**Стек**

* **Стек** – структура данных, работающая по принципу LIFO (Last In, First Out)
* В Intel IA-32 стек – область памяти, на которую указывают регистры SS:SP
  + Может использоваться для временного хранения значений
  + Используется для работы с процедурами

SS

16 бит

Последнее добавленное

Не является содержимым стека

SP

64 КБ

…

…

…

…

**TOP VALUE**

**Команда push**

* Помещает значение операнда на вершину стека
* push src
  + SP <- SP - 2
  + Memory[SS:SP] <- src

**Команда pop**

* Извлекает значение с вершины стека
* Pop dest
  + dest <- Memory[SS:SP]
  + SP <- SP + 2

При запуске com-программы в вершине стека находится значение **0**

**Правила работы со стеком**

* Все помещенные на стек значения должны быть извлечены
* **НЕ СЛЕДУЕТ** извлекать больше значений, чем было помещено в стек

**PUSHF:**

* Помещает значение из регистра флагов на вершину стека
* ≈ push Flags

**POPF:**

* Перемещает значение с вершины стека в регистр флагов
* ≈ pop Flags

**PUSHA:**

* Сохраняет значения всех регистров общего назначения на стек

**POPA:**

* Считывает значения всех регистров общего значения всех регистров общего назначения из стека

**Стек и регистр E(SP)**

* push SP
  + В Intel 8086 в стек помещается новое значение SP (после уменьшения)
  + Начиная с 80286 – старое значение SP (до уменьшения)
* pop SP
  + Увеличивает значение до того, как данные со старой вершины стека будут помещены в SP
* pop [esp]
  + Эффективный адрес высчитывается после увеличения esp