

**汇编语言程序设计大作业实验报告杀毒部分**

|  |  |
| --- | --- |
| 姓 名： | 马承乾 金为轩 李葳骏  李聪 李亚轩 连国鑫 |
| 年 级： | 2021级 |
| 指导教师： | 林颖 |
| 课程名 ： | 汇编语言程序设计 |

2023年 6 月

1. **实验目的**

1、对学生汇编语言程序实验课编程能力整体考查

2、考查学生团队合作编程的效率和团队学习能力

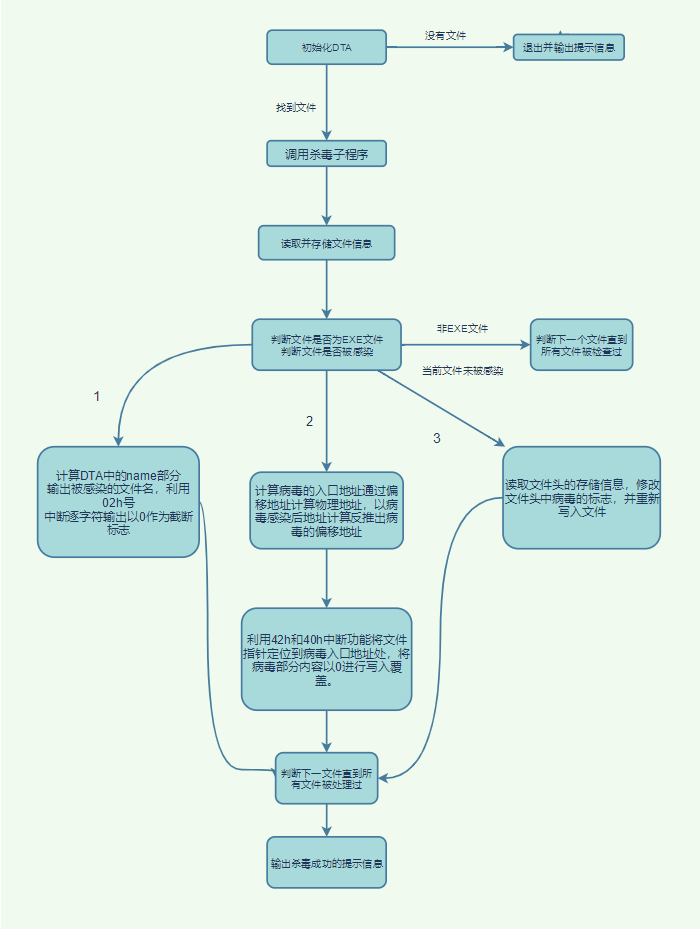
3、让学生完成具有一定的挑战性的任务

**二. 实验内容**

1、编写一个杀病毒程序kill.exe，运行 kill.exe能将DOS当前文件夹所有感染病毒EXE文件杀毒，未感染不杀。

1. **实验步骤以及结果**

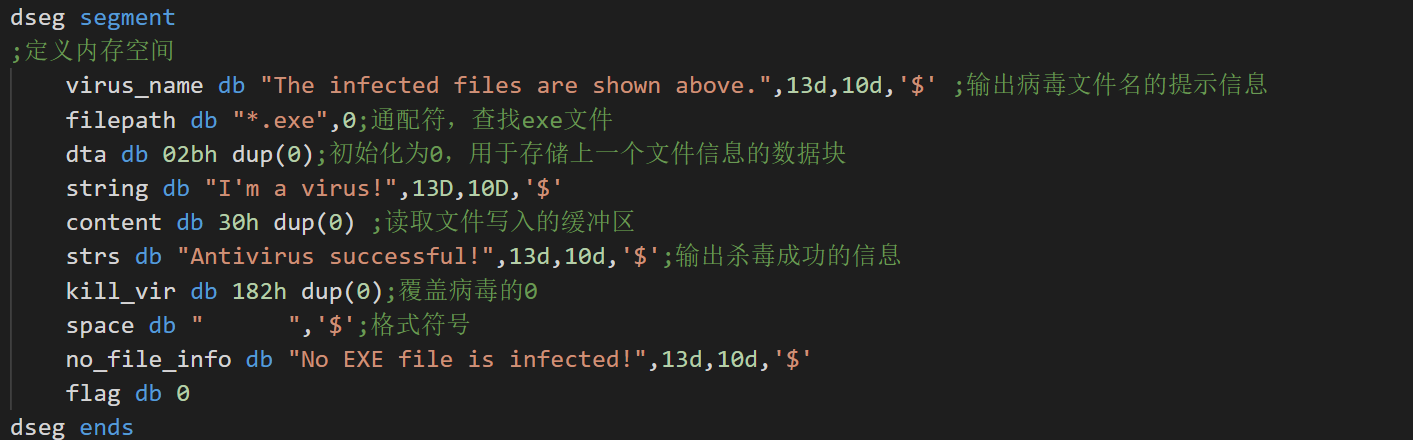
## 1、流程图



## 2、思路设计

首先，初始化相关变量和数据结构，然后通过查找符合条件的exe文件来定位潜在的病毒。对于每个符合条件的文件，检查其是否是有效的exe文件且未被感染。如果是有效且未感染的文件，将标志位设置为1，并输出病毒文件的名称。接下来，定位到病毒入口点，覆盖病毒代码并修改文件头，以清除病毒，然后将程序入口改回原程序入口。处理完所有的病毒文件后，输出相应的提示信息。如果没有发现病毒文件，则输出相应的提示信息。该杀毒程序的主要目标是查找和清除病毒感染的exe文件，以保护系统安全。

