

# Команды обработки стека



Стек работает по принципу **LIFO**.

Данные, помещенные в стек последними, будут первыми «вытолкнуты» из стека. В РС совместимых компьютерах нет аппаратного стека, поэтому данные стека

Хранятся в памяти. Вершина стека представлена парой **SS:ESP**

—  
Сегмент стека (**StackSegment**) и указатель вершины стека (**StackPointer**).

Стек растет в памяти «вниз», то есть новая порция данных записывается по Меньшему адресу. Впрочем, точный адрес данных внутри стека не имеет для

Нас значения, потому что любая операция над стеком имеет дело с его вершиной, на которую всегда указывает регистр (**ESP**). Стек может содержать 32-битные данные.

Микропроцессор имеет две команды для работы со стекком—**PUSH** и **POP**.

# Команды PUSH и POP : втолкнуть и вытолкнуть

Команда PUSH позволяет поместить в стек содержимое любого регистра или ячейки памяти. Формат команды следующий:

PUSH o1

Примериспользования:

push ea x - поместит EA X в стек

Мы можем сами реализовать команду PUSH с помощью следующей пары команд:

```
sub esp, 4 ;уменьшаем ESP на 4 (EA X—4-байтный регистр )
```

```
mov [ss:esp], ea x ;сохраняем EA X в стеке
```

В общем виде (с использованием оператора sizeof, «позаимствованного» из языков высокого уровня) команда push o1 может быть записана на псевдо-языке так:

```
ESP=ESP-sizeof(o1 )
```

Другая команда, POP, записывает в свой операнд значение вершины стека (последнее сохраненное в стеке значение). Тип операнда должен быть таким же, как у инструкции PUSH (другими словами, если вы поместили в стек 32-разрядный регистр, извлечение из стека должно происходить тоже в 32-разрядный регистр). Команду POP можно реализовать с помощью команд MOV и ADD:

```
mov eax,[ss:esp ] ;помещаем в EA X вершину стека
```

```
add esp, 4      ;"удаляем "последнее значение  
               ;тип adword в стеке
```

Рассмотрим несколько примеров:

```
push ea x ;сохранить значение регистра EA X в стеке
```

```
push es i ;сохранить значение регистра ESI в стеке
```

```
pop eax ;извлечь данные из стека в EA X
```

```
pop es i ;извлечь данные из стека в ESI
```

В результате выполнения этих команд мы поменяем местами значение регистров EAX и ESI : с начала помещаем в стек значение EAX, затем — ESI, после

этого извлекаем из стека последнее сохраненное значение (бывшее значение

```
1  %include "io.inc" ; STACK
2
3  section .text
4  global CMAIN
5  CMAIN:
6      mov ebp, esp
7
8      mov eax, 5      ; ВВОДИМ в eax 5
9      mov ebx, 6      ; ВВОДИМ в ebx 6
10     add eax, ebx    ; СЛОЖЕНИЕ
11     push eax        ; вставка в стек
12     mov edx, [esp]
13     pop ecx         ; вставка из стека
14     xor eax, eax
15     ret
```