



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 ...

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 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

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 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...

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 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 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...

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 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

21.08.2014, 09:03 [ТС]

Мануал по flat assembler

21.08.2014, 09:03. Просмотров 99777. Ответов 50

Метки (Все метки)

Ответ

2.1.14 Инструкции MMX

Инструкции MMX оперируют со сжатыми целочисленными типами и используют регистры MMX, которыми являются нижние 64-битные части 80-битных регистров FPU. Поэтому инструкции MMX не могут использоваться в одно и то же время с инструкциями FPU. Они могут оперировать со сжатыми байтами (восемь 8-битных целых чисел), сжатыми словами (четыре 16-битных целых чисел) или сжатыми двойными словами (два 32-битных целых числа). Использование сжатых форматов позволяет совершать операции одновременно над многими данными.

- "movq" копирует четверное слово из операнда-источника в операнд-адресат. По крайней мере одним из операндов должен являться регистр MMX, вторым может быть либо регистр MMX, либо 64-битное расположение в памяти.

Assembler

[Выделить код](#)

```
1 movq mm0,mm1 ; копирует четверное слово из регистра в регистр
2 movq mm2,[ebx] ; копирует четверное слово из памяти в регистр
```

- "movd" копирует двойное слово из операнда-источника в операнд-адресат. Одним из операндов должен быть регистр MMX, вторым может быть регистр общего назначения либо 32-битное расположение в памяти. Используется только нижнее двойное слово регистра MMX.

Все основные операции MMX имеют два операнда, где операндом-адресатом должен быть регистр MMX, а операндом-источником может быть либо регистр MMX, либо 64-битное расположение в памяти. Операция совершается на соответствующих элементах данных источника и адресата и сохраняется в элементах данных адресата. "paddb", "paddw" и "paddq" совершают сложение сжатых байтов, сжатых слов и сжатых двойных слов. "psubb", "psubw" и "psubd" совершают вычитание соответствующих типов. "paddsb", "paddsw", "psubsb" и "psubsw" совершают сложение или вычитание сжатых байтов или сжатых слов со знаковым насыщением. "paddusw", "paddusw", "psubusw" - это аналоги, но без знакового насыщения. "pmulhw" и "pmullw" совершают знаковое умножение сжатых слов и сохраняют верхние или нижние слова результатов в операнде-адресате. "pmaddwd" совершает умножение сжатых слов и складывает четыре промежуточных продукта в виде двойных слов в парах, чтобы получить результат в виде сжатых двойных слов. "pand", "por" и "pxor" совершают логические операции над четверными словами, "pandl" также производит логическое отрицание перед операцией "and". "rptreqb", "rptreqw" и "rptreqd" сравнивают на эквивалентность сжатые байты, сжатые слова или сжатые двойные слова. Если пара элементов данных эквивалентна, то соответствующий элемент данных операнда-адресата покрывается единичными битами, иначе нулевыми. "psrpgtb", "psrpgtw" и "psrpgtd" совершают похожую операцию, но они проверяют, больше ли элементы данных в операнде-адресате, чем соответствующие элементы данных в операнде-источнике. "packsswb" конвертирует сжатые знаковые слова в сжатые знаковые байты, "packssdw" конвертирует сжатые знаковые двойные слова в сжатые знаковые слова, используя насыщение, чтобы удовлетворить условиям переполнения. "packuswb" конвертирует сжатые знаковые слова в сжатые беззнаковые байты.

Конвертированные элементы данных из операнда-источника сохраняются в нижней части операнда-адресата, тогда как конвертированные элементы данных операнда-адресата сохраняются в его верхней части. "punprckhbw", "punprckhwd" и "punprckhdq" чередуют элементы данных из верхних частей источника и адресата и сохраняют результат в операнд-адресат. "punprcklbw", "punprcklwd" и "punprckldq" совершают те же операции, но с нижними частями операндов.

Assembler

[Выделить код](#)

```
1    paddsb mm0,[esi] ; складывает сжатые байты со знаковым насыщением
2    pcmpsqw mm3,mm7 ; проверяет сжатые слова на эквивалентность
```

- "psllw", "pslld" и "psllq" совершают логический сдвиг влево сжатых слов, сжатых двойных слов или одиночных четверных слов в операнде-адресате, на число битов, указанное в операнде-источнике. "psrlw", "psrld" и "psrlq" совершают логический сдвиг вправо сжатых слов, сжатых двойных слов или одиночных четверных слов. "psraw" и "psrad" совершают арифметический сдвиг сжатых слов или двойных слов. Операндом-адресатом должен быть регистр MMX, а операндом-источником может быть регистр MMX, 64-битное расположение в памяти или 8-битное непосредственное значение.

Assembler

[Выделить код](#)

```
1    psllw mm2,mm4 ; сдвигает слова влево логически
2    psrad mm4,[ebx] ; сдвигает двойные слова вправо арифметически
```

- "emms" делает регистры FPU используемыми для инструкций FPU. Эта инструкция должна быть применена перед использованием инструкций FPU, если в ход пускались инструкции MMX.

[Вернуться к обсуждению:](#)

[Мануал по flat assembler](#)

[Следующий ответ](#)

0

IT_Exp

Эксперт

87844 / 49110 / **22898**

Регистрация: 17.06.2006

Сообщений: 92,604

21.08.2014, 09:03

Заказываю контрольные, курсовые, дипломные и любые другие студенческие работы [здесь](#).

[Ошибка в flat assembler](#)

начал изучать ассемблер столкнулся с такой проблемой: перепечатаваю пример из книги: org...

[flat assembler массив](#)

У меня есть задание "Упорядочить по убыванию элементы каждого столбца 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 Pascal и Flat...

0