## 3、数据段定义



首先，在数据段（dseg segment）定义了一些数据和字符串的存储空间。其中的一些重要定义如下：

virus\_name: 存储输出病毒文件名的提示信息的字符串。

filepath: 存储要查找的文件路径的字符串，这里使用通配符\*.exe来 查找所有的EXE文件。

dta: 存储上一个文件信息的数据块，初始化为全零。

string: 存储病毒程序的字符串。

content: 用于读取文件内容的缓冲区。

strs: 存储杀毒成功的提示信息的字符串。

kill\_vir: 存储用于覆盖病毒的字符。

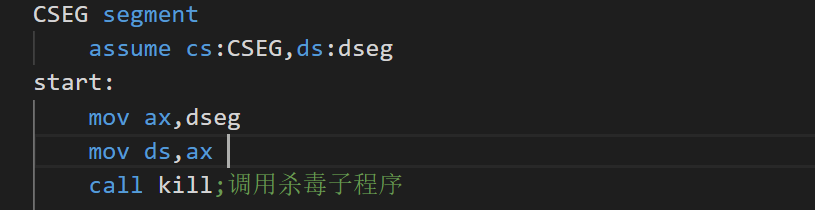
space: 用于输出病毒文件名前的空格的字符串。

no\_file\_info: 存储没有被感染的EXE文件的提示信息的字符串。

flag: 标志位，用于标记是否找到被感染的文件。

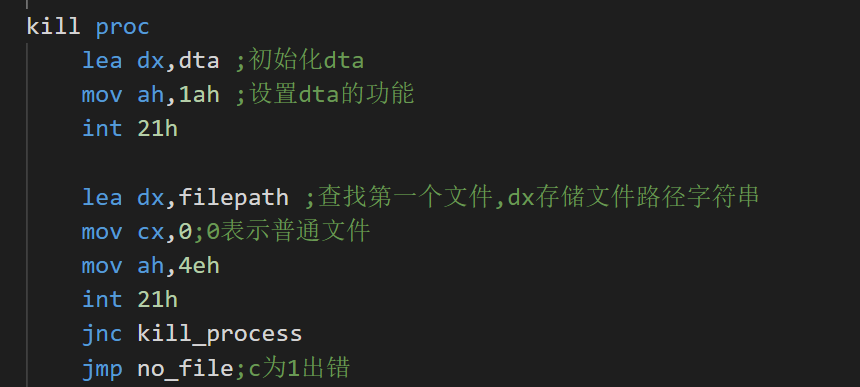
## 4、主程序实现

1. 调用kill标号进行杀毒



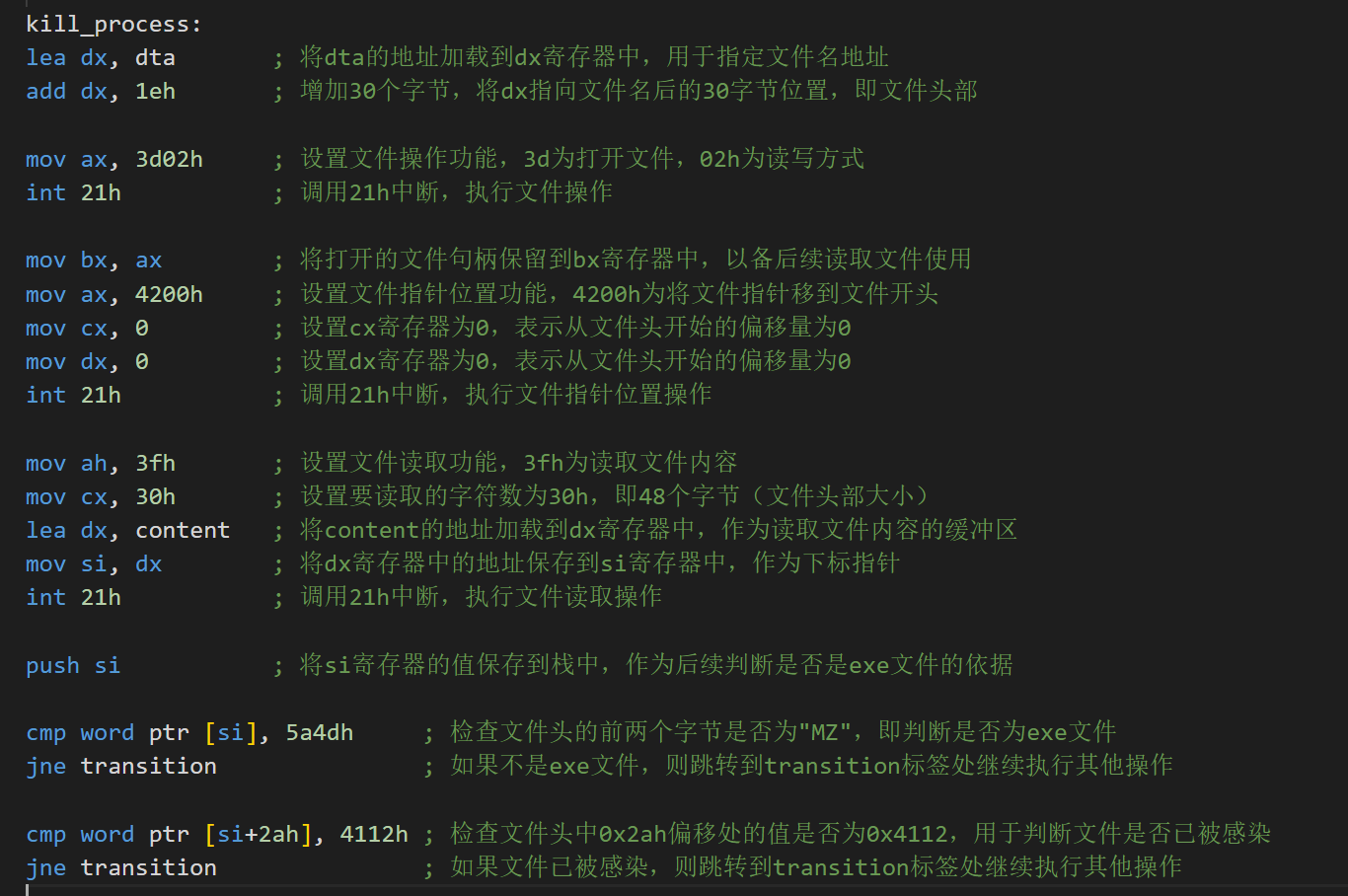
1. 初始化过程

接下来，在代码段（CSEG segment）中定义了程序的入口点start，然后调用了一个名为kill的杀毒标号（procedure）。



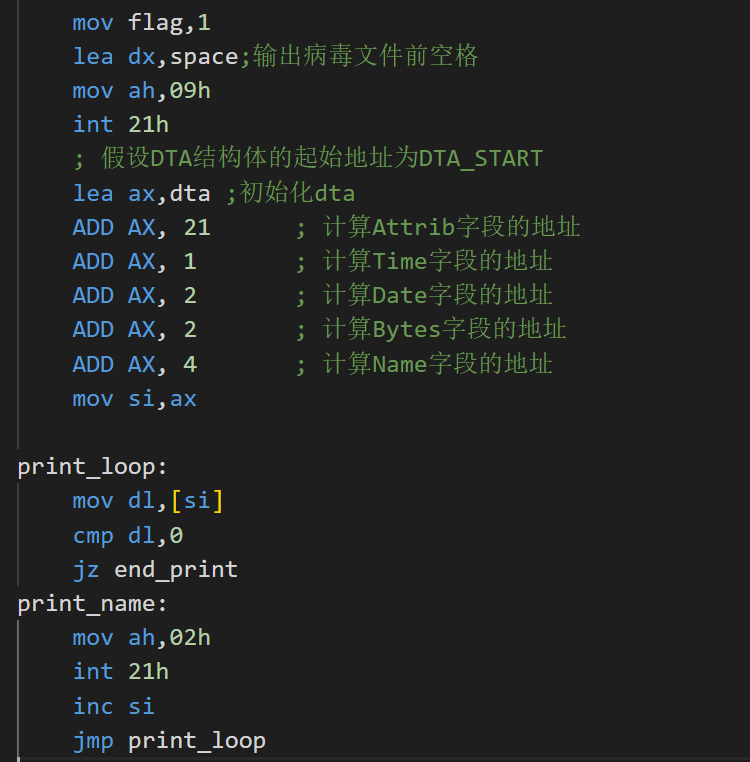
1. 查找文件并判断该文件是否是EXE文件

