Programowanie niskopoziomowe

Ćwiczenia laboratoryjne w środowisku Visual Studio 2019

Labolatoria 06

„Procedury i makroinstrukcje”

1 Zadania

1.1 Zad 1

Utwórz nowy projekt a następnie w tej samej solucji bibliotekę Nazwisko.lib i umieść w niej procedurę ScanInt. Wykorzystaj utworzoną bibliotekę w przykładzie lab06zad1.asm.

1.2 Zad 2

W przykładzie lab06zad2 jest umieszczone macro ReturnDescryptor oraz przykład użycia Invoke. Utwórz dwa dodatkowe macra i jedną dowolną dodatkową procedurę i użyj je w przykładzie.

Pozostałe wywołania procedur zastąp przez invoke.

Program ma realizować to samo, co zadanie arytmetyczne z laboratorium 4 wg wskazanego wariantu.

1.3 Zad 3

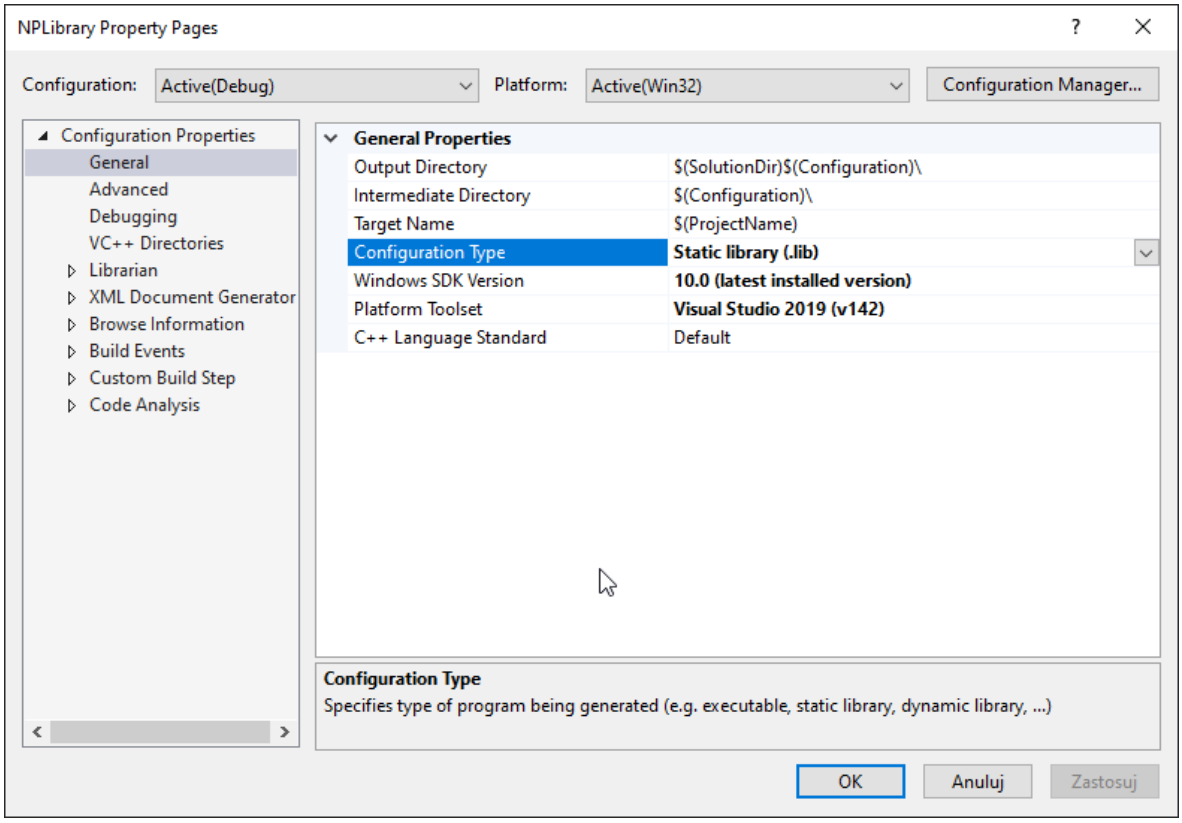
Na podstawie proc ScanInt utwórz procedury ScanBin (wczytywanie liczby w postaci binarnej) i ScanHex (wczytywanie liczby w postaci szesnastkowej). Przetestuj je w programie, w którym obliczamy sumę 3 liczb podanych kolejno w formacie: dziesiętnym, binarnym, szesnastkowym. Wynik ma być wyświetlany w postaci dziesiętnej.

---

2 Materiały pomocnicze:

2.1 Zadanie 1

1. W celu utworzenia pliku dll tworzymy pusty projekt dla języka c++ tak jak robiliśmy to poprzednio dla pliku exe, oraz ustawiamy w **Build Customization** środowisko na masm (nie ustawiamy Entry Point w linkerze).
2. Przechodzimy do właściwości projektu klikając prawym przyciskiem na projekt i wybierając **properties.**
3. Przechodzimy do **Configuration properties** -> **General** i zmieniamy **Configuration type** na Static library (.lib)



4. Następnie tworzymy nowy plik asm I wklejamy przykładową procedurę, która umieszcza w rejestrze EAX wartość 1000 i kończy swoje działanie.

```
.586P
.MODEL FLAT, STDCALL

PUBLIC PROC1

_TEXT SEGMENT

       PROC1 PROC
              MOV EAX, 1000
              RET
       PROC1 ENDP

_TEXT ENDS
END
```

5. Klikamy prawym przyciskiem na projekcie i wybieramy **Build**. Na konsoli powinna pojawić nam się ścieżka do utworzonej biblioteki.

