

\* Про .bss идущи - я ошибся  
 когда сказал, что зарезервиро-  
 ванная моя память будет разделяться  
 разноразмерного бинарного файла - это  
 не так.

— Пример type-extension.s -  
 \* Формат кода посмотрите: movzx

- movzx rax, byte[byte-var]
- Команды означают данные по  
 адресу byte-var остальные  
 дополнительные
- указание разноразмерных  
 (спецификатор "byte") при  
 использовании барранта  
 операции может  $\rightarrow$  решетка -  
 - разделитель
- movzx rax, dx
- размер обоих операндов избирается  
 имену указывает не нужно

-  $y \text{ movzx dest, src :}$   
dest - базовый регистр  
src - регистр или память (обязательно сдвиги на одинаковом масштабе)

- \* Знаковое расширение:  $\text{movsx}$   
Допускает не знаковая часть базового регистра знаком быть dest src.  
- Все базовые, за исключением сдвиговых аналогов  $\text{movzx}$ .

---

\* Базовое представление и значение операндов:  
{Пример math-operations. }

- add op1, op2 - сложение
- \* Что может быть на месте op1 и op2?
  - op1 - регистр, память
  - op2 - константа, регистр, память,

! Но, оп1 и оп2 не могут одновременно быть равны.

Почему?

[Докторит "objdump -d" - дает следующий

1) Рассмотрим грузы - add  
старых прошлых состояний до новых  
крайне сильной аргументации

2) Но если операнды memory  $\rightarrow$   
memory  $\rightarrow$  в лобине не  
load memory  $\rightarrow$  reg  $\rightarrow$  store memory,  
то они этого не могут сделать.  
но если не пересекут эти обработки  
в момент, иначе им могут  
перенести.

• sub op1, op2 — вычитание  
- Вычитание на op1 и op2  
аналогично add

• neg op1 — отрицание  
- Унарное операнде, т.е. требует  
единого слова операнда

- OPT - пример для настав

Переносение [пример flags.s]

mov eax, 0xffffffff

add eax, 1

- Что в eax?

- В eax → 0, ошибка, что произошло переносение.

- А как мы не знаем значение eax? Как можно, произошло не переносение?
- Ответ: яз. Del 2000 используется регистр EFLAGS.

[ показать картинку, показать в отладчике.

Он же синхронизирует данные:

- CF - производит бессвязное переносение, т.к. возможен перенос из старшего регистра

- OF — ~~запоминает результат~~, т.е. избранный результат не может быть преобразован дальше пока нет новое задание.
- ZF — выставляет единицу если результат равен нулю
- SF — выставляет единицу если результат отрицательный

Sleeping-test:

|   | ZF | SF | OF | CF |
|---|----|----|----|----|
| 1 | 0  | 1  | 1  | 0  |
| 2 | 1  | 0  | 0  | 1  |
| 3 | 0  | 0  | 1  | 0  |
| 4 | 0  | 1  | 0  | 1  |

Группа команд для симметрического процессора:

set z - ZF

sets - ST

setc - CF

seto - OF

---

\* Установка: mul [mul-s]  
mul op

- Установка eax на op, результат помещается в EDX и EAX: первая половина в EAX, вторая половина в EDX  
[Назначение пример 1]
- В зависимости от разряда операнда, mul может возвращать результат:

| <u>Разряд op</u> | <u>Несколько битов младшей</u> | <u>Результат</u> |
|------------------|--------------------------------|------------------|
| 8 Sat            | AL                             | AX               |
| 16 Sat           | AX                             | DX: AX           |
| 32 Sat           | EAX                            | EDX: EAX         |

[Пример 2, с 8 битами операнда]

- mul может брать любые только  
один CF и OF:

$CF = 1 \text{ и } OF = 1$  — если старшая  
разность неизвестна  $\neq 0$

$CF = 0 \text{ и } OF = 0$  — если старшая  
разность неизвестна  $= 0$

- Для остальных разностей побеждены  
не определены (пример из § 15)

A Знаковое умножение: imul  
[примеры I, II из iMul.S]

Имеет три формы:

I: imul op

II: imul op1, op2 — пример 3

III: imul op1, op2, op3 — пример 4

- Варианты: II и III формы не  
используют EDX, т.е. при  
перемещении байтами регистров  
отрабатывается. Всё это  $\Rightarrow$   
 $\Rightarrow$  II и III формой не пользуются

- Деление: [пример div.s]
  - div op
- Делит EDX: EAX на op.
  - В EAX может остаток, в EDX - остаток.
- Результат операции будет  
таким же, как у mul.

[Пример 1, разделяет значение на 0]

- \* Знаковое деление: i div
  - Т.к. требуется знаковое разделять значение EDX: EAX, остаток можно знаковое разделять:
- cbw       $8 \rightarrow 16$
- cwd       $16 \rightarrow 32$
- cdq       $32 \rightarrow 64$
- \* Важно: div / idiv не берут account никаких flags из ядра в б/у EFLAGS

Допустимо ли это для OF:

- OF будет обнулена, если результат нацело делится на 2 и не имеет остатка
- [. INT-MIN, INT-MAX]

- Пример 1 из flags.s

```
mov eax, 0x7fffff  
add eax, 1
```

недопустимое значение  
знаков  
указатель

$$0x7fffff = 2^{31} - 1$$

$$\Rightarrow 2^{31} - 1 + 1 = 2^{31} \notin [-2^{31}; 2^{31} - 1]$$

$$\Rightarrow OF = 1$$

- Пример 2:

```
mov eax, 0x8fffff  
add eax, 1
```

отрицательное  
указатель

$$\Rightarrow 0x8fffff = -1$$

указатель

$$\Rightarrow (-1) + 1 = 0 \in [-2^{31}; 2^{31} - 1]$$

$$\Rightarrow OF = 0$$

- Пример 3:

mov eax, 0x80000000

sub eax, 1

†

$$0x80000000 = -2^{31}$$

отрицательное  
число

$$\Rightarrow -2^{31} - 1 \notin [-2^{31}; 2^{31} - 1]$$

Пример 4:

mov eax, 0

sub eax, 1

$$\Rightarrow 0 - 1 = -1 \in [-2^{31}, 2^{31}] =$$

$$\Rightarrow OF = 0$$