

# Машинно-зависимые языки программирования, лекция 4

Каф. ИУ7 МГТУ им. Н. Э. Баумана, 2021 г.



# Использование стека подпрограммами

**Стековый кадр (фрейм)** — механизм передачи аргументов и выделения временной памяти с использованием аппаратного стека. Содержит информацию о состоянии подпрограммы.

Включает в себя:

- параметры
- адрес возврата (обязательно)
- локальные переменные



# Прерывания


- Прерывание - особая ситуация, когда выполнение текущей программы приостанавливается и управление передаётся программе-обработчику возникшего прерывания.
- Виды прерываний:
  - аппаратные (асинхронные) - события от внешних устройств;
  - внутренние (синхронные) - события в самом процессоре, например, деление на ноль;
  - программные - вызванные командой `int`.



# Маскирование прерываний

Внешние прерывания, в зависимости от возможности запрета, делятся на:

- **маскируемые** — прерывания, которые можно запрещать установкой соответствующего флага;
- **немаскируемые** (англ. Non-maskable interrupt, NMI) — обрабатываются всегда, независимо от запретов на другие прерывания



## Таблица векторов прерываний в реальном режиме работы процессора

- Вектор прерывания — номер, который идентифицирует соответствующий обработчик прерываний. Векторы прерываний объединяются в таблицу векторов прерываний, содержащую адреса обработчиков прерываний.
- Располагается в самом начале памяти, начиная с адреса 0.
- Доступно 256 прерываний.
- Каждый вектор занимает 4 байта - полный адрес.
- Размер всей таблицы - 1 Кб.



# Срабатывание прерывания

- Сохранение в текущий стек регистра флагов и полного адреса возврата (адреса следующей команды) - 6 байт
- Передача управления по адресу обработчика из таблицы векторов
- *Настройка стека?*
- *Повторная входимость (реентерабельность), необходимость запрета прерываний?*



## IRET - возврат из прерывания

- Используется для выхода из обработчика прерывания
- Восстанавливает FLAGS, CS:IP
- При необходимости выставить значение флага обработчик меняет его значение непосредственно в стеке



## Перехват прерывания

- Сохранение адреса старого обработчика
- Изменение вектора на "свой" адрес
- Вызов старого обработчика до/после отработки своего кода
- При деактивации - восстановление адреса старого обработчика





## Установка обработчика прерывания в DOS

- int 21h
  - AH=35h, AL= номер прерывания - возвращает в ES:BX адрес обработчика (в BX 0000:[AL\*4], а в ES - 0000:[AL\*4+2].)
  - AH=25h, AL=номер прерывания, DS:DX - адрес обработчика



# Некоторые прерывания

- 0 - деление на 0
- 1 - прерывание отладчика, вызывается после каждой команды при флаге TF
- 3 - "отладочное", int 3 занимает 1 байт
- 4 - переполнение при команде INTO (команда проверки переполнения)
- 5 - при невыполнении условия в команде BOUND (команда контроля индексов массива)
- 6 - недопустимая (несуществующая) инструкция
- 7 - отсутствует FPU
- 8 - таймер
- 9 - клавиатура
- 10h - прерывание BIOS



# Резидентные программы

- Резидентная программа - та, которая остаётся в памяти после возврата управления DOS
- Завершение через функцию 31h прерывания 21h / прерывание 27h
- DOS не является многозадачной операционной системой
- Резиденты - частичная реализация многозадачности
- Резидентная программа должна быть составлена так, чтобы минимизировать используемую память



## Завершение с сохранением в памяти

- **int 27h**
  - DX = адрес первого байта за резидентным участком программы (смещение от PSP)
- **int 21h, ah=31h**
  - AL - код завершения
  - DX - объём памяти, оставляемой резидентной, в параграфах



# Порты ввода-вывода

- Порты ввода-вывода - механизм взаимодействия программы, выполняемой процессором, с устройствами компьютера.
- IN - команда чтения данных из порта ввода
- OUT - команда записи в порт вывода
- Пример:

```
IN al, 61h  
OR al, 3  
OUT 61h, al
```



## Вопросы к рубежному контролю

1. Регистры общего назначения.
2. Сегментные регистры. Адресация в реальном режиме. Понятие сегментной части адреса и смещения.
3. Регистры работы со стеком.
4. Структура программы. Сегменты.
5. Прерывание 21h. Примеры ввода-вывода.
6. стек. Назначение, примеры использования.
7. Регистр флагов.
8. Команды условной и безусловной передачи управления.
9. Организация многомодульных программ.
10. Подпрограммы. Объявление, вызов.
11. Арифметические команды.
12. Команды побитовых операций.
13. Команды работы со строками.
14. Прерывания. Обработка прерываний.
15. Работа с портами ввода-вывода.