



Машинно-зависимые языки программирования, лекция 4

Каф. ИУ7 МГТУ им. Н. Э. Баумана, 2024 г.



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



Прерывание BIOS 10h

АН = 00h	установка видеорежима, код в AL
АН = 02h	установить позицию курсора
АН = 08h	считать символ и атрибуты из позиции курсора
АН = 09h	записать символ и атрибуты в позицию курсора
АН = 0Ch	задать пиксель
АН = 0Dh	прочитать цвет пикселя



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



Способы вызова старого обработчика прерывания

1 Вызов в начале командой CALL

2.1 Безусловная передача управления в конце командой JMP на переменную

2.2 Безусловная передача управления в конце командой JMP с сохранением адреса в машинном коде команды



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 - отдельное адресное пространство для 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
```



Список основных портов ввода-вывода

000-00F Контроллер DMA
010-01F Контроллер DMA (PS/2)
020-02F Главный контроллер прерываний
030-03F Подчиненный контроллер прерываний
040-05F Programmable Interval Timer (PIT)
060-06F Контроллер клавиатуры
070-071 Часы реального времени
080-083 DMA Page Register
090-097 Programmable Option Select (PS/2)
0A0-0AF PIC #2
0C0-0CF DMAC #2
0F0-0FF Математический сопроцессор, PCJr Disk Controller
100-10F Programmable Option Select (PS/2)
170-17F Hard Drive 1 (AT)
1F0-1FF Hard Drive 0 (AT)
200-20F Game Adapter
210-217 Expansion Card Ports
278-27F Parallel Port 3

2A2-2A3 Clock
2B0-2DF EGA/Video
2E2-2E3 Data Acquisition Adapter (AT)
2E8-2EF Последовательный порт COM4
2F8-2FF Последовательный порт COM2
300-31F Prototype Adapter, Periscope Hardware Debugger
360-36F Network
370-377 Контроллер дисководов FDD
378-37F Параллельный порт LPT2
380-38F SDLC Adapter
390-39F Cluster Adapter
3B0-3BF Monochrome Adapter
3BC-3BF Параллельный порт LPT1
3C0-3CF EGA/VGA
3D0-3DF Color Graphics Adapter
3E0-3EF Последовательный порт COM3
3F0-3F7 Контроллер дисководов FDD
3F8-3FF Последовательный порт COM1



Макроопределения

Макроопределение (макрос) - именованный участок программы, который ассемблируется каждый раз, когда его имя встречается в тексте программы.

- Определение:
имя MACRO параметры
.....
ENDM
- Пример:
load_reg MACRO register1, register2
push register1
pop register2
ENDM



Директива присваивания =

Директива присваивания служит для создания целочисленной макропеременной или изменения её значения и имеет формат:

Макроимя = Макровыражение

- Макровыражение (или Константное выражение) - выражение, вычисляемое препроцессором, которое может включать целочисленные константы, макроимена, вызовы макрофункций, знаки операций и круглые скобки, результатом вычисления которого является целое число
- Операции: арифметические (+, -, *, /, MOD), логические, сдвигов, отношения



Директивы отождествления EQU, TEXTEQU

Директива для представления текста и чисел:

Макроимя EQU нечисловой текст и не макроимя ЛИБО число

Макроимя EQU <Операнд>

Макроимя TEXTEQU Операнд

- Пример:

```
X EQU [EBP+8]
```

```
MOV ESI,X
```



Макрооперации

- % - вычисление выражение перед представлением числа в символьной
- форме
- <> - подстановка текста без изменений
- & - склейка текста
- ! - считать следующий символ текстом, а не знаком операции
- ;; - исключение строки из макроса



Блоки повторения

- REPT число ... ENDM - повтор фиксированное число раз
- IRP или FOR:
IRP form,<fact_1[,fact_2,...]> ... ENDM
Подстановка фактических параметров по списку на место формального
- IRPC или FORC:
IRPC form,fact ... ENDM
Подстановка символов строки на место формального параметра
- WHILE:
WHILE cond ... ENDM



Директивы условного ассемблирования

- IF:
IF c1
...
ELSEIF c2
...
ELSE
...
ENDIF
- IFB <par> - истинно, если параметр не определён
- IFNB <par> - истинно, если параметр определён
- IFIDN <s1>,<s2> - истинно, если строки совпадают
- IFDIF <s1>,<s2> - истинно, если строки разные
- IFDEF/IFNDEF <name> - истинно, если имя объявлено/не объявлено



Директивы управления листингом

- Листинг - файл, формируемый компилятором и содержащий текст ассемблерной программы, список определённых меток, перекрёстных ссылок и сегментов.
- TITLE, SUBTTL - заголовок, подзаголовок на каждой странице
- PAGE высота, ширина
- NAME - имя программы
- .LALL - включение полных макрорасширений, кроме ;;
- .XALL - по умолчанию
- .SALL - не выводить тексты макрорасширений
- .NOLIST - прекратить вывод листинга



Комментарии

comment @

... многострочный текст...

@