Языки программирования (Asm)

Занятие №7. Строковые команды. Команды работы с флагами.

МИРЭА - РТУ, БК №252

Москва 2020

CLD / STD

Как мы помним, в регистре флагов EFLAGS находится флаг DF (direction flag), отвечающий за направление обработки строк.
Команда CLD (clear direction flag) сбрасывает флаг направления (задает значение DF = 0).

```
1 cld ; Команда не имеет операндов
```

Если нужно выставить флаг направления (задать значение DF = 1), используется команда STD (set direction flag).
Данная команда также не имеет операндов.

```
1 std
```

Строковые команды

Центральный процессор имеет специальные команды для обработки строк. В языках ассемблера они представлены следующими командами.

1. MOVS
2. LODS
3. STOS
4. CMPS
5. SCAS

В NASM данные команды могут использоваться только с одним из следующих окончаний

1. B - для работы с байтом.
2. W - для работы со словами (слово = 2 байтам).
3. D - для работы с двойными словами (4 байтами).
4. Q - для работы с 8 байтами.

MOVS - Move string

Команда MOVSB побайтово копирует строку, на которую указывает адрес в регистре ESI в строку, на которую указывает значение регистра EDI.

```
1 cld       ; Сбросить флаг DF
2 lea       esi, [str1] ; ESI указывает на исходную строку
3 lea       edi, [str2] ; EDI указывает на результирующую строку
4 movsb     ; Копирует первый байт
5 movsb     ; Копирует второй байт
```

MOVSB копирует 1 байт
MOVSW копирует 2 байта
MOVSD копирует 4 байта
MOVSQ копирует 8 байт

LODS - Load string

Команда LODS копирует данные из памяти по адресу, сохраненному в регистре ESI в регистр AL/AX/EAX/RAX.

```
1 lea esi, [str1]    ; Передаем адрес строки
2 lodsb     ; Загрузить один байт строки в AL
3 mov [symbol], al
4 mov eax, 4
5 mov ebx, 1
6 mov ecx, symbol
7 mov edx, 1
8 int 0x80
```

LODSB копирует 1 байт в регистр AL
LODSW копирует 2 байта в регистр AX
LODSD копирует 4 байта в регистр EAX
LODSQ копирует 8 байт в регистр RAX

REP, REPE, REPNE, REPZ, REPNZ

Строковые команды зачастую используются вместе с одной из команд REP, которые нужны для повторения другой команды нужное число раз.
Например, команда **REPE CMPSB** найдет пару неравных байт в строках по адресам ESI и EDI. Размер строк задается значением регистра ECX. Команда завершится тогда, когда найдутся различные байты, либо когда значение ECX достигнет нуля. Порядковый номер отличающегося байта можно получить увеличив значение регистра EDI на 1

```
1 inc  edi
2 mov  eax, edi
3 add  eax, 0x30
```

REP, REPE, REPNE, REPZ, REPNZ

Команда REPNE выполняет идущую за ней команду пока либо значение регистра ECX не станет равно 0, либо ZF не станет равен 1. При каждом выполнении команды REPNE значение регистра ECX автоматически уменьшается на единицу.

```
1 repne lodsb
```

Повторение, выполняемое командами REPE/REPZ завершится в случае, если значение регистра ECX достигло нуля либо ZF = 0. При каждом выполнении этих команд значение регистра ECX также автоматически уменьшается на единицу.

```
1 repnz movsb
```

STOS - Store String

```
1 mov ecx, 10          ; Счетчик для команды repne
2 mov al, '#'          ; Значение
3 lea edi, [str1]      ; Адрес строки, в которую будут записаны символы
4 repne stosb
```

Команда STOS заполняет указанную область памяти значением, находящимся в регистре AL (AX, EAX или RAX). Регистр EDI должен указывать на нужную область памяти.

STOSB копирует 1 байт из регистра AL
STOSW копирует 2 байта из регистра AX
STOSD копирует 4 байта из регистра EAX
STOSQ копирует 8 байт из регистра RAX

CMPS - Compare strings

Команда CMPS выполняет сравнение двух строк. CMPSB выполняет побайтовое сравнение строк. Эту команду можно использовать следующим образом.

```
1  mov esi, str1
2  mov edi, str2
3  mov ecx, lens2
4  cld
5  repe  cmpsb
6  jecxz  equal
7
8  mov eax, 4
9  mov ebx, 1
10 mov ecx, msg_neq
11 mov edx, len_neq
12 int 0x80
13 jmp exit
14
15 equal:
```

SCAS - Scan string

Команда SCAS выполняет поиск символа (или блока из нескольких байт) в определенной области памяти.

```
1 mov ecx, [len]
2 mov edi, my_string
3 mov al, 'e'
4 cld
5 repne scasb
6 je found ; Если символ найден, перейти по метке
```

SCASB сравнивает по 1 байту
SCASW сравнивает по 2 байта
SCASD сравнивает по 4 байта
SCASQ сравнивает по 8 байт
Важно помнить, что данный поиск не универсален. Приведем пример.

SCAS - Scan string

Допустим, мы хотим найти в строке последовательность "1234"

```
1 mov ecx, [len]
2 mov edi, my_string
3 mov eax, '1234'
4 cld
5 repne scasd
6 je found ; Если символ найден, перейти по метке
```

Если исходная строка была "abcd1234qw... то команда SCASD на втором шаге найдет совпадение, т.к. она считывает блоки по 4 байта. А если нужная строка будет находиться со смещением, например "asd1234vbnn то программа не найдет ее. Будут сравниваться блоки "asd1"и "234v" , они оба не совпадают с заданной подстрокой.