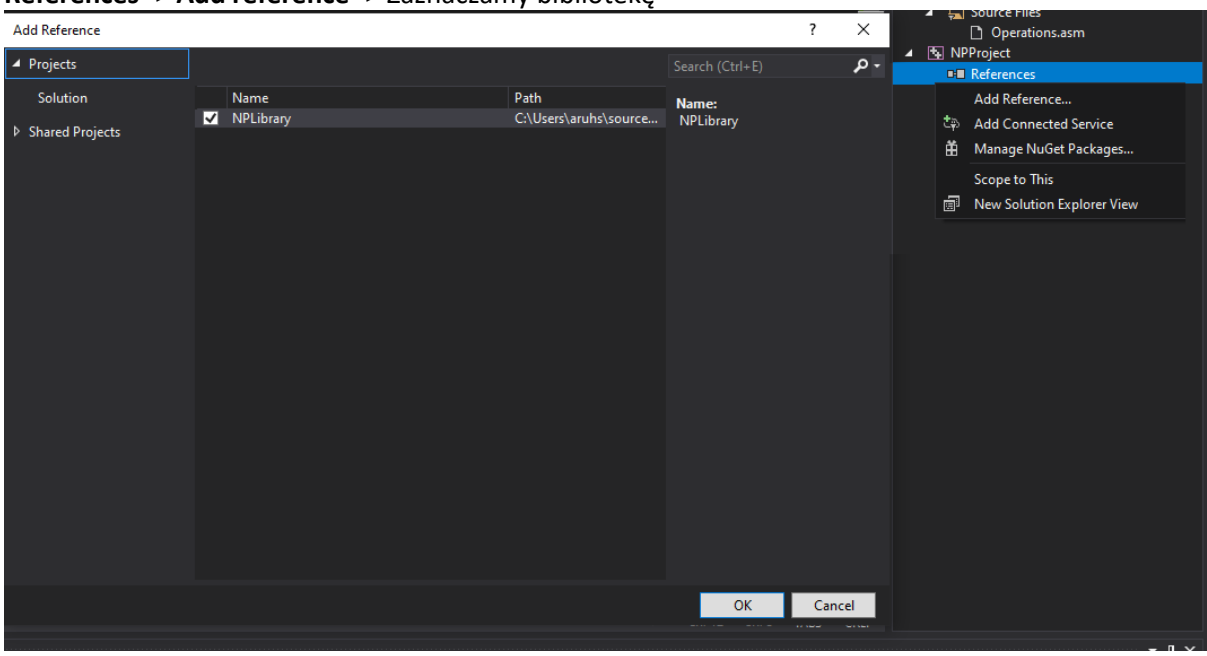
6. Dodajemy do solucji drugi projekt : Empty Project c++, a następnie

    a. Linkera:
    PPM na projekt -> **Properties** -> **Linker** -> **Advanced** -> **Entry point** = main

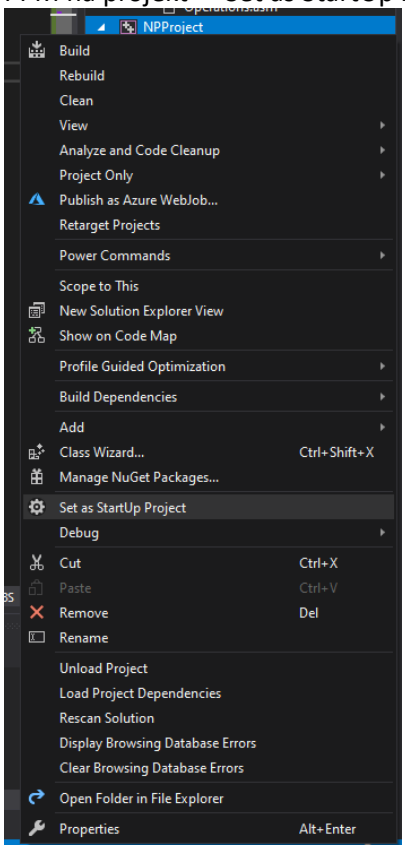
    b. Ustawmy build dependencies na MSMA:
    Build dependencies -> **Build customizations** -> zaznacz **masm**

7. Dodajmy do projektu wcześniej utworzoną bibliotekę

    References -> **Add reference** -> Zaznaczamy bibliotekę



8. Ustawmy nasz projekt aplikacji jako „StartUp” –
    PPM na projekt -> Set as StartUp Project



9. Następnie ustawiamy wszystko to co przy standardowym projekcie masm32 (build customization, entry point, tworzymy nowy plik)
10. Przykładowy program wykorzystujący bibliotekę:

```
;---prototypy ---
ExitProcess PROTO :DWORD
;--- procedura w naszej bibliotece ----
PROC1 PROTO

_DATA SEGMENT

_DATA ENDS
;------------
_TEXT SEGMENT
main proc

    call PROC1

    push    0
    call    ExitProcess
main endp
_TEXT   ENDS
END
```