

C programkód:

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      FILE* fp;
7
8      fp = fopen("vezeteknev.txt", "w");
9
10     fprintf(fp, "Szeli Márk, Gazdaságinformatikus, B8VNQ7\n");
11
12     fclose(fp);
13
14     return 0;
15 }
```

Létrehozott fájl helye:

| Név | Módosítás dátuma | Típus | Méret |
|---------------|-------------------|--------------------|-------|
| bin | 2021.03.06. 15:33 | Fájlmappa | |
| obj | 2021.03.06. 15:33 | Fájlmappa | |
| main | 2021.03.06. 15:49 | C fájl | 1 KB |
| neptunkod.cbp | 2021.03.06. 15:32 | CBP fájl | 2 KB |
| vezeteknev | 2021.03.06. 15:38 | Szöveges dokume... | 1 KB |

Létrehozott fájl tartalma:

vezeteknev – Jegyzettömb

Fájl Szerkesztés Formátum Nézet Súgó

Szeli Márk, Gazdaságinformatikus, B8VNQ7

Dependency Walker KERNEL32.DLL API hívásai:

Dependency Walker - [neptunkod]

File Edit View Options Profile Window Help

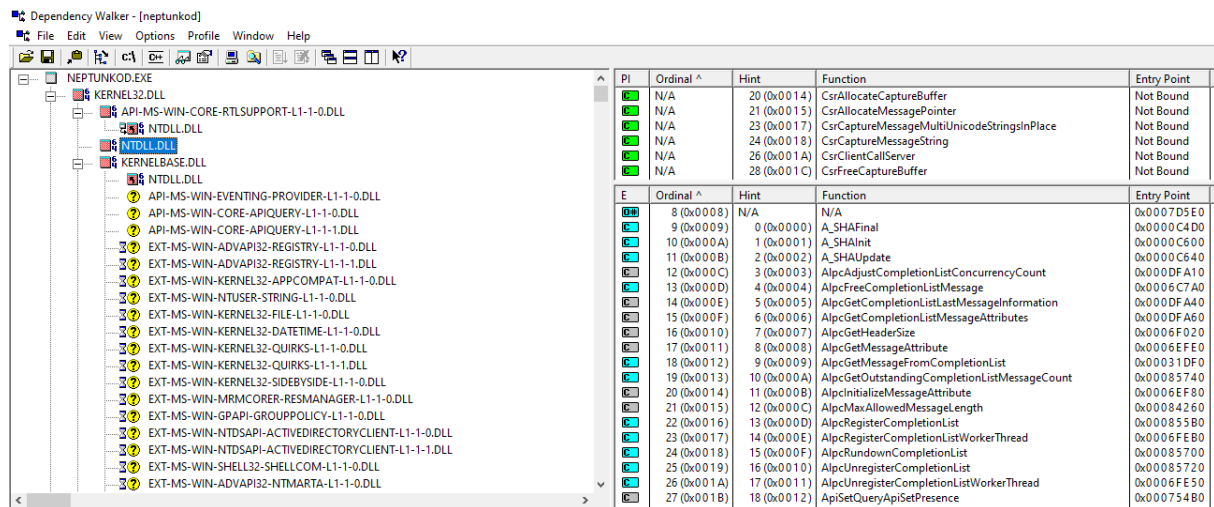
NEPTUNKOD.EXE

- KERNEL32.DLL
 - API-MS-WIN-CORE-RTLSUPPORT-L1-1-0.DLL
 - NTDLL.DLL
 - KERNELBASE.DLL
 - NTDLL.DLL
 - API-MS-WIN-EVENTING-PROVIDER-L1-1-0.DLL
 - API-MS-WIN-CORE-APIQUERY-L1-1-0.DLL
 - API-MS-WIN-CORE-APIQUERY-L1-1-1.DLL
 - EXT-MS-WIN-ADVAPI32-REGISTRY-L1-1-0.DLL
 - EXT-MS-WIN-ADVAPI32-REGISTRY-L1-1-1.DLL
 - EXT-MS-WIN-KERNEL32-APPCOMPAT-L1-1-0.DLL
 - EXT-MS-WIN-NTUSER-STRING-L1-1-0.DLL
 - EXT-MS-WIN-KERNEL32-FILE-L1-1-0.DLL
 - EXT-MS-WIN-KERNEL32-DATETIME-L1-1-0.DLL
 - EXT-MS-WIN-KERNEL32-QUIRKS-L1-1-0.DLL
 - EXT-MS-WIN-KERNEL32-QUIRKS-L1-1-1.DLL
 - EXT-MS-WIN-KERNEL32-SIDEBSIDE-L1-1-0.DLL
 - EXT-MS-WIN-MRMCORER-RESMANAGER-L1-1-0.DLL
 - EXT-MS-WIN-GPAPI-GROUPPOLICY-L1-1-0.DLL
 - EXT-MS-WIN-NTDSAPI-ACTIVEDIRECTORYCLIENT-L1-1-0.DLL
 - EXT-MS-WIN-NTDSAPI-ACTIVEDIRECTORYCLIENT-L1-1-1.DLL
 - EXT-MS-WIN-SHELL32-SHELLCOM-L1-1-0.DLL
 - EXT-MS-WIN-ADVAPI32-NTMARTA-L1-1-0.DLL

| PI | Ordinal ^ | Hint | Function | Entry Point |
|----|-----------|------------|---------------------------------|-------------|
| | N/A | 0 (0x0000) | RtlAddFunctionTable | Not Bound |
| | N/A | 2 (0x0002) | RtlCaptureContext | Not Bound |
| | N/A | 3 (0x0003) | RtlCaptureStackBackTrace | Not Bound |
| | N/A | 4 (0x0004) | RtlCompareMemory | Not Bound |
