

KPU Lab 1 DLL

Formål:

At opnå erfaring med brugen af DLL-filer i C++.

Forudsætninger

At du har læst på MSDN om DLL-filer i C++.

Hjælp

Har du problemer med at komme i gang med opgaven, så kan du få noget hjælp ved at læse artiklen "Step-by-step guide to constructing and using DLLs", som ligger som pdf-fil i fildelingen (men den er noget forældet).

Delopgave 1.1:

Lav en simpel DLL kodet i C++.

1. Start Visual Studio og lav et nyt C++-projekt. Dette projekt skal være et Win-32 projekt af typen DLL, og husk at der ikke skal være kryds ved "empty project", men at der skal være kryds ved "export symbols".
2. Lav en funktion som ligger 2 tal sammen, og lav en anden funktion, som ligger 2 strenge sammen - denne laves i 2 udgaver en til datatypen char* og en til string typen.
3. Efter at DLL'en er færdig kontrolleres at den eksporterer de rigtige funktioner med utility programmet **Dumpbin**, som er et program der skal kaldes fra Visual Studio's commando prompt (se figur).

The screenshot shows a Windows command prompt window titled "Visual Studio Command Prompt (2010)". The command entered is "dumpbin /exports lab-01.1.1.dll". The output displays the export characteristics of the DLL, including the timestamp (Wed Jan 25 13:23:54 2012), version (0.00), ordinal base (1), number of functions (3), and number of names (3). It then lists the exports with their corresponding RVA and names. Finally, it provides a summary of the sections in the file.

```
C:\Users\per\Documents\Kurser\KPU1\Løsningsforslag\KPU1-lab01\Debug>dumpbin /exports lab-01.1.1.dll
Microsoft (R) COFF/PE Dumper Version 10.00.40219.01
Copyright (C) Microsoft Corporation. All rights reserved.

Dump of file lab-01.1.1.dll
File Type: DLL

Section contains the following exports for Lab-01.1.1.dll

00000000 characteristics
4F1FF45A time date stamp Wed Jan 25 13:23:54 2012
    0.00 version
        1 ordinal base
        3 number of functions
        3 number of names

ordinal hint RVA             name

    1      0 0001100F myAdd = @ILT+10<_myAdd>
    2      1 00011262 myAddStr = @ILT+605<_myAddStr>
    3      2 00011096 myAddString = @ILT+145<_myAddString>

Summary

1000 .data
1000 .idata
3000 .rdata
1000 .reloc
1000 .rsrc
7000 .text
10000 .textbss

C:\Users\per\Documents\Kurser\KPU1\Løsningsforslag\KPU1-lab01\Debug>
```

Delopgave 1.2:

Lav en simpel applikation som bruger DLL-en fra opg. 1.1.

1. Tilføj et nyt C++-projekt til solutionen. Dette projekt skal være en EXE af typen Win32-Console program.
2. Dette program skal benytte funktionerne fra DLL'en udviklet i opgave 1.1 via **load-time linking**.

Ved load-time-linking skal compileren bruge .h-filen, linkeren skal have adgang til .lib-filen og ved kørsel af .exe-fil skal loaderen kunne finde dll'en (den skal ligge sammen med exe-filen eller et sted i path'en).

Delopgave 1.3:

Lav en applikation som loader en DLL dynamisk, og bruger den.

1. Tilføj et nyt C++-projekt til solutionen. Dette projekt skal være en EXE af typen Win32-Console program.
2. Dette program skal benytte funktionerne fra DLL'en udviklet i opgave 1.1 via **run-time linking** som vist i msdn-artiklen.

Delopgave 2.1:

Lav en DLL kodet i C++ som eksporterer en klasse

1. Lav selv en simpel klasse som placeres i en DLL fil.

Delopgave 2.2:

Lav en simpel applikation som bruger klassen fra DLL-filen lavet i opgave 2.1

1. Lav et testprogram som benytter klassen i DLL-filen lavet i opg. 2.1 via load-time-linkning.