Другие темы раздела

FASM Уроки Iczelion'a на FASM https://www.cyberforum.ru/ fasm/thread1240590.html
Уроки Iczelion'a на FASM Урок первый. MessageBox на FASM format PE GUI include 'win32ax.inc' ; import data in the same section invoke MessageBox,NULL,msgBoxText,msgBoxCaption,MB_OK ...

Побайтовый вывод файла FASM
Пытаюсь ввести в консоль файл в шестнадцатеричном виде, но происходит ошибка при выполнении. format PE console 4.0 include 'win32a.inc' xor ebx, ebx ; invoke CreateFile,\ ...

FASM Вывод адреса на консоль
Пытаюсь на консоль вывести адрес fin: invoke printf, не робит - как правильно надо? format PE console 4.0 entry start include 'win32a.inc' section '.data' data readable fin ...

FASM ГСЧ на макросах Всем привет. Понадобилось заюзать ГСЧ посредством макросов, чтобы каждый раз на стадии компиляции, использовалось уникальное значение. Учитывая семантику препроцессора (там чёрт ногу сломит),... https://www.cyberforum.ru/ fasm/thread1213146.html

FASM Создание окна на fasm https://www.cyberforum.ru/ fasm/thread1209394.html
Всем привет. Только что начал изучать ассемблер fasm. Возник первый вопрос: как создать окно? Прошу не просто дать мне код, а ещё объяснить что значит. Заранее благодарен

FASM Вызываем функции из clib (библиотека Си) в DOS
Вобщем, сбылась мечта идиота. Теперь, нежели писать свой ввод/вывод(особенно всегда напрягал ввод/вывод вещественных чисел на экран), можно воспользоваться стандартными ф-циями из библиотеки языка...

Организовать вычисления по формуле FASM
привет, всем активным участникам этого чудесного форума!!! помогите, пожалуйста, написать программу на Fasm Assembler. задание: Создать программу на языке Ассемблер, что позволяет организовать...

FASM Как сделать выход по ESC org 100h old dw 0 jmp start number dw 0 c dw 0 start: xor ax,ax mov es,ax cli https://www.cyberforum.ru/ fasm/thread1161834.html

FASM Получение CLSID image/png https://www.cyberforum.ru/ fasm/ thread1160365.html
Всем ку! int GetEncoderClsid(const WCHAR* format, CLSID* pClsid) { UINT num = 0; // number of image encoders UINT size = 0; // size of the image encoder array in bytes...

FASM Вывод трех строк в один MessageBox Здравствуйте, помогите, пожалуйста, с такой проблемой: не могу вывести 3 строки (Год+Месяц+День) в один MessageBox Вот такой код: format PE GUI 4.0 entry start include 'win32ax.inc' include... https://www.cyberforum.ru/ fasm/thread1142589.html

MikI
Ушел с форума
13987 / 7000 / **813**
Регистрация: 11.11.2010
Сообщений: 12,592

10.08.2014, 08:44 **[TC]** **0**

Мануал по flat assembler

10.08.2014, 08:44. **Просмотров** 99777. **Ответов** 50
Метки (Все метки)

Ответ

2.1.1. Инструкции перемещения данных

- "mov" переносит байт, слово или двойное слово из операнда-источника в операнд-адресат. Этот мнемоник может передавать данные между регистрами общего назначения, из этих регистров в память, обратно, но не может перемещать данные из памяти в память. Он также может передавать непосредственное значение в регистр общего назначения или в память, сегментный регистр в регистр общего назначения или в память, регистр общего назначения в сегментный регистр или в память, контрольный или отладочный регистр в регистр общего назначения и назад. "mov" может быть ассемблирована только если размер операнда-источника и размер операнда-адресата совпадают. Ниже приведены примеры каждой из перечисленных комбинаций:

Assembler Выделить код

```
1       mov bx,ax       ; поместить содержимое регистра общего назначения AX в регистр общего назначения BX
2       mov [char],al   ; поместить содержимое регистра общего назначения AL в ячейку памяти размером 8 бит
3       mov bl,[char]   ; поместить содержимое ячейки памяти в регистр общего назначения BL
4       mov dl,32       ; поместить непосредственное значение в регистр общего назначения DL
5       mov [char],32   ; поместить непосредственное значение в ячейку памяти
6       mov ax,ds       ; поместить содержимое сегментного регистра DS в регистр общего назначения AX
7       mov [bx],ds     ; поместить содержимое сегментного регистра в ячейку памяти адресуемую через регистр BX
8       mov ds,ax       ; поместить содержимое регистра общего назначения AX в сегментный регистр DS
9       mov ds,[bx]     ; поместить содержимое ячейки памяти, адресуемую через регистр BX, в сегментный регистр
10      mov eax,cr0     ; поместить содержимое контрольного регистра CR0 в регистр общего назначения EAX
11      mov cr3,ebx     ; поместить содержимое регистра общего назначения EBX в контрольный регистр CR3
```

