Operációs rendszerek BSC

3B. gyak.

2021. 02. 24.

Készítette:

Miliczki József Bsc

GÉIK - Programtervező Informatikus

Y86I0I

Miskolc, 2021

4. Feladat

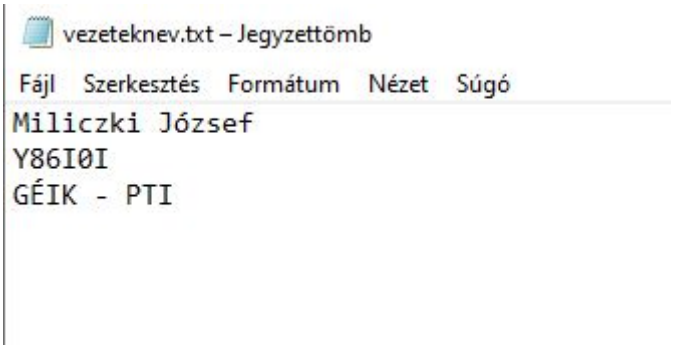
1) Készítsen egy neptunkod.c nevű forráskódot, amely egy vezeteknev.txt fájlt létrehoz, olvas, majd bezár. Tartalma: Név, Szak, Neptunkod etc.

Írtam egy rendkívül egyszerű kódot amely teljesíti a fenti kritériumokat:

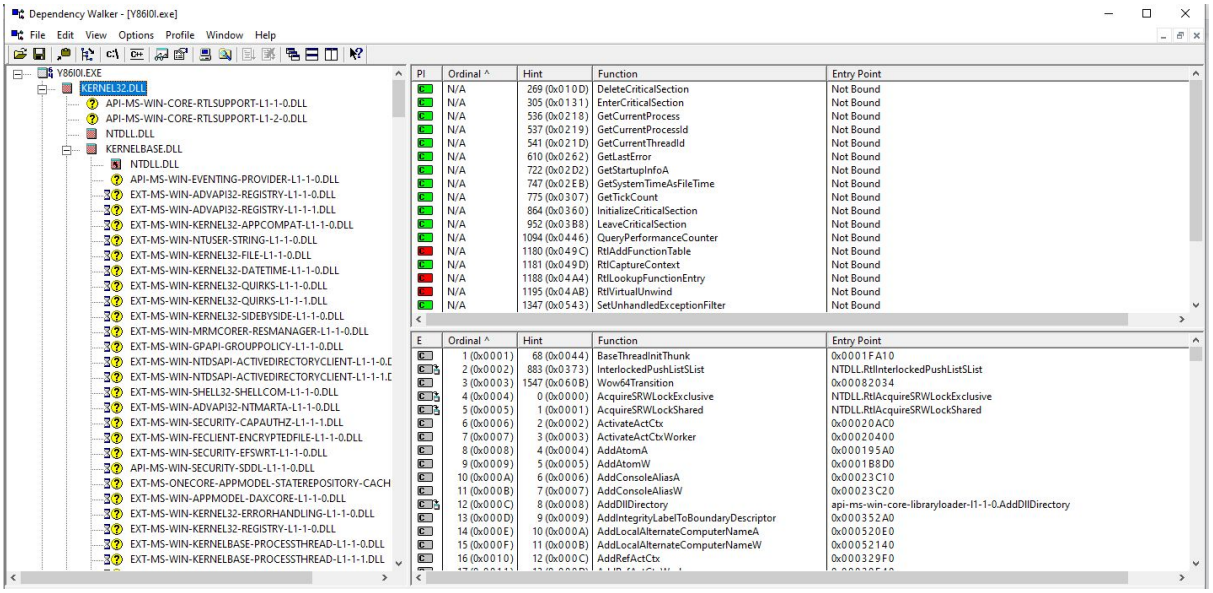
```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    FILE * file;
    char s[300];
    file = fopen("vezeteknev.txt", "r+");
    for (int i=0; i<4; i++)
        {
        fscanf(file, "%s", s);
        printf("%s\n", s);
        }
    return 0;
}
```

A vezeteknev.txt tartalma:



Nyissuk meg Dependency Walker-el a most legyártott programunkat:



a.) Vizsgálja meg, hogy a neptunkod.exe milyen API hívásokat használ a kernel32.dll-ből (Win alrendszer DLL)!

A fenti képből is látható hogy ténylegesen mennyi DLL-t használ a kernel32, ezekből jó párat ki fogok most emelni, és megvizsgálni milyen szolgáltatásokhoz engednek hozzá:

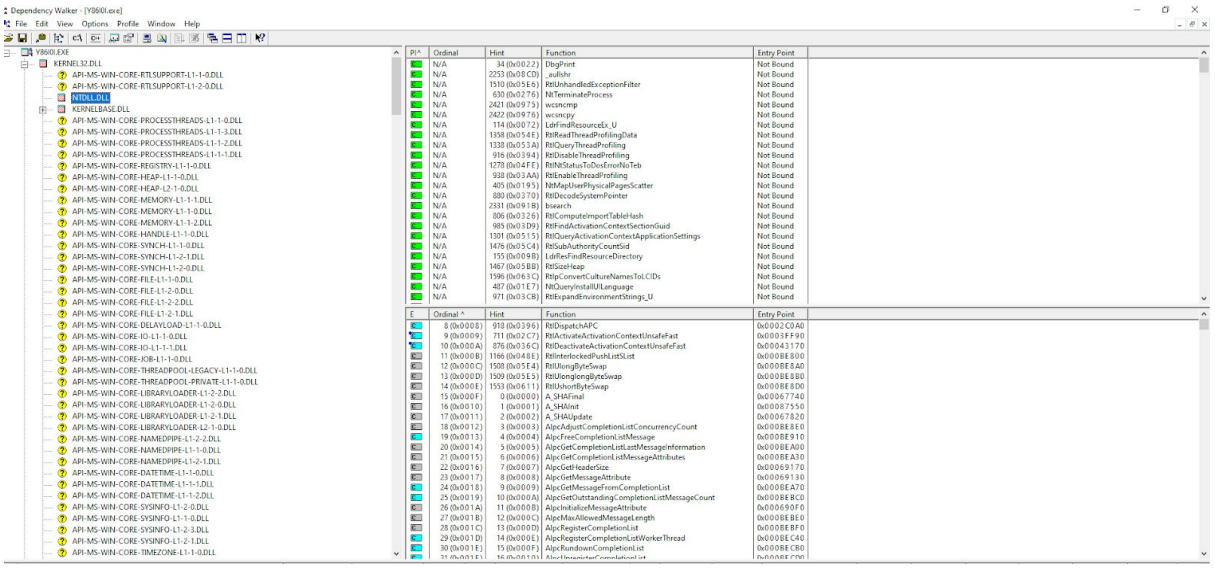
- **API-MS-WIN-CORE-PROCESSTHREADS-L1-1-0.DLL** - Ez az OS magja, és alapvető DLL file. Ezt abból lehet tudni hogy a Windows 7-es architektúra óta API-MS-WIN-el kezdődik minden olyan DLL file ami gyárilag jön az OS-el. Természetesen ez azt jelenti hogy vannak kívülről hozott Dynamic Link Library-k is.
- **API-MS-WIN-CORE-REGISTRY-L1-1-0.DLL -** A Windows registry kezelője. Számos funkcióhívás történik a System32-vel ezért a file hiánya súlyos következményekkel járhat. Maga a DLL 2012-ben készült Windows 8-ra, és utolsó frissítése 2019-ben jött ki az Oracle féle virtual machine-ra.
- **API-MS-WIN-CORE-HEAP-L1-1-0.DLL -** A Windows veremkezelő DLL-e. Olyan funkciókat találunk benne mint az Alloc és a Free. Azért ezeket emelem ki mert a fogalom előjön C programozáskor is, és funkciójuk identikus. A Windows 8-al jött be

- **API-MS-WIN-CORE-CONSOLE-L1-1-0.DLL -** A konzol működtetéséért felelős DLL. A hivatalos Microsoft dokumentáció szerint olyan funkciókat tartalmaz, amely a megjeleníthető karakterekkel foglalkozik, valamint a bevitt karaktereket fordítja le egy megadott kódtábla alapján. Nem meglepő, de főleg a bevitel és kiírás a feladata ennek a DLL-nek.
- **API-MS-WIN-CORE-ERRORHANDLING-L1-1-0.DLL** - Egy másik nagyon fontos DLL; hibakezelés. Olyan funkciókat találhatunk itt, amelyek azt biztosítják, hogy a processzek le tudják kommunikálni a problémákat, valamint végső esetben elindítsanak egy folyamatot amely eljuttatja az Error kódot a felhasználó irányába. Több módja van a hiba lekommunikálásának, a GetErrorMode például 4 féle hibát jelezhet.

Rengeteg más DDL található még a listán, viszont nem sorolhatom fel mindet. Lényegében alapvető Windows DLL file-okat használ a programom, amely természetesen nem meglepő.

b.) Milyen függőségei vannak a kernel32.dll-nek!
Őszinte leszek, ezt nagyon sokáig kutattam otthon és nem találtam rá kielégítő választ. A kernel32.dll-nek lehetséges hogy nincsenek függőségei? Inkább tőle függenek DLL-ek, és nem fordítva? Ha a kérdés a függvényekre gondol, akkor tudom mit kell keresnem, de a Microsoft oldala se adott útmutatást....

c.) Keresse meg NTDLL.DLL-t! Mi ennek a szerepe? Vizsgálja meg az exportált függvényeket, milyen információkat kap az NT API-ról!



Az **NTDLL.DLL** egy olyan erőforrás, amely nem a Win32 Subsystem része, pontosabban miatta létezhetnek az “egy réteggel” feljebb elhelyezkedő DLL fájloknak. Mivel technikailag nem alapszik egyik erőforrásfájlon sem, Natív API-nak is szokták nevezni. Használata főleg akkor kerül elő, mikor a rendszer elindul, és még nem állt fel a Win32 Subsystem. Mivel funkcionalitását megosztja a Kernel32-vel, így a rendszer felállása után “nagyon kevés” applikáció használja magát az NTDLL.DLL-t. Fundamentális természete miatt az NTDLL futtatható fájlok nem indíthatóak el a felhasználó által. Az ilyen fájlokat natív fájloknak nevezzük, ilyen például az **autochk.exe,** amely a jól ismert “kék halál”-nál indul el. Legkönnyebb az NTDLL.DLL-re úgy gondolni, mint egy lépcsőre ami utat nyit a ténylegesen runtime idő alatt használt DLL-ek számára!