклад використання підпрограми:

```
mov al, 'A'
call WriteChar
```

WriteString

Підпрограма виводить на екран рядок символів, який завершується нулевим байтом. Адреса текстового рядка передається через регістр DX. Приклад використання:
В сегменті даних

```
.data
message db "Hello, world!", 0
```

В сегменті коду

```
.code
    lea dx, message
    call WriteString
```

WriteByteHex

Підпрограма виводить на екран однобайтове ціле число з регістру AL у шістнадцятковій системі числення (діапазон чисел 0…FF). Приклад використання:

```
mov al, 0FCh
call WriteByteHex
```

WriteWordHex

Підпрограма виводить на екран двобайтове ціле число з регістру AX у шістнадцятковій системі числення (діапазон чисел 0…FFFF). Приклад використання:

```
mov ax, 0A81Eh
call WriteWordHex
```

WriteUnsigned

Підпрограма виводить на екран двобайтове ціле число з регістру AX як десяткове беззнакове (діапазон чисел 0…65535). Приклад використання:

```
mov ax, 45869
call WriteUnsigned
```

WriteInt

Підпрограма виводить на екран двобайтове ціле число з регістру AX як десяткове число із знаком (діапазон чисел –32768…+32767). Приклад використання:

```
mov ax, -23765
call WriteInt
```

ReadChar

Підпрограма вводить з клавіатури один символ і поміщає його ASCII-код в регістр AL. При цьому введений символ виводиться на екран. Використання:

```
Call ReadChar
```

ReadString

Підпрограма вводить з клавіатури рядок символів і поміщає його в буфер, адреса якого передається через регістр DX. Рядок автоматично завершується нульовим байтом. Довжина буфера вказується через регістр CX (довжина буфера визначається із врахуванням завершального нульового байта). Введення рядка завершується при натисканні клавіші Enter або при досягненні кінця буфера. В регістрі AX повертається кількість уведених символів. Приклад використання:

В сегменті даних

```
.data
buffer db 256 dup (0)
```

В сегменті коду

```
.code
     lea dx, buffer
     mov cx, 10
     call ReadString
```

ReadHex

Підпрограма вводить з клавіатури шістнадцяткове число (від однієї до чотирьох цифр у діапазоні 0…FFFF) і перетворює його в двобайтове двійкове число в регістрі AX. Введення завершується натисканням клавіші Enter. Шістнадцяткові цифри в діапазоні від A до F можна вводити як в нижньому так і у верхньому регістрах. Підпрограма перевіряє уведене шістнадцяткове число на коректність. Якщо число містить лише шістнадцяткові цифри, то CF=0,

а в регістрі AX повертається результат. Якщо введене число містить невірні символи, то CF=1, AX=0.

```
call ReadHex
jc hexerror    ; обробити помилку
```

ReadInt

Підпрограма вводить з клавіатури десяткове число із знаком і перетворює його в двобайтове двійкове число в регістрі AX. Число, яке вводиться може починатися сиволами «+» або «–». Діапазон чисел становить –32768…+32767. Уведення числа завершується натисканням клавіші Enter. Десяткові цифри перевіряються на коректність за допомогою функції IsDigit. Якщо введене десяткове число коректне, то CF=0, а AX містить результат. Якщо введене число містить недесяткові цифри або виходить за межі дозволеного діапазону, то CF=1, AX=0.

```
call ReadInt
jc decerror    ; обробити помилку
```

Допоміжні функції

IsDigit

Підпрограма перевіряє, чи є символ в AL ASCII-кодом десяткової цифри. Якщо AL містить ASCII-код десяткової цифри (30h…39h), то CF=0. Якщо символ не є десятковою цифрою то CF=1.

IsHex

Підпрограма перевіряє, чи є символ в AL ASCII-кодом шістнадцяткової цифри. Якщо AL потрапляє в один з діапазонів ‘0’…‘9’, ‘A’…‘F’ або ‘a’…‘f’ (коди 30h…39h, 41h…46h або 61h…66h), то CF=0. Якщо код в AL не представляє шістнадцяткову цифру, то CF=1.

atoh

Підпрограма перетворює ASCII-рядок шістнадцяткових цифр в ціле число. На рядок, який містить від однієї до чотирьох шістнадцяткових цифр і закінчується нульовим байтом, вказує регістр DX. Шістнадцяткові цифри перевіряються на коректність за допомогою функції IsHex. Якщо перетворення пройшло успішно, то CF=0, а AX містить результат. Якщо були некоректні шістнадцяткові цифри, то CF=1, AX=0.

chtoh

Підпрограма перетворює ASCII-код шістнадцяткової цифри в ціле число в діапазоні 0…F (0000…1111 у двійковій системі). Перевірка цифри на коректність не проводиться!

Макроси

.begin

Макрос описує точку входу в програму (Start:) і реалізує код ініціалізації сегментних регістрів DS, ES.

.end

Макрос генерує код завершення програми (функція 4Ch int 21h) і повернення в операційну систему.

Використання бібліотеки асемблерних підпрограм asmio16

Бібліотека підпрограм asmio16 використовує модель пам'яті SMALL і призначена для створення 16-розрядних програм компіляторам TASM та компонувальником TLINK. Бібліотека міститься у файлі asmio16.lib. Для підключення бібліотеки до програми використовується заголовковий файл asmio16.inc. Файли asmio16.lib та asmio16.inc повинні міститися в одному каталозі з програмою або шлях до них повинен задаватися командою PATH з командного рядка.

Скелет програми, яка використовує бібліотеку asmio16, має такий вигляд:

```
include asmio16.inc

.data
     ; тут розміщуються дані

.code
.begin
     ; тут розміщується код

.end
```

Приклад програми із використанням бібліотеки asmio16:

```
include asmio16.inc

.data
mes db "Hello, world!",0

.code

.begin

     lea dx, mes
     call WriteString

.end
```