1.

#include <stdio.h>

#include <Windows.h>

#include <stdlib.h>

#include <tchar.h>

IMAGE\_DOS\_HEADER DOS\_HEADER;

IMAGE\_NT\_HEADERS NT\_HEADERS;

int \_tmain(int argc, \_TCHAR\* argv[])

{

FILE\* pfile;

errno\_t error;

if ((error = fopen\_s(&pfile, "C:/EmptyProject1.exe", "r")) != 0)

{

printf("無法開啟文件");

getchar();

}

else {

fread(&DOS\_HEADER, sizeof(struct \_IMAGE\_DOS\_HEADER), 1, pfile);

fseek(pfile, DOS\_HEADER.e\_lfanew, 0);

fread(&NT\_HEADERS, sizeof(struct \_IMAGE\_NT\_HEADERS), 1, pfile);

printf("(DWORD)Signature is %08X\n", NT\_HEADERS.Signature);

printf("(WORD)Machine is %04X\n", NT\_HEADERS.FileHeader.Machine);

printf("(WORD)NumberOfSections is %04X\n", NT\_HEADERS.FileHeader.NumberOfSections);

printf("(DWORD)TimeDateStamp is %08X\n", NT\_HEADERS.FileHeader.TimeDateStamp);

printf("(DWORD)PointerToSymbolTable is %08X\n", NT\_HEADERS.FileHeader.PointerToSymbolTable);

printf("(DWORD)NumberOfSymbols is %08X\n", NT\_HEADERS.FileHeader.NumberOfSymbols);

printf("(WORD)SizeOfOptionalHeader is %04X\n", NT\_HEADERS.FileHeader.SizeOfOptionalHeader);

printf("(WORD)Characteristics is %04X\n", NT\_HEADERS.FileHeader.Characteristics);

printf("sizeof IMAGE\_FILE\_HEADER is %d bytes", sizeof(\_IMAGE\_FILE\_HEADER));

}

system("pause");

return 0;

}

2.

#include <stdio.h>

#include <Windows.h>

#include <stdlib.h>

#include <tchar.h>

IMAGE\_DOS\_HEADER DOS\_HEADER;

IMAGE\_NT\_HEADERS NT\_HEADERS;

int \_tmain(int argc, \_TCHAR\* argv[])

{

FILE\* pfile;

errno\_t error;

if ((error = fopen\_s(&pfile, "C:/EmptyProject1.exe", "r")) != 0)

{

printf("無法開啟文件");

getchar();

}

else {

fread(&DOS\_HEADER, sizeof(struct \_IMAGE\_DOS\_HEADER), 1, pfile);

fseek(pfile, DOS\_HEADER.e\_lfanew, 0);

fread(&NT\_HEADERS, sizeof(struct \_IMAGE\_NT\_HEADERS), 1, pfile);

printf("(WORD)Magic is %04X\n", NT\_HEADERS.OptionalHeader.Magic);

//\_IMAGE\_OPTIONAL\_HEADER的大小已經由IMAGE\_FILE\_HEADER當中的SizeOfOptionalHeader來決定

//所以不需要寫sizeof(struct \_IMAGE\_OPTIONAL\_HEADER來證明\_IMAGE\_OPTIONAL\_HEADER的大小)

printf("(BYTE)MajorLinkerVersion is %02X\n", NT\_HEADERS.OptionalHeader.MajorLinkerVersion);

printf("(BYTE)MinorLinkerVersion is %02X\n", NT\_HEADERS.OptionalHeader.MinorLinkerVersion);

printf("(DWORD)SizeOfCode is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfCode);

printf("(DWORD)SizeOfInitializedData is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfInitializedData);

printf("(DWORD)SizeOfUninitializedData is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfUninitializedData);

printf("(DWORD)AddressOfEntryPoint is %08X\n", NT\_HEADERS.OptionalHeader.AddressOfEntryPoint);

printf("(DWORD)BaseOfCode is %08X\n", NT\_HEADERS.OptionalHeader.BaseOfCode);

printf("(DWORD)BaseOfData is %08X\n", NT\_HEADERS.OptionalHeader.BaseOfData);

printf("(DWORD)SectionAlignment is %08X\n", NT\_HEADERS.OptionalHeader.SectionAlignment);

printf("(DWORD)FileAlignment is %08X\n", NT\_HEADERS.OptionalHeader.FileAlignment);

printf("(WORD)MajorOperatingSystemVersion is %04X\n", NT\_HEADERS.OptionalHeader.MajorOperatingSystemVersion);

printf("(WORD)MinorOperatingSystemVersion is %04X\n", NT\_HEADERS.OptionalHeader.MinorOperatingSystemVersion);

printf("(WORD)MajorImageVersion is %04X\n", NT\_HEADERS.OptionalHeader.MajorImageVersion);

printf("(WORD)MinorImageVersion is %04X\n", NT\_HEADERS.OptionalHeader.MinorImageVersion);

printf("(WORD)MajorSubsystemVersion is %04X\n", NT\_HEADERS.OptionalHeader.MajorSubsystemVersion);

printf("(WORD)MinorSubsystemVersion is %04X\n", NT\_HEADERS.OptionalHeader.MinorSubsystemVersion);

printf("(DWORD)Win32VersionValue is %08X\n", NT\_HEADERS.OptionalHeader.Win32VersionValue);

printf("(DWORD)SizeOfImage is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfImage);

printf("(DWORD)SizeOfHeaders is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfHeaders);

printf("(DWORD)CheckSum is %08X\n", NT\_HEADERS.OptionalHeader.CheckSum);

printf("(WORD)Subsystem is %04X\n", NT\_HEADERS.OptionalHeader.Subsystem);

printf("(WORD)DllCharacteristics is %04X\n", NT\_HEADERS.OptionalHeader.DllCharacteristics);

printf("(DWORD)SizeOfStackReserve is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfStackReserve);

printf("(DWORD)SizeOfStackCommit is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfStackCommit);

printf("(DWORD)SizeOfHeapReserve is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfHeapReserve);

printf("(DWORD)SizeOfHeapCommit is %08X\n", NT\_HEADERS.OptionalHeader.SizeOfHeapCommit);

printf("(DWORD)LoaderFlags is %08X\n", NT\_HEADERS.OptionalHeader.LoaderFlags);

printf("(DWORD)NumberOfRvaAndSizes is %08X\n", NT\_HEADERS.OptionalHeader.NumberOfRvaAndSizes);

}

system("pause");

return 0;

}