



| | | | | |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------|-------------------------------------------------------------------------------------|-----|
|  | Instytut Informatyki Politechniki Śląskiej Zespół Mikroinformatyki i Teorii Automatów Cyfrowych | |  | |
| Rok akademicki: | Rodzaj studiów*: SSI/NSI/NSM | Języki Asemblerowe | LAB | III |

TEMAT:

Zapoznanie się z oprogramowaniem MASM 32, WINDBG OLLYDBG RADASM x86 (INTEL).

CEL:

Celem ćwiczenia jest poznanie innych niż Microsoft VC asemblerów i programów debuggerów procesorów x86.

ZAŁOŻENIA:

- Zainstalować MASM32 .
- Zainstalować WinDbg (Windows Debugger).
- Zainstalować ollyDbg.
- Zainstalować RadAsm.
- Otworzyć program w edytorze qeditor.exe z pakietu masm32.
- Skompilować i uruchomić program 1.
- Znaleźć w internecie na stronie MSDN (lub w pliku win32.hlp) parametry wywołania funkcji MessageBox i Exit Process.

WYKONANIE:

Pytanie 1: Opisz parametry wywołania funkcji MessageBox i ExitProcess.

Zamienić sposób wywołania w programie na INVOKE

Pytanie 2: Przedstaw dwa różne sposoby wywołania funkcji MessageBox (przy użyciu push i invoke).

Zapoznać się z opcjami masm32 wywołując polecenie: C:\masm32\bin\ml.exe /?.

Przeprowadzić kompilację programu z linii poleceń wykorzystując przełącznik /coff oraz opcjonalnie z przełącznikiem /c i bez niego.

W razie napotkania błędu "a subsystem can't be inferred and must be defined" użyć przełącznika /link /SUBSYSTEM:WINDOWS.

Docelowym wynikiem powyższych czynności powinno być otrzymanie pliku exe.

Pytanie 3: Czy ml.exe automatycznie wywołuje linkera? Czy bez użycia dodatkowych przełączników program kompiluje/linkuje się poprawnie?

Pytanie 4: Do czego służą użyte przełączniki?

Skompilować program z linii poleceń używając przełącznika masm32 do generowania pliku listingu dla różnych kombinacji dyrektyw .NOLIST .NOCREF .LISTALL. Znaleźć ewentualne różnice w plikach lst.

Pytanie 5: Do czego służy dyrektywa .NOLIST?

Pytanie 6: Do czego służy dyrektywa .NOCREF?

Pytanie 7: Do czego służy dyrektywa .LISTALL?

Przeprowadzić kompilację z linii poleceń używając przełącznika masm32 do generowania pliku map. Przeanalizować powstały plik .map.

Pytanie 8: Co znajduje się w pliku map?

Zmodyfikować odpowiednie pliki *.bat z katalogu c:\masm32\bin tak, aby kompilacja ze środowiska qeditor powodowała automatyczne wygenerowanie plików lst i map.

Pytanie 9: Jakich modyfikacji i w których plikach dokonałeś?

Skompilować i uruchomić program 2. Jeśli występują błędy dotyczące braku definicji jakichś funkcji (np. wsprintf) należy dodać odpowiedni plik nagłówkowy z lokalizacji c:\masm\include.

Pytanie 10: Czy przy kompilacji wystąpiły jakieś błędy? Jakie pliki nagłówkowe dodałeś do swojego programu, żeby go skompilować?

Program 2 używa debuggera dbgwin.exe do śledzenia wartości jednego z rejestrów.

Pytanie 11: W jaki sposób odbywa się to debuggowanie - porównaj ze znanymi Ci sposobami debuggowania programu.

Pytanie 12: Jakie funkcje/makra wykorzystuje program 2 do debuggowania w dbgwin.exe? Podaj jeszcze co najmniej 3 inne funkcje służące do podobnych celów (np. analizując plik nagłówkowy dla debuggera).

Pytanie 13: Jaki plik nagłówkowy oraz jaką bibliotekę musi inkludować plik programu 2, aby makra debuggera działały poprawnie?

Uruchomić debuggowanie programu 1 w WinDbg i OllyDbg (w domu lub na zajęciach). Dla każdego z debuggerów odpowiedzieć na poniższe pytania:

Pytanie 14: Od jakiego adresu rozpoczyna się kod programu?

Pytanie 15: Czy tzw. plik symboli programu jest potrzebny debuggerowi? Czy i gdzie ustawiamy ścieżkę do tego pliku w debuggerze?

Pytanie 16: W jaki sposób ustawiamy pułapkę w testowanym debuggerze?

Pytanie 17: Do czego służy INT 3?

Sprawozdanie powinno zawierać wnioski - Twoje prywatne spostrzeżenia dotyczące rzeczy, które poznałeś na zajęciach :)

Program 1

```
.386
.model flat, stdcall
option casemap :none ; case sensitive
.nolist
.nocref
; #####
include \masm32\include\windows.inc
include \masm32\include\user32.inc
include \masm32\include\kernel32.inc
includelib \masm32\lib\user32.lib
includelib \masm32\lib\kernel32.lib
; #####
.list
.cref
.data
Tytul_okna db "Minimum MASM32", 0
Tekst_w_oknie db " --- To jest mój pierwszy program assemblerowy pod W2k! --- ", 0
.code
start:
push MB_OK
push offset Tytul_okna
push offset Tekst_w_oknie
push 0
call MessageBox
push 0
call ExitProcess

end start
```

Program 2

```
.686
.model flat, stdcall
option casemap :none ; case sensitive
.nolist
.nocref
;#####
    include c:\masm32\include\windows.inc
    include c:\masm32\include\masm32.inc
    include c:\masm32\include\kernel32.inc

    includelib c:\masm32\lib\masm32.lib
    includelib c:\masm32\lib\kernel32.lib

    include c:\masm32\include\debug.inc ;makra do debugowania PrintText
PrintDec
    includelib c:\masm32\lib\debug.lib
;#####
.list
.cref

.code

start:
    PrintText "Pomiar częstotliwości."
    mov ecx,0
    .repeat
        push ecx
        cpuid

        rdtsc
        mov ebx,eax
        invoke Sleep, 1000
        rdtsc
        sub eax,ebx

        PrintDec eax
        pop ecx
    inc ecx
    .until ecx==10

    invoke ExitProcess, 0

end start
```