Code:

```
#include <stdio.h>
#include <windows.h>

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        printf("Usage: %s <filename>\n", argv[0]);
        return 1;
    }

    HANDLE fileHandle = CreateFile(argv[1], GENERIC_READ, FILE_SHARE_READ, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
    if (fileHandle == INVALID_HANDLE_VALUE)
    {
        printf("Error opening file %s\n", argv[1]);
        return 1;
    }

    HANDLE mappingHandle = CreateFileMapping(fileHandle, NULL, PAGE_READONLY, 0, 0, NULL);
    if (mappingHandle == NULL)
    {
        printf("Error creating file mapping\n");
        CloseHandle(fileHandle);
        return 1;
```

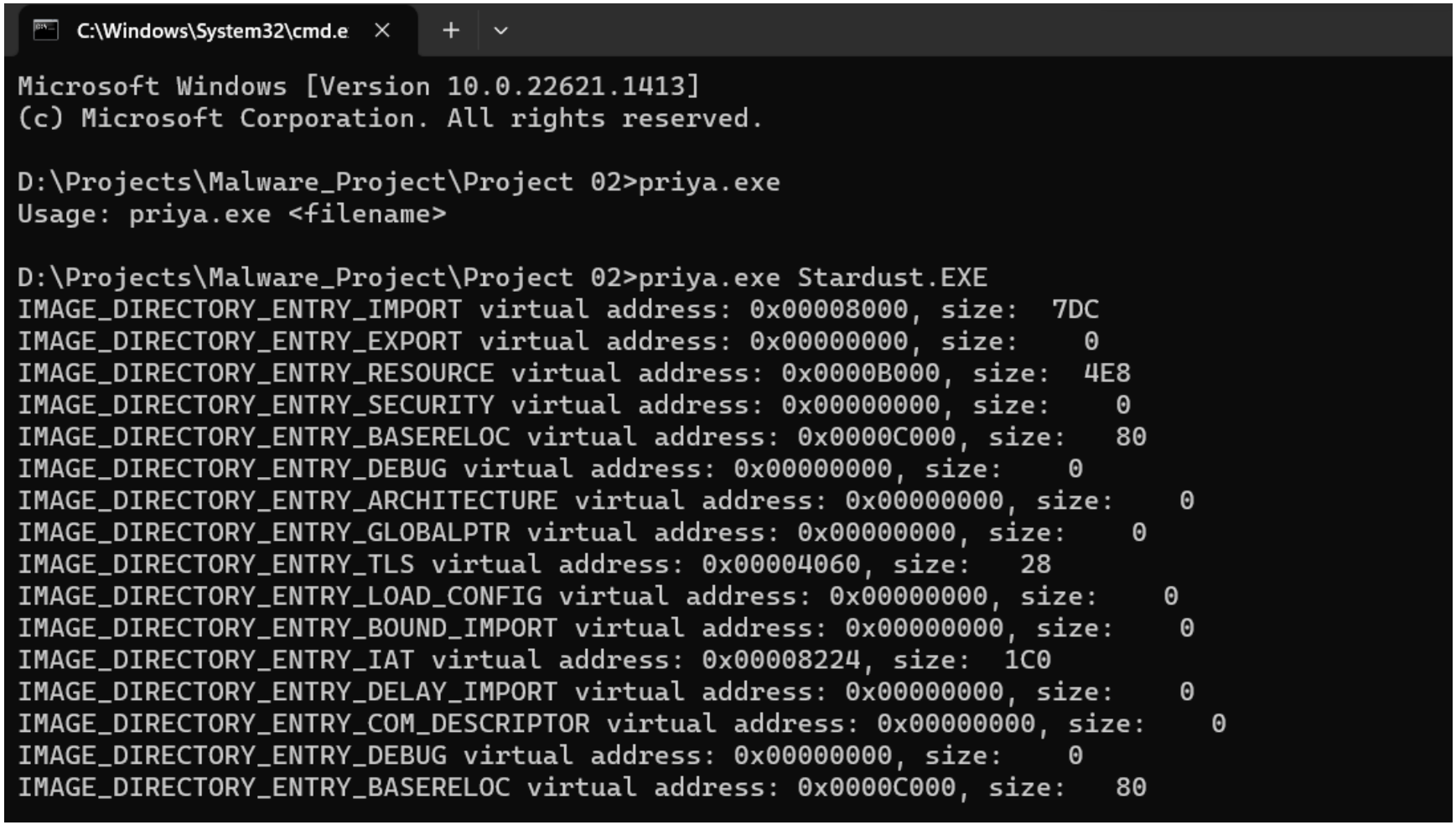
```
}

LPVOID mapView = MapViewOfFile(mappingHandle, FILE_MAP_READ, 0, 0, 0);
if (mapView == NULL)
{
    printf("Error creating file mapping view\n");
    CloseHandle(mappingHandle);
    CloseHandle(fileHandle);
    return 1;
}

PIMAGE_DOS_HEADER dosHeader = (PIMAGE_DOS_HEADER)mapView;
if (dosHeader->e_magic != IMAGE_DOS_SIGNATURE)
{
    printf("Invalid DOS signature\n");
    UnmapViewOfFile(mapView);
    CloseHandle(mappingHandle);
    CloseHandle(fileHandle);
    return 1;
}

PIMAGE_NT_HEADERS ntHeaders = (PIMAGE_NT_HEADERS)((LPBYTE)dosHeader + dosHeader->e_lfanew);
if (ntHeaders->Signature != IMAGE_NT_SIGNATURE)
{
    printf("Invalid NT signature\n");
    UnmapViewOfFile(mapView);
```

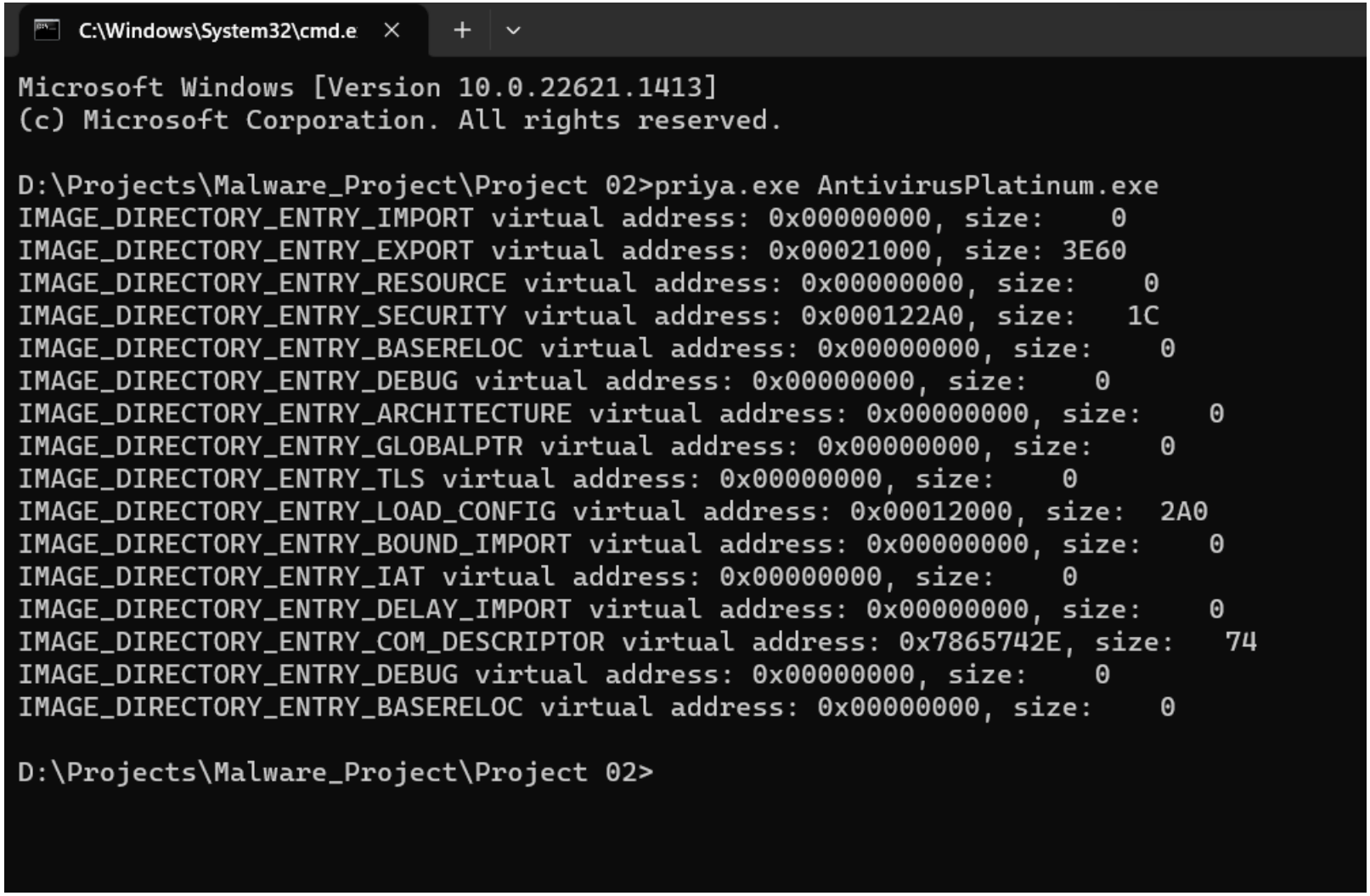
```
    CloseHandle(mappingHandle);
    CloseHandle(fileHandle);
    return 1;
}

PIMAGE_DATA_DIRECTORY importDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_IMPORT];
printf("IMAGE_DIRECTORY_ENTRY_IMPORT virtual address: 0x%08X, size: % 4X\n", importDirectory->VirtualAddress, importDirectory->Size);
PIMAGE_DATA_DIRECTORY exportDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_EXPORT];
printf("IMAGE_DIRECTORY_ENTRY_EXPORT virtual address: 0x%08X, size: % 4X\n", exportDirectory->VirtualAddress, exportDirectory->Size);