| | N/A | 5 (0x0005) | RtlDeleteFunctionTable | Not Bound |
| | N/A | 8 (0x0008) | RtlInstallFunctionTableCallback | Not Bound |

| E | Ordinal ^ | Hint | Function | Entry Point |
|---|-------------|-------------|---------------------------------|---------------------------------------|
| | 1 (0x0001) | 0 (0x0000) | RtlAddFunctionTable | ntdll.RtlAddFunctionTable |
| | 2 (0x0002) | 1 (0x0001) | RtlCaptureContext | ntdll.RtlCaptureContext |
| | 3 (0x0003) | 2 (0x0002) | RtlCaptureStackBackTrace | ntdll.RtlCaptureStackBackTrace |
| | 4 (0x0004) | 3 (0x0003) | RtlCompareMemory | ntdll.RtlCompareMemory |
| | 5 (0x0005) | 4 (0x0004) | RtlDeleteFunctionTable | ntdll.RtlDeleteFunctionTable |
| | 6 (0x0006) | 5 (0x0005) | RtlInstallFunctionTableCallback | ntdll.RtlInstallFunctionTableCallback |
| | 7 (0x0007) | 6 (0x0006) | RtlLookupFunctionEntry | ntdll.RtlLookupFunctionEntry |
| | 8 (0x0008) | 7 (0x0007) | RtlPcToFileHeader | ntdll.RtlPcToFileHeader |
| | 9 (0x0009) | 8 (0x0008) | RtlRaiseException | ntdll.RtlRaiseException |
| | 10 (0x000A) | 9 (0x0009) | RtlRestoreContext | ntdll.RtlRestoreContext |
| | 11 (0x000B) | 10 (0x000A) | RtlUnwind | ntdll.RtlUnwind |
| | 12 (0x000C) | 11 (0x000B) | RtlUnwindEx | ntdll.RtlUnwindEx |
| | 13 (0x000D) | 12 (0x000C) | RtlVirtualUnwind | ntdll.RtlVirtualUnwind |

Dependency Walker NTDLL.DLL:



| PI | Ordinal ^ | Hint | Function | Entry Point |
|----|-----------|-------------|---|-------------|
| 0 | N/A | 20 (0x0014) | CsrAllocateCaptureBuffer | Not Bound |
| 1 | N/A | 21 (0x0015) | CsrAllocateMessagePointer | Not Bound |
| 2 | N/A | 23 (0x0017) | CsrCaptureMessageMultiUnicodeStringsInPlace | Not Bound |
| 3 | N/A | 24 (0x0018) | CsrCaptureMessageString | Not Bound |
| 4 | N/A | 26 (0x001A) | CsrClientCallServer | Not Bound |
| 5 | N/A | 28 (0x001C) | CsrFreeCaptureBuffer | Not Bound |

| E | Ordinal ^ | Hint | Function | Entry Point |
|----|-------------|-------------|--|-------------|
| 6 | 8 (0x0008) | N/A | N/A | 0x0007D5E0 |
| 7 | 9 (0x0009) | 0 (0x0000) | A_SHAFinal | 0x0000C4D0 |
| 8 | 10 (0x000A) | 1 (0x0001) | A_SHAInit | 0x0000C600 |
| 9 | 11 (0x000B) | 2 (0x0002) | A_SHAUpdate | 0x0000C640 |