在kill子过程中，首先使用int 21h中断来初始化dta（目录表项）的值，指示DOS在dta中返回文件信息。然后使用int 21h中断来查找第一个满足条件的文件，然后进入具体的杀毒判断程序，并将文件信息存储在dta中。如果没有文件的话就输出没有文件的提示信息然后退出程序。



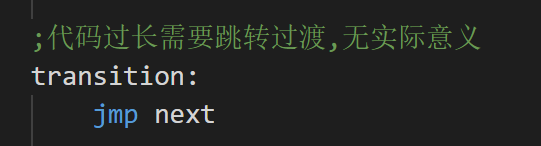
接着，打开找到的文件并设置文件指针开始读取文件内容以判断文件的信息，读取文件头部的内容到content缓冲区中。首先通过检查文件头的起那两个字节是否为“MZ”来判断它是否为一个EXE文件，不是的话则调用transition对下一个文件进行检查。然后检查文件头是否被感染主要是通过检查文件头中0x2ah偏移处的值是否为0x4112来判断文件是否被感染。如果是，则将标志位flag设为1，表示找到需要杀毒的文件，如果不是则去判断下一个文件。

1. 从DTA中获取文件名并输出



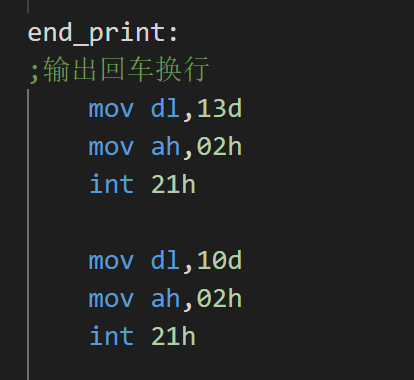
在找到被感染的文件后，首先输出病毒文件前面的空格，目的是为了调整文件名的输出格式。其次，输出病毒文件名主要是利用DTA中的name部分存储的信息，使用lea指令计算出DTA结构体中各字段的地址，并将计算结果保存到si寄存器中。然后进入一个循环，循环中首先将si指向的地址处的字符加载到dl寄存器中，然后使用int 21h中断和ah=02h功能来输出字符。然后递增si寄存器的值，指向下一个字符，然后无条件跳转回循环的开始处，继续打印下一个字符，直到遇到文件名结束符号（0）为止。这样就实现了逐字符打印文件名的功能。