PIMAGE_DATA_DIRECTORY resourceDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_RESOURCE];
printf("IMAGE_DIRECTORY_ENTRY_RESOURCE virtual address: 0x%08X, size: % 4X\n", resourceDirectory->VirtualAddress, resourceDirectory->Size);

PIMAGE_DATA_DIRECTORY securityDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_SECURITY];
printf("IMAGE_DIRECTORY_ENTRY_SECURITY virtual address: 0x%08X, size: % 4X\n", securityDirectory->VirtualAddress, securityDirectory->Size);

PIMAGE_DATA_DIRECTORY baserelocDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_BASERELOC];
printf("IMAGE_DIRECTORY_ENTRY_BASERELOC virtual address: 0x%08X, size: % 4X\n", baserelocDirectory->VirtualAddress, baserelocDirectory->Size);

PIMAGE_DATA_DIRECTORY debugDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_DEBUG];
printf("IMAGE_DIRECTORY_ENTRY_DEBUG virtual address: 0x%08X, size: % 4X\n", debugDirectory->VirtualAddress, debugDirectory->Size);

PIMAGE_DATA_DIRECTORY architectureDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_ARCHITECTURE];
```

```
    printf("IMAGE_DIRECTORY_ENTRY_ARCHITECTURE virtual address: 0x%08X, size: % 4X\n", architectureDirectory->VirtualAddress,
architectureDirectory->Size);

    PIMAGE_DATA_DIRECTORY globalptrDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_GLOBALPTR];
    printf("IMAGE_DIRECTORY_ENTRY_GLOBALPTR virtual address: 0x%08X, size: % 4X\n", globalptrDirectory->VirtualAddress, globalptrDirectory-
>Size);

    PIMAGE_DATA_DIRECTORY tlsDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_TLS];
    printf("IMAGE_DIRECTORY_ENTRY_TLS virtual address: 0x%08X, size: % 4X\n", tlsDirectory->VirtualAddress, tlsDirectory->Size);

    PIMAGE_DATA_DIRECTORY loadconFigDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG];
    printf("IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG virtual address: 0x%08X, size: % 4X\n", loadconFigDirectory->VirtualAddress,
loadconFigDirectory->Size);

