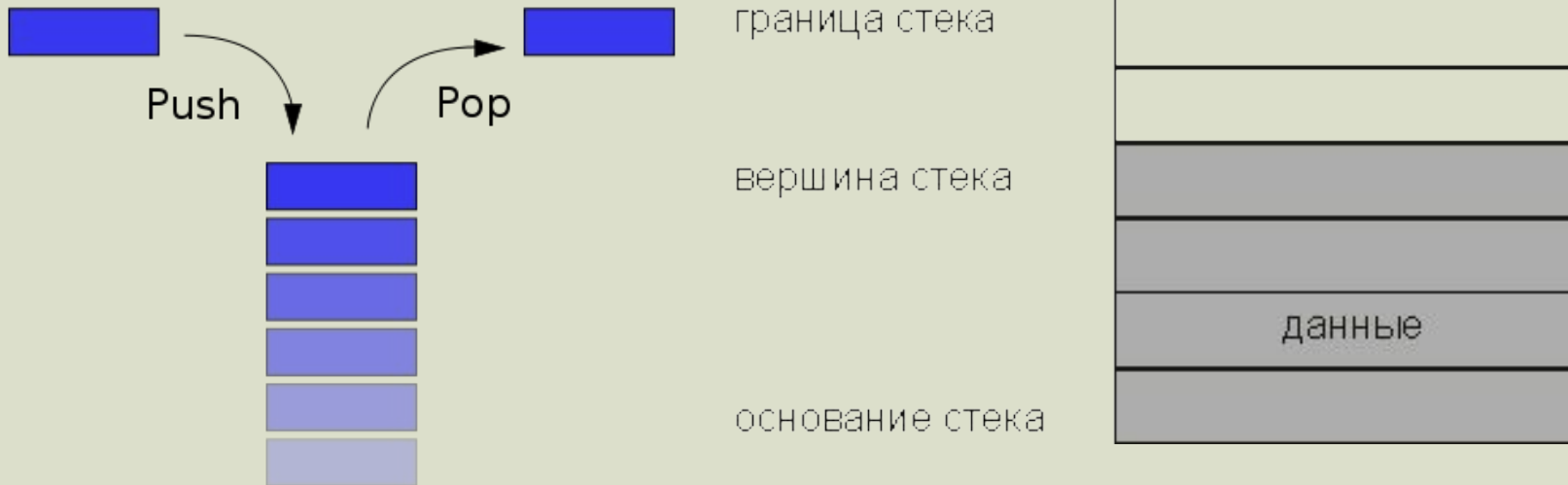


# Стек

**Стек** – это определенный динамический способ хранения данных, при котором в каждый момент времени доступ возможен только к одному из элементов, а именно к тому, который был занесен в стек последним.

**Стек** — структура данных с методом доступа к элементам LIFO (англ. Last In — First Out, «последним пришел — первым вышел»).



# Стек

Регистры x86:

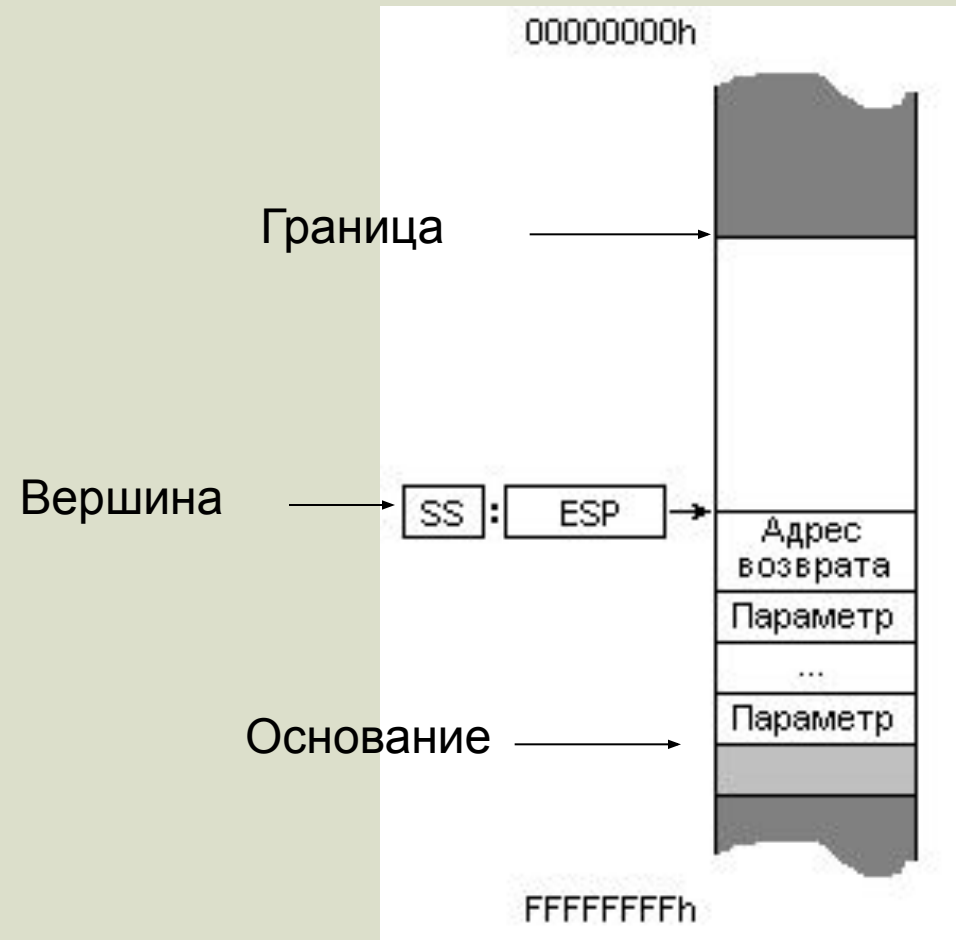
**SS** – хранит границу стека  
(начало сегмента стека)

**ESP** – хранит вершину стека  
(адрес элемента, записанного  
в стек последним)

**EBP** – регистр для  
манипуляций со стеком.

Позволяет получить доступ к  
предпоследнему,  
предпредпоследнему, и т.д.  
элементу.

## Реализация стека на уровне ЦП



# Стек

## Реализация стека на уровне ЦП

Команды для работы со стеком:

**PUSH** – добавить данные в стек. Команда записывает данные в стек и уменьшает регистр ESP.

**POP** – извлечь данные из стека. Команда извлекает данные и увеличивает ESP.

```
PUSH BX
```

```
PUSH CX
```

```
<здесь любые манипуляции с BX и CX>
```

```
POP CX
```

```
POP BX
```

# Стек

## Реализация стека на уровне ЦП

Регистры x86:

**SS** – хранит границу стека (начало сегмента стека)

**ESP** – хранит вершину стека (адрес элемента, записанного в стек последним)

**EBP** – регистр для манипуляций со стеком. Позволяет получить доступ к предпоследнему, предпредпоследнему, и т.д. элементу.

;код для 32-битного процессора

**mov EBP,ESP** ;получаем доступ к последнему

**inc [EBP]** ;можем изменить последний

**add EBP,4** ;получаем доступ к предпоследнему

**inc [EBP]** ;можем изменить предпоследний

# Стек

```
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int x=5, y=7;
    //Поменять местами значения переменных
    __asm
    {
        push x
        push y
        pop x; Извлекаем 7
        pop y; Извлекаем 5
    }
    printf("x=%d\ny=%d\n",x,y);
    return 0;
}
```

Примеры работы со стеком

# Стек

```
int _tmain(int argc, _TCHAR* argv[])
{
    int x;
    //Получить доступ к элементу внутри стека
    __asm
    {
        push 3
        push 5
        push 7
        push 9
        mov eax, [esp + 8]; В eax записываем 5
        mov x, eax
        add esp, 16; Удаляем 4 элемента
    }
    printf("x=%d\n",x);
    return 0;
}
```

Примеры работы со стеком