- "xchg" меняет местами значения двух операндов. Инструкция может поменять два байтовых операнда, операнды размером в слово и размером в двойное слово. Порядок операндов не важен. В их роли могут выступать два регистра общего назначения либо регистр общего назначения и адрес в памяти. Например:

Assembler Выделить код

```
1       xchg ax,bx      ; обмен содержимым регистров общего назначения AX и BX
2       xchg al,[char]  ; обмен содержимым регистра общего назначения AL и ячейки памяти
```

- "push" уменьшает значение указателя стекового фрейма (регистр ESP), потом переводит операнд на верх стека, на который указывает ESP. Операндом может быть память, регистр общего назначения, сегментный регистр или непосредственное значение размером в слово или двойное слово. Если операнд - это непосредственное значение

и его размер не определен, то в 16-битном режиме по умолчанию он обрабатывается как слово, а в 32-битном режиме как двойное слово. Мнемоники "pushw" и "pushd" - это варианты этой инструкции, которые сохраняют соответственно слова и двойные слова. Если на одной строке содержится несколько операндов (разделенных пробелами, а не запятыми), компилятор проассемблирует цепь инструкций "push" с этими операндами.

Вот примеры с одиночными операндами:

Assembler — Выделить код

```
1       push ax         ; сохранить содержимое регистра общего назначения AX в стеке
2       push es         ; сохранить содержимое сегментного регистра в стеке
3       pushw [bx]      ; сохранить содержимое ячейки памяти в стеке
4       push 1000h      ; поместить в стек непосредственное значение
```

- Инструкция "pusha" сохраняет в стек содержимое восьми регистров общего назначения. У неё нет операндов. Существует две версии этой инструкции: 16-битная и 32-битная. Ассемблер автоматически генерирует версию, соответствующую текущему режиму, но, используя мнемоники "pushaw" или "pushad",это можно изменить для того, чтобы всегда получать, соответственно, 16- или 32-битную версию. 16-битная версия этой инструкции сохраняет регистры общего назначения в таком порядке: AX, CX, DX, BX, значение регистра SP перед тем, как был сохранен AX, далее BP, SI и DI. 32-битная версия сохраняет эквивалентные 32-битные регистры в том же порядке.
- "pop" переводит слово или двойное слово из текущей верхушки стека в операнд-адресат и после уменьшает ESP на указатель на новую верхушку стека. Операндом может служить память, регистр общего назначения или сегментный регистр. Мнемоники "popw" и "popd" - это варианты этой инструкции, восстанавливающие соответственно слова и двойные слова. Если на одной строке содержится несколько операндов, разделенных пробелами, компилятор ассемблирует цепочку инструкций с этими операндами.

Assembler — Выделить код

```
1       pop bx          ; восстанавливает регистр общего назначения
2       pop ds          ; восстанавливает сегментный регистр
3       popw [si]       ; восстанавливает память
```

- "popa" восстанавливает регистры, сохраненные в стек инструкцией "pusha", кроме сохраненного значения SP (или ESP)? который будет проигнорирован. У этой инструкции нет операндов. Чтобы ассемблировать 16 или 32-битную версию этой инструкции, используйте мнемоники "popaw" или "popad".

Вернуться к обсуждению:
Мануал по flat assembler

Следующий ответ

1

IT_Exp
Эксперт
87844 / 49110 / **22898**
Регистрация: 17.06.2006
Сообщений: 92,604

10.08.2014, 08:44

Заказываю контрольные, курсовые, дипломные и любые другие студенческие работы здесь.

Ошибка в flat assembler
начал изучать ассемблер сталкнулся с такой проблемой: перепечатываю пример из книги: org...

flat assembler массив
У меня есть задание "Упорядочить по убыванию элементы каждого столбцу матрицы" Числа произвольные...

Массив в Flat Assembler
Всем добрый день! Подскажите, почему не работает массив в Flat Assembler? org 100h jmp start...

Как использовать Flat Assembler в Free Pascal?
Я недавно хотел разработать так ради прикола мини ОС с использованием в связке Free Pasal и Flat...

0

Powered by vBulletin® Version 3.8.9
Copyright ©2000 - 2021, vBulletin Solutions, Inc.