| 10 | 12 (0x000C) | 3 (0x0003) | AlpcAdjustCompletionListConcurrencyCount | 0x0000FA10 |
| 11 | 13 (0x000D) | 4 (0x0004) | AlpcFreeCompletionListMessage | 0x00006C7A0 |
| 12 | 14 (0x000E) | 5 (0x0005) | AlpcGetCompletionListLastMessageInformation | 0x0000DFA40 |
| 13 | 15 (0x000F) | 6 (0x0006) | AlpcGetCompletionListMessageAttributes | 0x0000DFA60 |
| 14 | 16 (0x0010) | 7 (0x0007) | AlpcGetHeaderSize | 0x00006F020 |
| 15 | 17 (0x0011) | 8 (0x0008) | AlpcGetMessageAttribute | 0x00006FE00 |
| 16 | 18 (0x0012) | 9 (0x0009) | AlpcGetMessageFromCompletionList | 0x00031DF0 |
| 17 | 19 (0x0013) | 10 (0x000A) | AlpcGetOutstandingCompletionListMessageCount | 0x000085740 |
| 18 | 20 (0x0014) | 11 (0x000B) | AlpcInitializeMessageAttribute | 0x00006EF80 |
| 19 | 21 (0x0015) | 12 (0x000C) | AlpcMaxAllowedMessageLength | 0x000084260 |
| 20 | 22 (0x0016) | 13 (0x000D) | AlpcRegisterCompletionList | 0x000085580 |
| 21 | 23 (0x0017) | 14 (0x000E) | AlpcRegisterCompletionListWorkerThread | 0x000085FB0 |
| 22 | 24 (0x0018) | 15 (0x000F) | AlpcShutdownCompletionList | 0x000085700 |
| 23 | 25 (0x0019) | 16 (0x0010) | AlpcUnregisterCompletionList | 0x000085720 |
| 24 | 26 (0x001A) | 17 (0x0011) | AlpcUnregisterCompletionListWorkerThread | 0x000085F50 |
| 25 | 27 (0x001B) | 18 (0x0012) | ApiSetQueryApiSetPresence | 0x0000754B0 |

Az ntdll.dll fájl egy, a Microsoft által létrehozott fájl az "NT Layer DLL" leírásával, és az NT kernel funkciókat tartalmazó fájl. Mivel ez a fájl a Windows része, soha nem szabad törölni vagy eltávolítani ezt a fájlt, ha úgy gondoljuk, hogy fertőzött, akkor hagyjuk, hogy az antivírus program kezelje.

A Native API egy könnyű alkalmazásprogramozási felület (API), amelyet a Windows NT és a felhasználói mód alkalmazásai használnak. Ezt az API-t a Windows NT indítási folyamatának korai szakaszában használják, amikor más összetevők és API-k továbbra sem érhetők el. Ezért néhány Windows-összetevőt, például a kliens/szerver futásidejű alrendszer (CSRSS), a Native API segítségével valósítunk meg. A Native API-t olyan szubrutinok is használják, mint például a kernel32.dll fájlban, amelyek a Windows API-t valósítják meg, azt az API-t, amely alapján a legtöbb Windows-összetevő létrejön.