1. 帮助过渡



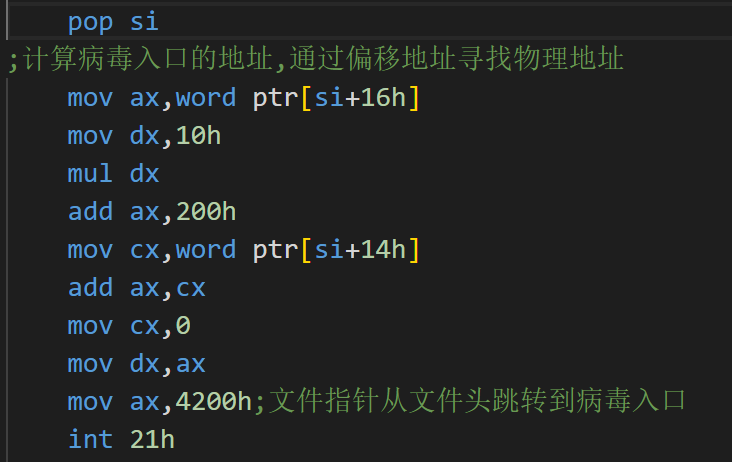
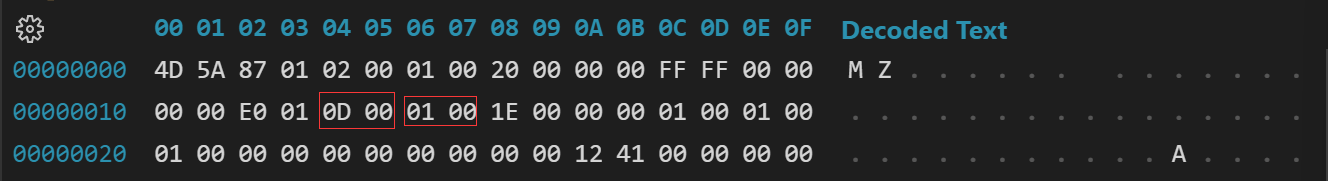
在上图中，transition标号并无实际的意义，而只是起到帮助跳转的过程，因为在程序的跳转过程中，在代码段中跳转的长度过长，超出短转移可跳转的的字节数目，其中一种方法就是在中间加一个过渡的标号，帮助进行跳转。

1. 输出名字后再次输出回车换行



在结束输出被感染文件的名字之后，输出一个回车以及换行符号，使输出格式更加美观。

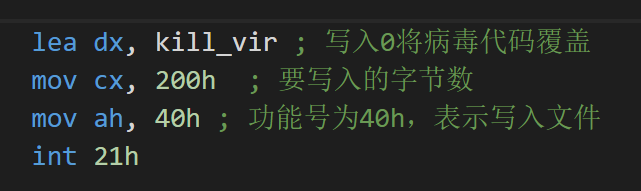
1. 寻找病毒代码入口的物理地址



在上图中的代码中，我们通过被感染文件的文件头，来寻找病毒的物理地址，以便于进文件指针的定位。先从栈中将si寄存器移出栈，si中存储的是寻找到文件后，该被感染文件的文件头内容的偏移地址，因此通过si寄存器我们能够访问文件头中的操作。下图中的二进制文件中，偏移量14h~15h存储的是病毒的起始的地址，而16h~17h中存储的是装入模块代码相对段值（CS），则通过该两项内容可以求得病毒模块的实际入口地址。将16h~17h中的内容赋给ax寄存器，由于16h~17h中存储的是代码段的相对段值，而该段值也是相对文件头的，因此需要先乘以10h之后再加上200h，求得cs段的物理地址后，由于14h~15h中所存储的内容是相对于cs段的相对位移量，因此将14h~15h的内容赋给cx寄存器，将ax寄存器加上cx寄存器中的值即为病毒段的物理地址，也是相对于文件最头部的相对位移量。

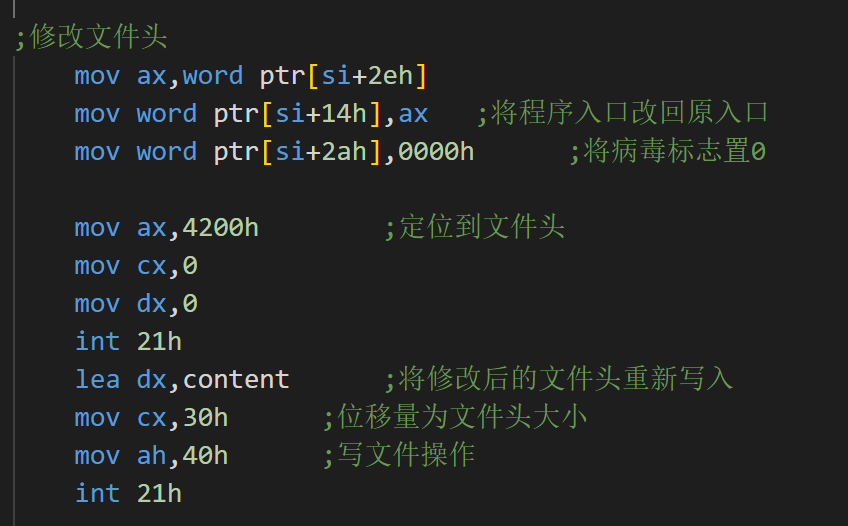
然后再调用DOS中断指令的42h号功能，由于文件指针移动指令的位移量为CX:DX两个寄存器，将cx寄存器的值置零，将ax寄存器的值赋给dx寄存器，此时位移量即为病毒入口地址相对于文件开头的位移量，则执行过该指令后，文件指针移动到病毒代码的入口地址。

1. 覆盖病毒代码起到杀毒作用



移动到病毒代码的入口地址后，然后就需要将写入的病毒代码使用0进行覆盖，因为覆盖过后，病毒代码则不能再起作用，也即起到了杀毒的作用。因此将kill\_vir的地址赋给dx寄存器，cx寄存器中需要存储的是要写入的字节数目，由于病毒代码的长度为165h左右，而其中由于写入了文件的名字，因此其长度大概为170h左右，因此写入200h个字节数目，保证病毒代码被充分覆盖掉。至此，病毒代码以及被覆盖。杀毒功能基本实现。