    PIMAGE_DATA_DIRECTORY boundImportDirectory = &ntHeaders-
>OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT];
    printf("IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT virtual address: 0x%08X, size: % 4X\n", boundImportDirectory->VirtualAddress,
boundImportDirectory->Size);

    PIMAGE_DATA_DIRECTORY iatDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_IAT];
    printf("IMAGE_DIRECTORY_ENTRY_IAT virtual address: 0x%08X, size: % 4X\n", iatDirectory->VirtualAddress, iatDirectory->Size);

    PIMAGE_DATA_DIRECTORY delayImportDirectory = &ntHeaders-
>OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT];
    printf("IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT virtual address: 0x%08X, size: % 4X\n", delayImportDirectory->VirtualAddress,
delayImportDirectory->Size);
```

```
    PIMAGE_DATA_DIRECTORY descriptorDirectory = &ntHeaders-
>OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR];
    printf("IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR virtual address: 0x%08X, size: % 4X\n", descriptorDirectory->VirtualAddress,
descriptorDirectory->Size);

    PIMAGE_DATA_DIRECTORY entryDebugDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_DEBUG];
    printf("IMAGE_DIRECTORY_ENTRY_DEBUG virtual address: 0x%08X, size: % 4X\n", entryDebugDirectory->VirtualAddress,
entryDebugDirectory->Size);

    PIMAGE_DATA_DIRECTORY relocDirectory = &ntHeaders->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_BASERELOC];
    printf("IMAGE_DIRECTORY_ENTRY_BASERELOC virtual address: 0x%08X, size: % 4X\n", relocDirectory->VirtualAddress, relocDirectory->Size);

    UnmapViewOfFile(mapView);
    CloseHandle(mappingHandle);
    CloseHandle(fileHandle);

    return 0;
}
```

OUPTPUT


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.
D:\Projects\Malware_Project\Project 02>priya.exe
Usage: priya.exe <filename>
D:\Projects\Malware_Project\Project 02>priya.exe Stardust.EXE
