ĆWICZENIE NR 4

Zadanie: utworzyć dwa wątki, zaimplementować algorytm Newtona Raphsona w każdym z nich. Ustawić ilość iteracji n tak aby wyliczenie jednego pierwiastka x zajmowało około jedną sekundę. Każdy wątek powinien w pętli wyliczyć pięć pierwiastków. Wyświetlać wyniki na ekran. Następnie sprawdzić czy na procesorze wielordzeniowym (można zastosować koligację rdzeni w menedżerze zadań) czy zauważalna jest szybkość wykonania programu.

$$\begin{cases} x_1 = \frac{a}{b} \\ x_n = \frac{1}{b}\left(x_{n-1} + \frac{a}{x_{n-1}}\right) \end{cases}$$

a – liczba pierwiastkowana

b – stopień pierwiastka

Przykład: W systemie Windows wątek to zwykła procedura uruchomiona w nieco inny sposób. Każdą procedurę można uruchomić jako osobny wątek. Służy do tego funkcja `CreateThread`.

```
INVOKE CreateThread, NULL, 0, OFFSET Thread1, 0,0,OFFSET Thread1ID
```

Parametry to kolejno atrybuty ochrony, rozmiar stosu wątku, adres procedury uruchamianej jako wątek, opcjonalny parametr przekazywany do procedury, flagi uruchomieniowe, adres zmiennej przechowującej identyfikator wątku. Wątki współdzielą dane miedzy sobą oraz z wątkiem głównym.

Synchronizacja odbywa się poprzez zdarzenia – są to zmienne boolowskie. Funkcja `WaitForSingleObject` zatrzymuje wątek w zależności od ustawionego zdarzenia. Ze względu na synchronizację większej ilości wątków użyto funkcji `WaitForMultipleObjects` która działa identycznie lecz zamiast jednego zdarzenia podaje się jako parametr tablicę zdarzeń.

```
INVOKE WaitForMultipleObjects, 2, OFFSET hEvent1, TRUE, INFINITE
```

Pierwszy parametr to ilość elementów w tablicy, następnie adres początku tablicy, wartość zdarzenia po której wątek główny ma kontynuować, maksymalny czas oczekiwania.

Kod przykładu:

```
include \masm32\include\masm32rt.inc

.data?
hInstance dd ?
hEvent1 dd ?
hEvent2 dd ?
```

```
Thread1ID dd ?
Thread2ID dd ?
tmpV dd ?


String db 20 dup(?)
.data

txt1 db " 1 ", NULL
txt2 db " 2 ", NULL

.code

Thread1 proc hIns:DWORD

mov ecx, 5

    p1:
            push ecx
            INVOKE WriteConsole, hInstance, OFFSET txt2, 3,ADDR tmpV, NULL
            INVOKE Sleep, 500
            pop ecx
    loop p1

     INVOKE SetEvent, hEvent1
ret
Thread1 endp

Thread2 proc hIns:DWORD

mov ecx, 5

    p1:
            push ecx
            INVOKE WriteConsole, hInstance, OFFSET txt1, 3,ADDR tmpV, NULL
            INVOKE Sleep, 800
            pop ecx
    loop p1

    INVOKE SetEvent, hEvent2

ret
Thread2 endp

Start:

INVOKE AllocConsole
INVOKE GetStdHandle, STD_OUTPUT_HANDLE
mov hInstance, eax


INVOKE CreateEvent, NULL, TRUE, FALSE, NULL
mov hEvent1, eax

INVOKE CreateEvent, NULL, TRUE, FALSE, NULL
mov hEvent2, eax

INVOKE CreateThread, NULL, 0, OFFSET Thread1, 0,0,OFFSET Thread1ID
INVOKE CreateThread, NULL, 0, OFFSET Thread2, 0,0,OFFSET Thread2ID

INVOKE WaitForMultipleObjects, 2, OFFSET hEvent1, TRUE, INFINITE

INVOKE FreeConsole
INVOKE ExitProcess,0

END Start
```