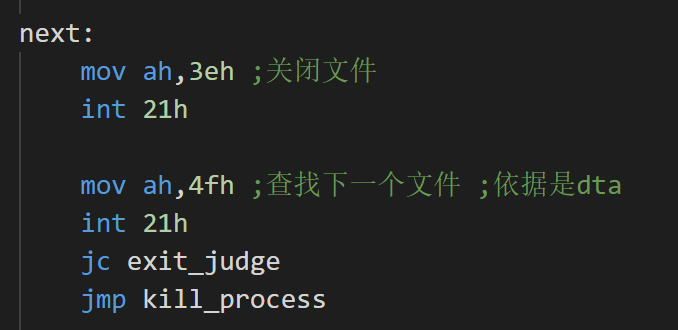
1. 将修改后的文件的信息通过修改文件头进行修改



完成覆盖操作后，接下来需要将文件头再修改回正确的状态，由于被感染文件的程序入口为病毒代码的起始地址，因此为了使其先运行正常的代码，需要将程序入口改为原程序的入口，在文件2eh~2fh中存储着原程序的入口地址，因此将其再覆盖到14h~15h的位置，同时将2ah~2bh位置的病毒标记设置为0表示此时未被感染。

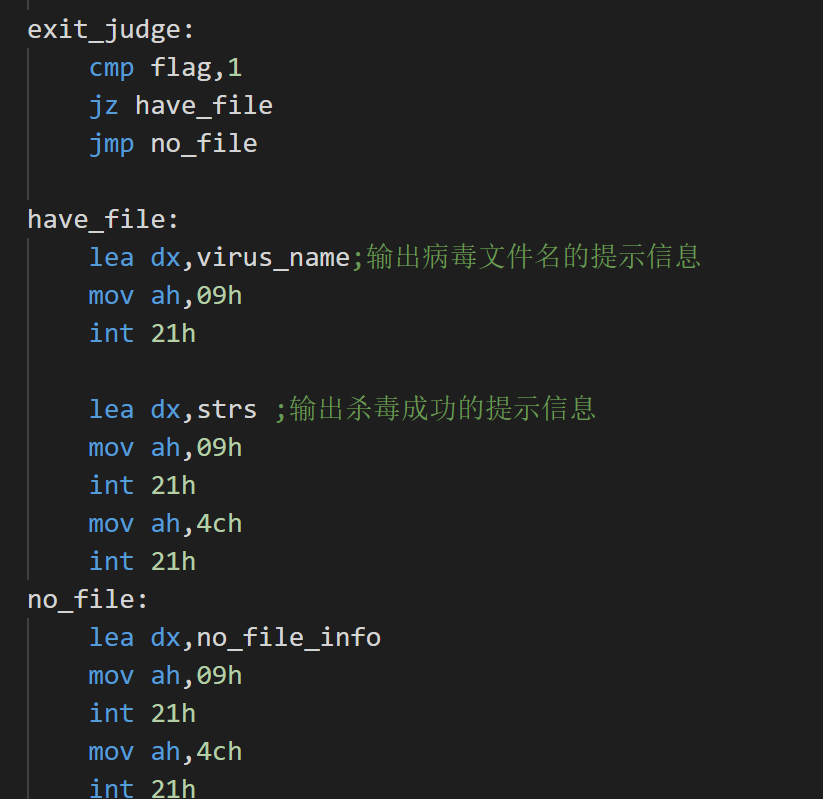
调用42h号的指令，将文件指针移动到文件头的位置，并调用DOS中断指令中的40h将修改后的文件头重新进行写入，写入的字节数目也即文件头的大小，即30h。

1. 依照DTA继续搜寻文件



调用3eh号指令关闭所打开的文件，并调用4fh号指令查找下一个文件。而如果能找到就跳转到kill\_process标号继续进行杀毒操作，如果找不到对应的文件就跳转到exit\_judge标号。

