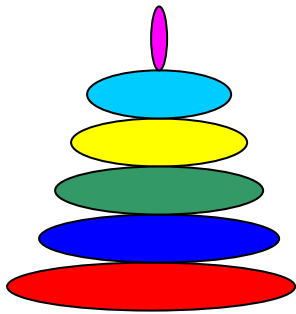


Лекция №7

Тема "Стековая память"

Стек - специальный сегмент памяти, который используется для временного хранения произвольных данных.



Стек часто называют структурой **LIFO** (Last In - First Out – последний вошел - первый вышел).

Принцип работы стека:

элемент, записанный в стек последним, считывается из него первым.

Основные функции, которые выполняет стек:

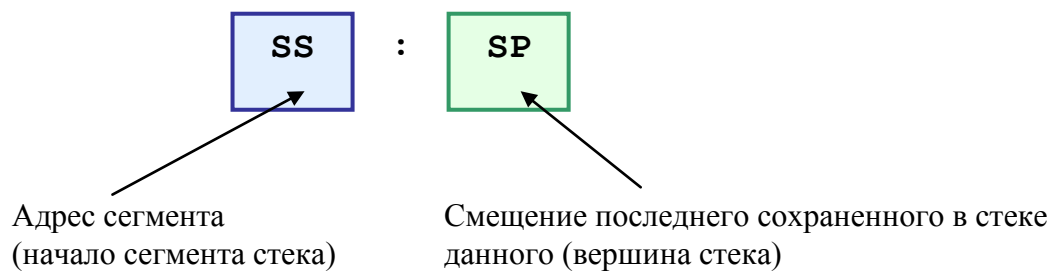
- временное сохранение значений регистров;
- хранение адресов возврата из подпрограмм;
- хранение динамических переменных.

Требования к памяти, отводимой для стека:

- размер стека не должен превышать 64 Кбайт;
- адрес должен быть кратным 16.

Самый простой способ задания стека – это использование директивы **STACK**.

Адрес доступной ячейки стека



Запись в стек

Команда **push** («протолкнуть») записывает в стек свой операнд.

| |
|-----------------------------|
| push <i>ИСТОЧНИК</i> |
|-----------------------------|

| Источник | | |
|----------|----|-----|
| i16 | sr | m16 |

Команда **push** выполняет два действия:

1. Уменьшает на 2 байта значение SP.
2. Заданное значение регистра записывает по адресу [SS : SP].

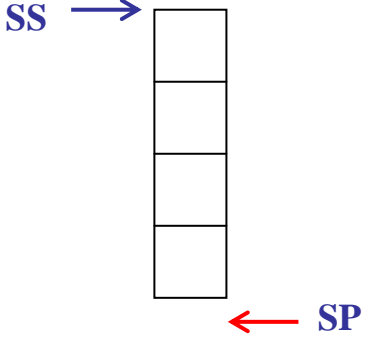
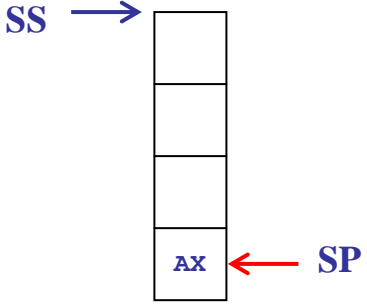
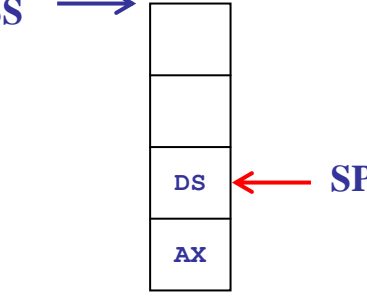
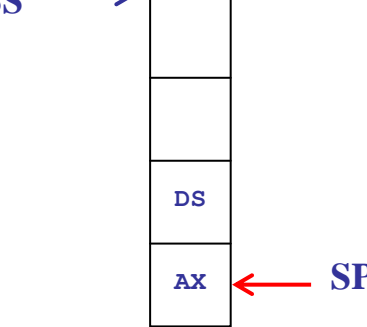
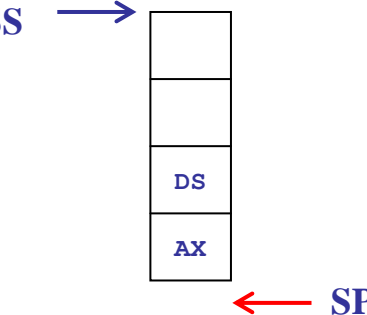
Чтение из стека

Команда **pop** («вытаскивает») считывает слово из вершины стека и присваивает его значение указанному операнду.

| |
|------------------------------|
| pop <i>назначение</i> |
|------------------------------|

| Назначение | |
|------------------|-----|
| sr (кроме CS) | m16 |

Пример работы операторов

| | |
|---|--|
| <p>Исходное состояние стека. Регистр SS указывает на начало сегмента стека, регистр SP - на вершину стека.</p> <p>SP указывает на ячейку, лежащую под «дном» стека и не входящую в него.</p> |  |
| <p>push ax</p> <ol style="list-style-type: none"> 1. Уменьшает на 2 байта значение SP. 2. Значение регистра ax записывается по адресу [SS:SP]. |  |
| <p>push ds</p> <ol style="list-style-type: none"> 1. Уменьшает на 2 байта значение SP. 2. Значение регистра ds записывается по адресу [SS:SP]. |  |
| <p>pop ds</p> <ol style="list-style-type: none"> 1. Значение по адресу [SS:SP] записывается в регистр ds. 2. Увеличивается на 2 байта значение SP. |  |
| <p>pop ax</p> <ol style="list-style-type: none"> 3. Значение по адресу [SS:SP] записывается в регистр ax. 4. Увеличивается на 2 байта значение SP. |  |

Пример

| Нижняя память | | | | |
|---------------|------|--|--|----------------------------------|
| 0F00: | 0000 | | | STACK 12 ; стек размером 12 байт |
| | 0002 | | | ... |
| | 0004 | | | mov ax, 100 |
| | 0006 | | | push ax ; SP2 |
| SS : SP3 | 0008 | | | mov bx, 200 |
| SS : SP2 | 000A | | | push bx ; SP3 |
| SS : SP1 | 000C | | | |

В регистре SS содержится адрес начала стекового сегмента - 0F00.

Регистр SP указывает на смещение относительно этого начального адреса в интервале от 0000 до 000A.

Последний байт стека имеет смещение 000C (12).

Стек растет в направлении уменьшения адресов памяти и уменьшается в противоположную сторону.

В стек можно записать только слово

Как работать с байтом?

Поместить в стек "*".

Для этого расширяем до слова и в стек уже записываем слово:

```
mov ax, "*"
push ax
```

При этом в силу «перевернутого» представления слов в памяти код символа "*" займет в стеке байт, на который указывает регистр SP:

