22. Стек. Аппаратная поддержка вызова подпрограмм.

Параметры в подпрограмму передаются через стек.

Перед вызовом подпрограммы вызывающая программа кладёт в стек параметры, затем выполняет команду `CALL`, подпрограмма сама в своём начале сохраняет текущее значение регистра `SP` в регистре `BP` и дополнительно уменьшает `SP` на размер области памяти, необходимый для хранения локальных переменных.

`SP` меняется автоматически, но его можно также менять и напрямую.

Стек.

- **Стек** - структура данных, работающая по принципу LIFO (last in, first out) - последним пришёл, первым вышел.
- **Сегмент стека** - область памяти программы, используемая её подпрограммами, а также (вынужденно) обработчиками прерываний.
- `SP` (Stack Pointer - указатель на вершину стека), `BP` (Base Pointer - вспомогательный регистр, используемый программистами и компиляторами для составления подпрограмм).
- В x86 **стек "растёт вниз", в сторону уменьшения адресов (от старших адресов к младшим) (от конца сегмента к началу)**. В таком случае удобно определять переполнение стека, т.е. нужно просто отследить момент, когда `SP` стал равен нулю. Если бы этой механики не было, то приходилось бы где-то хранить размер стека.
- При запуске программы `SP` указывает на конец сегмента.

Команды непосредственной работы со стеком.

Каждая такая команда делает сразу несколько действий (работают за несколько тактов процессора; на аппаратном уровне разбиты на атомарные команды):

`PUSH`:

1. Уменьшает указатель вершины стека (регистр `SP`) на размер источника (того, что мы кладём в стек)
2. Записывает значение из источника по адресу `SS:SP`

`POP`:

1. Считывает значение из вершины стека (с адреса `SS:SP`) и записывает его в приёмник
2. Увеличивает указатель вершины стека (регистр `SP`) на размер приёмника (того, что мы достаём из стека)

Команды:

- `PUSH <источник>` - поместить данные в стек. Уменьшает `SP` на размер источника и записывает значение по адресу `SS:SP`.
- `POP <приёмник>` - считать данные из стека. Считывает значение с адреса `SS:SP` и увеличивает `SP`.
- `PUSHA` - поместить в стек регистры `AX`, `CX`, `DX`, `BX`, `SP`, `BP`, `SI`, `DI`.
- `POPA` - загрузить регистры из стека **(`SP` игнорируется)**
- `PUSHF` - поместить в стек содержимое регистра флагов
- `POPF` - загрузить регистр флагов из стека

Аппаратная поддержка вызова подпрограмм

Смотри вопросы про подпрограммы