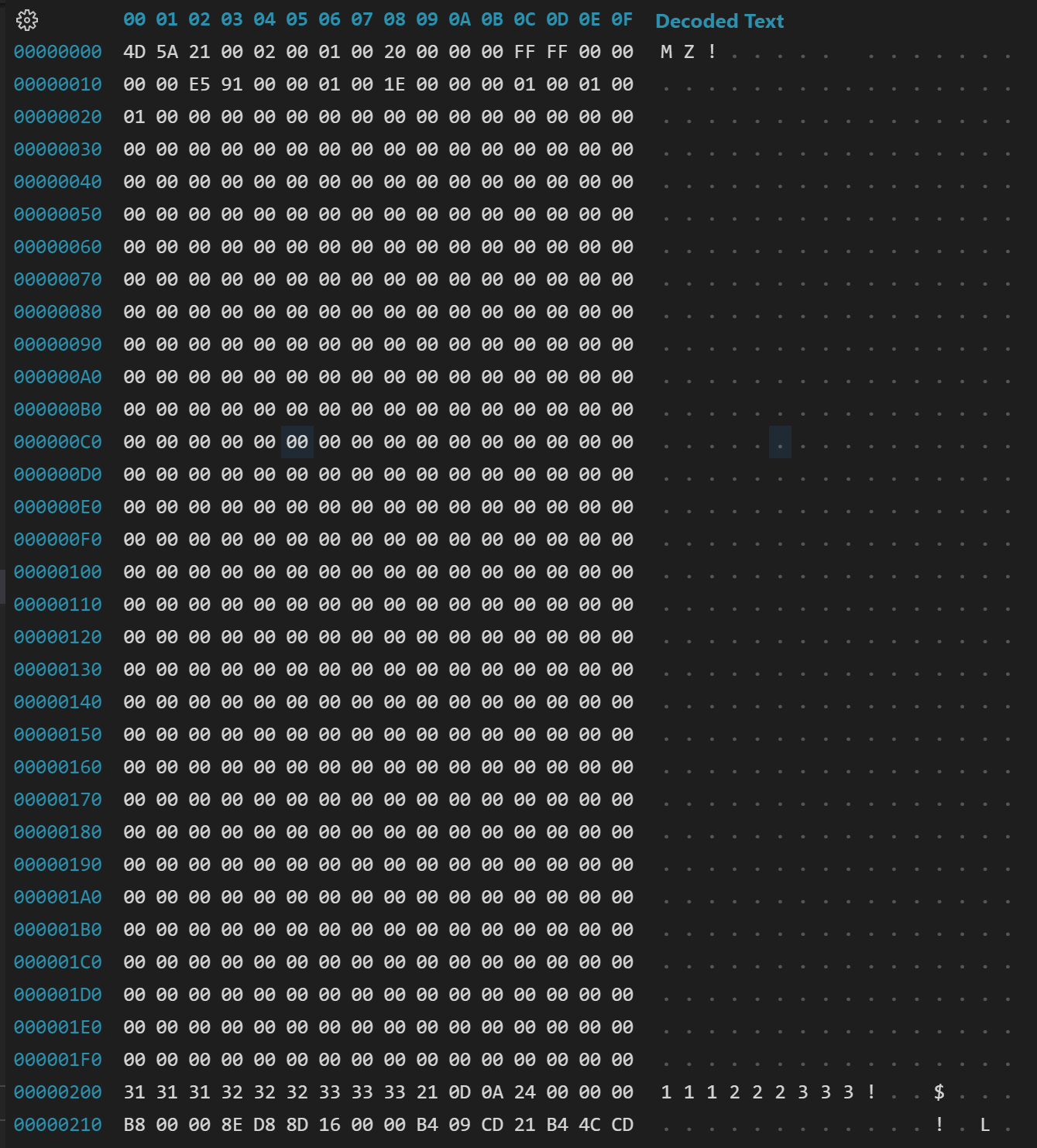
1. 输出提示信息并退出



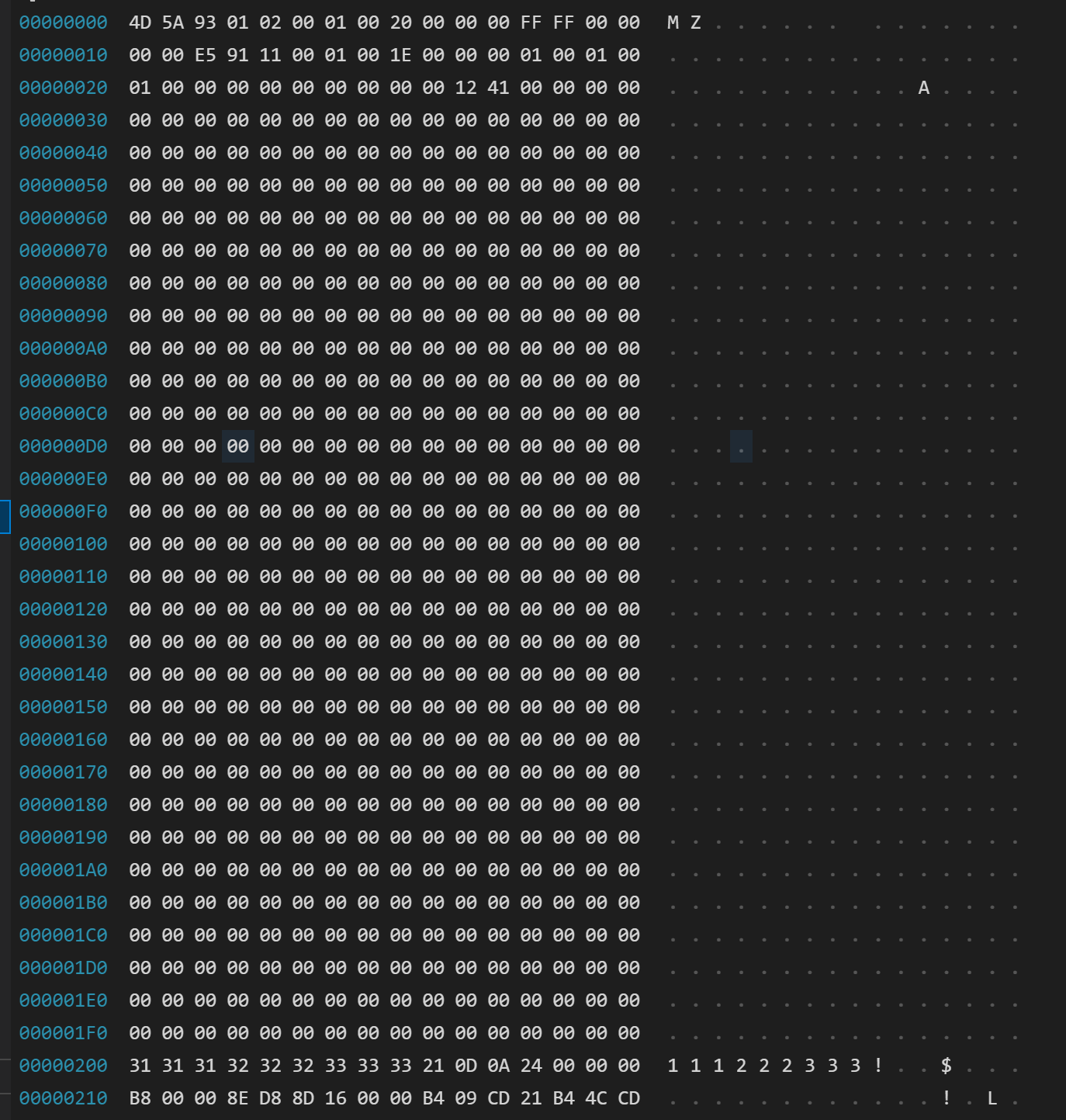
exit\_judge标号的作用是为了选择提示信息，如果flag为1，说明在整个的程序运行过程中找到了病毒文件，则输出"The infected files are shown above."，并接着输出"Antivirus successful!"而如果flag=0，说明无病毒文件被找到，则输出"No EXE file is infected!"。