IMAGE_DIRECTORY_ENTRY_IMPORT virtual address: 0x00008000, size: 7DC
IMAGE_DIRECTORY_ENTRY_EXPORT virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_RESOURCE virtual address: 0x0000B000, size: 4E8
IMAGE_DIRECTORY_ENTRY_SECURITY virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_BASERELOC virtual address: 0x0000C000, size: 80
IMAGE_DIRECTORY_ENTRY_DEBUG virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_ARCHITECTURE virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_GLOBALPTR virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_TLS virtual address: 0x00004060, size: 28
IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_IAT virtual address: 0x00008224, size: 1C0
IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_DEBUG virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_BASERELOC virtual address: 0x0000C000, size: 80


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.
D:\Projects\Malware_Project\Project 02>priya.exe AntivirusPlatinum.exe
IMAGE_DIRECTORY_ENTRY_IMPORT virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_EXPORT virtual address: 0x00021000, size: 3E60
IMAGE_DIRECTORY_ENTRY_RESOURCE virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_SECURITY virtual address: 0x000122A0, size: 1C
IMAGE_DIRECTORY_ENTRY_BASERELOC virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_DEBUG virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_ARCHITECTURE virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_GLOBALPTR virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_TLS virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG virtual address: 0x00012000, size: 2A0
IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_IAT virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR virtual address: 0x7865742E, size: 74
IMAGE_DIRECTORY_ENTRY_DEBUG virtual address: 0x00000000, size: 0
IMAGE_DIRECTORY_ENTRY_BASERELOC virtual address: 0x00000000, size: 0
D:\Projects\Malware_Project\Project 02>