Чтобы получить список функций, экспортируемых DLL модулем, Вы должны использовать следующие функции:

MapAndLoad в imagehlp.pas.

ImageRvaToVa в imagehlp.pas.

Также Вы должны определить следующие структуры:

```
    const IMAGE_SIZEOF_SHORT_NAME = 8;
IMAGE_NUMBEROF_DIRECTORY_ENTRIES = 16; IMAGE_DATA_DIRECTORY =
packed record VirtualAddress : DWORD; Size : DWORD;
PIMAGE_DATA_DIRECTORY = ^IMAGE_DATA_DIRECTORY; IMAGE_SECTION_HEADER
= packed record Name : packed array [0..IMAGE_SIZEOF_SHORT_NAME-1] of Char;
PhysicalAddress : DWORD; // or VirtualSize (union); VirtualAddress : DWORD;
SizeOfRawData : DWORD; PointerToRawData : DWORD; PointerToRelocations :
DWORD; PointerToLinenumbers : DWORD; NumberOfRelocations : WORD;
NumberOfLinenumbers : WORD; Characteristics : DWORD; end;
PIMAGE_SECTION_HEADER = ^IMAGE_SECTION_HEADER;
IMAGE_OPTIONAL_HEADER = packed record { Standard fields. } Magic : WORD;
  MajorLinkerVersion : Byte; MinorLinkerVersion : Byte; SizeOfCode : DWORD;
SizeOfInitializedData : DWORD; SizeOfUninitializedData : DWORD; AddressOfEntryPoint
: DWORD; BaseOfCode : DWORD; BaseOfData : DWORD; { NT additional
fields. } ImageBase : DWORD; SectionAlignment : DWORD; FileAlignment :
DWORD; MajorOperatingSystemVersion : WORD; MinorOperatingSystemVersion :
WORD; MajorImageVersion : WORD; MinorImageVersion : WORD;
MajorSubsystemVersion : WORD; MinorSubsystemVersion : WORD; Reserved1 :
DWORD; SizeOfImage : DWORD; SizeOfHeaders : DWORD; CheckSum :
DWORD; Subsystem : WORD; DllCharacteristics : WORD; SizeOfStackReserve :
DWORD; SizeOfStackCommit : DWORD; SizeOfHeapReserve : DWORD;
SizeOfHeapCommit : DWORD; LoaderFlags : DWORD; NumberOfRvaAndSizes :
DWORD; DataDirectory: packed array[0..IMAGE_NUMBEROF_DIRECTORY_ENTRIES-1]
of IMAGE_DATA_DIRECTORY; end; PIMAGE_OPTIONAL_HEADER =
^IMAGE_OPTIONAL_HEADER; IMAGE_FILE_HEADER = packed record Machine
: WORD; NumberOfSections : WORD; TimeDateStamp : DWORD;
PointerToSymbolTable : DWORD; NumberOfSymbols : DWORD;
SizeOfOptionalHeader : WORD; Characteristics : WORD; end;
PIMAGE_FILE_HEADER = ^IMAGE_FILE_HEADER; IMAGE_NT_HEADERS = packed
```

record Signature : DWORD; FileHeader : IMAGE_FILE_HEADER; OptionalHeader : IMAGE_OPTIONAL_HEADER; end; PIMAGE_NT_HEADERS = ^IMAGE_NT_HEADERS; type LOADED_IMAGE = record ModuleName: pchar; //Имя модуля hFile: thandle; //Дескриптор файла MappedAddress: pchar; // the base address of mapped file FileHeader: PIMAGE_NT_HEADERS; //The Header of the file. LastRvaSection: PIMAGE_SECTION_HEADER; NumberOfSections: integer; Sections: PIMAGE_SECTION_HEADER ; Characteristics: integer; fSystemImage: boolean; fDOSImage: boolean; Links: LIST_ENTRY; SizeOfImage: integer; end; PLOADED_IMAGE= ^LOADED_IMAGE;

Более подробно об этих структурах, смотри на http://www.csn.ul.ie/~caolan/publink/winresdump/winresdump/doc/msdn_peeringpe.html

Сам код:

unit Unit1; interface uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Menus,structures,imagehlp; type TForm1 = class(TForm) ListBox1: TListBox; MainMenu1: TMainMenu; File1: TMenuItem; Open1: TMenuItem; OpenDialog1: TOpenDialog; ListBox2: TListBox; procedure Open1Click(Sender: TObject); private public procedure DLLFuncstoList(fname: string; alistbox: tlistbox); end; var Form1: TForm1; implementation {$R *.DFM} procedure TForm1.DLLFuncstoList(fname: string; alistbox: tlistbox); var fih: LOADED_IMAGE; pexpdir: PIMAGE_EXPORT_DIRECTORY; pexpnames: pdword; //pointer to list of exported fucntions pt1: PImageSectionHeader; i: integer; exportedfuncname: pchar; //exported function name begin alistbox.items.clear; pt1:= nil; MapAndLoad(pchar(fname), pchar('#0'), @fih, true, true); //load the file into memory. pExpDir:= PIMAGE_EXPORT_DIRECTORY(fih.FileHeader.OptionalHeader. DataDirectory[IMAGE_DIRECTORY_ENTRY_EXPORT].VirtualAddress); pExpDir:= PIMAGE_EXPORT_DIRECTORY(ImageRvaToVa (fih.FileHeader, fih.MappedAddress, DWORD(pExpDir), pt1)); pExpNames:= pExpDir.pAddressOfNames; pExpNames:= PDWORD(ImageRvaToVa (fih.FileHeader, fih.MappedAddress, dword(pExpNames), pt1)); pt1:= nil; for i:= 0 to pexpdir.NumberOfNames-1 do begin exportedfuncname:= pchar(ImageRvaToVa (fih.FileHeader, fih.MappedAddress, dword(pExpNames^), pt1)); alistbox.items.add(exportedfuncname); inc(pexpnames); end; UnMapAndLoad(@fih); //Un load the mapped file from memory. end; procedure TForm1.Open1Click(Sender: TObject); var begin if opendialog1.execute = true then begin DLLFuncstoList(opendialog1.filename, listbox1); end; end; end.