1. **实验结果与分析**

未被感染的条件下，一个正常exe的二进制文件如下：



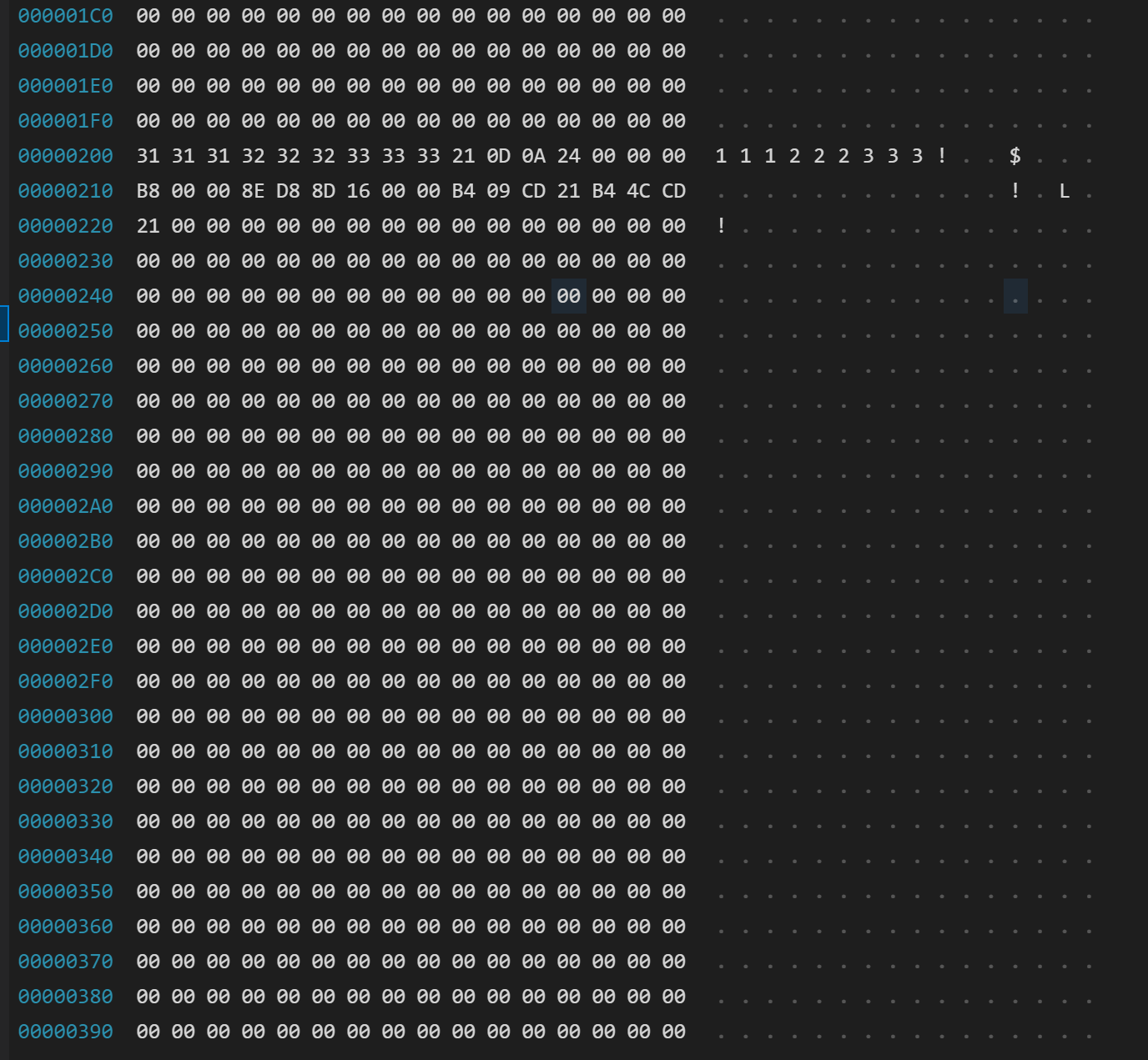
被感染后的exe的二进制文件如下：





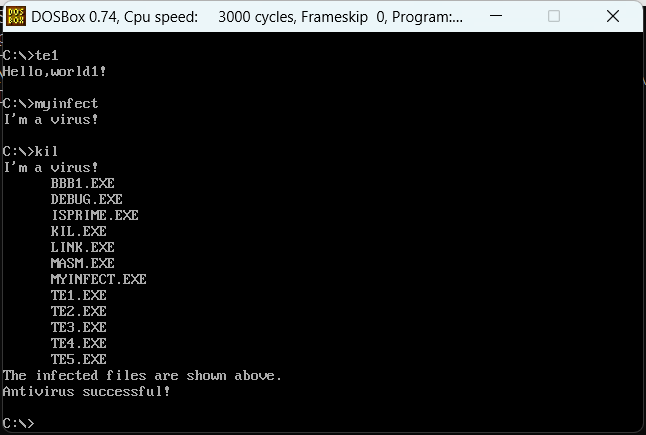
从文件中可见，其末尾被添加了感染病毒的代码段。

而杀毒后的二进制文件如图：

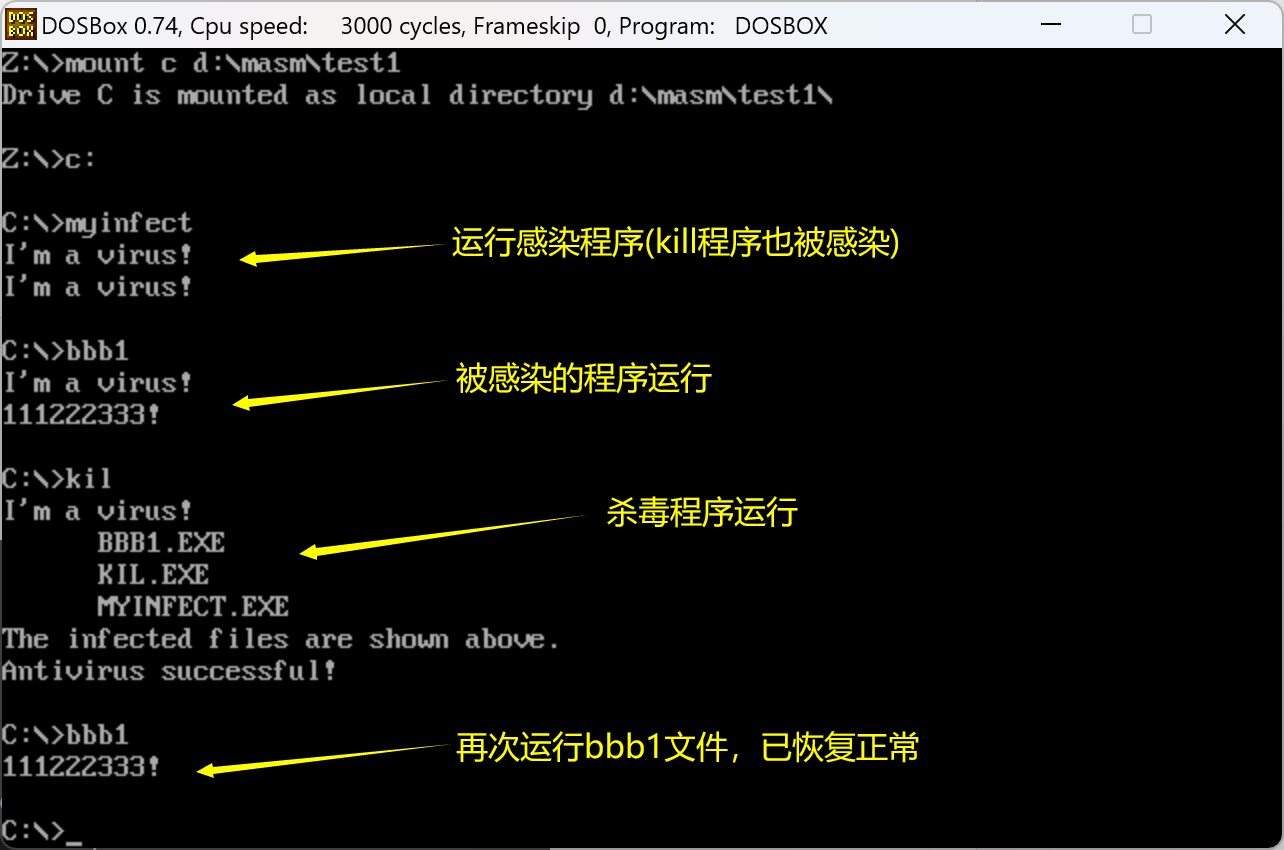


从图中可见，其末尾部分的病毒段代码被覆盖为0

总体的杀毒效果如图所示：



杀毒后文件运行效果



1. **实验总结**

在本次以汇编语言实现杀毒程序的实验中，我深刻地认识到了汇编语言的重要性和特殊性，同时也了解到了exe文件的结构、读取文件、写入文件操作等知识，也对汇编语言的指令更加熟练，对代码的编写更加熟练。

汇编语言作为一门重要的底层语言，它不仅可以直接操作硬件，还可以直接访问内存，因此在编写底层程序、驱动程序等方面具有很大的优势。在实验中，我对汇编语言的指令集、寄存器、堆栈等概念有了更加深刻的认识，并且掌握了基本的汇编语言编程技能。通过实验，我还了解到了如何使用调试器来调试汇编语言程序，这对于定位和解决程序中的错误非常有帮助。而exe文件是Windows操作系统中最常见的可执行文件格式之一，了解它的结构对于开发和调试Windows程序都非常有帮助。在实验中，我了解到了exe文件的结构，包括文件头、段表、节表等内容，并且学习了如何使用文件操作指令来读取和写入文件。这些知识不仅对于编写杀毒程序有用，也对其他类型的程序开发有很大的帮助。而通过本次实验我们也了解到了汇编语言的特殊性，它需要我们更加谨慎地编写程序，避免出现指令错误或者内存溢出等问题。同时，在实验中，我还学习到了如何编写杀毒程序，包括如何扫描目标文件，并且进行病毒检测等操作。这些知识对于计算机系统的开发和维护都非常重要。

通过本次以汇编语言实现杀毒程序的实验，我深刻地认识到了汇编语言的特殊性和重要性，同时也学习到了exe文件的结构、读取文件、写入文件操作等知识。这些知识对于我今后的学习和工作都将有很大的帮助。