**Строковые команды**

**Строка** – последовательность (массив) символов

Строка (как и любой массив) имеет 2 основных характеристики:

* Адрес (где находится)
* Длина строки

Подходы:

* Pascal-строки (length – prefixed string)
* C-строки

**Счетчик ссылок**

Подсчет ссылок широко используется в автоматическом управлении памятью

Значение счетчика ссылок – кол-во переменных, ссылающихся на объект (строку, массив).

* Если = 0, память, занятая объектом, освобождается

**Unicode**

Unicode – стандарт кодирования символов, позволяющих представить знаки почти всех письменных языков.

Существуют различные способы представления (Unicode Transformation Formats):

* UTF-8
* UTF-16 (UTF-16LE, UTF-16BE)
* URF-32 (UTF-32LE, UTF-32BE)
* UTF-7 (Фуфло, но можно почитать)

Чем отличаются USC-2 и UTF-16? (Почитать, подумать)

Что такое BOM-байты? (Почитать)

**Строковые команды**:

* LODS
* STOS
* MOVS
* SCAS
* CMPS
* INS
* OUTS

Могут работать с данными размером 1,2,4 и 8 байт

* LODSB/W/D/Q
* STOSB/W/D/Q
* MOVSB/W/D/Q
* SCASB/W/D/Q
* CMPSB/W/D/Q
* INSB/W/D/Q
* OUTSB/W/D/Q

**LODS**

* Загружает элемент из памяти в регистр.
  + **LODSB:** mov al, [DS:SI]
  + **LODSW**: mov ax, [DS:SI]
* Загружает:
  + Всегда в AL/AX
  + Всегда из памяти по адресу DS:SI
* Изменяет **SI**
* ~~И еще кое-что~~ :}

**STOS**

* Записывает элемент из регистра в память
  + **STOSB:** mov [ES: DI], AL
  + **STOSW:** mov [ES:DI], AX
* Записывает:
  + Всегда из AL/AX
  + Всегда в память по адресу ES:DI
* Изменяет **DI**
* ~~И еще кое что~~ :{

**MOVS**

* Копирует элементы в памяти
  + **MOVSB:** mov [ES:DI], [DS:SI]
  + **MOVSW:** mov [ES:DI], [DS:SI]
* Копирует:
  + Всегда из памяти по адресу DS:SI
  + Всегда в память по адресу ES:DI
* Изменяет **SI** и **DI**
* ~~И еще кое-что~~ ಠ\_ಠ

**SCAS**

* Сравнивает значение регистра с элементом в памяти
  + SCASB: cmp AL, [ES:DI]
  + SCASW: cmp AX, [ES:DI]
* Сравнивает:
  + Всегда значение регистра AL/AX
  + Всегда с элементом по адресу ES:DI
* Изменяет **DI**
* ~~И еще кое-что~~ (¬\_¬)

**CMPS**

* Сравнивает значение элемента в памяти с элементом в памяти
  + CMPSB: cmp [DS:SI], [ES:DI]
  + CMPSW: cmp [DS:SI], [ES:DI]
* Сравнивает:
  + Всегда элементы по адресу DS:SI
  + Всегда с элементом по адресу ES:DI
* Изменяет **SI** и **DI**
* ~~И еще кое-что~~ ʘ\_ʘ

**Кроме копирования/сравнения все строковые команды изменяют значения регистров SI и(или) DI**

* **LODS:** only SI
* **STOS**: only DI
* **MOVS**: SI and DI
* **SCAS**: only DI
* **CMPS**: SI and DI
* Значение SI и(или) DI изменяется в большую или меньшую сторону в зависимости от флага DF:
  + DF = 0 => увеличивается
  + DF = 1 => уменьшается
* Значение изменяется на количество байт, с которыми работает команда!

|  |  |  |
| --- | --- | --- |
| **LODS:** | | |
| mov reg, [DS:SI] | SI <- SI ± n |  |
| **STOS** | | |
| mov [ES:DI], reg |  | DI <- DI ± n |
| **MOVS** | | |
| Mov [ES:DI], [DS:SI] | SI <- SI ± n | DI <- DI ± n |
| **SCAS** | | |
| cmp reg, [ES:DI] |  | DI <- DI ± n |
| **CMPS** | | |
| Cmp [DS:SI], ES:DI] | SI <- SI ± n | DI <- DI ± n |

**Префиксы повторения**

* Перед строковыми команды могут указываться **префиксы**

Перед LODS, STOS, MOVS

* + REP
  + REPE/REPZ

Перед SCAS и CMPS

* + REPNE/REPNZ
* Если перед строковой командой указан REP-префикс, команда будет повторяться несколько раз

|  |  |  |
| --- | --- | --- |
| **ПРЕФИКС** | **УСЛОВИЕ 1** | **УСЛОВИЕ 2** |
| REP | CX=0 | No |
| REPE/REPZ | CX=0 | ZF = 0 |
| REPNE/REPNZ | CX=0 | ZF = 1 |

*i: (Пока равно/не равно, не больше CX элементов)*

**С помощью строковых команд можно реализовать работу:**

* Со строками
* С массивами скалярных типов