

Temat zajęć	Wątki w systemie Windows
Zakres materiału	Wykorzystanie usług systemu Windows do tworzenia programów wielowątkowych

Materiał teoretyczny

- *wątki w systemie Windows*
- *muteksy w systemie Windows*
- *typ HANDLE*
- *funkcje `CreateThread()`, `GetCurrentThreadId()`, `WaitForSingleObject()`*
- *funkcje `CreateMutex()` i `ReleaseMutex()`*

Treść zadania

Napisać języku C program spełniający poniższe wymagania:

- program oblicza wartość liczby π wykorzystując wzor Leibniza:

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

- obliczenia wartości wyrazów ciągu wykonuje się używając danych typu `double`;
- program akceptuje dokładnie dwa argumenty wywołania; oba są liczbami całkowitymi i oznaczają odpowiednio: pierwszy – liczbę n wyrazów ciągu Leibniza przetwarzanych przez kod i liczbę w wątków, w których odbędzie się przetwarzanie; zakładamy, że: $1 < n < 1000000000$ oraz $1 < w < 100$;
- program powołuje do życia w wątków i przekazuje każdemu z nich inny zakres wyrazów ciągu do przeliczenia
- jeżeli n nie dzieli się bez reszty przez w , to $n-1$ pierwszych wątków otrzymuje równe co do rozmiaru zakresy wyrazów, a wątek ostatni dostaje zakres powiększony o resztę z tego dzielenia;
- każdy wątek rozpoczynając pracę wypisuje na `stdout` swój identyfikator, liczbę wyrazów w swoim zakresie i indeks pierwszego wyrazu;
- każdy wątek oblicza sumę wyrazów w swoim zakresie ciągu, a następnie aktualizuje wspólną, globalną zmienną, która w efekcie ma zawierać sumę wszystkich wyrazów wyliczonych przez wszystkie wątki (**uwaga!** aktualizacja tej zmiennej jest **sekcją krytyczną** i musi być chroniona muteksem);
- każdy z wątków na zakończenie pracy wyprowadza na `stdout` swój identyfikator i obliczoną sumę;
- program czeka na zakończenie wszystkich wątków, a następnie wypisuje pomnożoną przez 4 wartość wspólnej zmiennej globalnej (czyli sumy elementów) oraz czas jaki upłynął od włączenia pierwszego wątku do zakończenia ostatniego;

- następnie program jeszcze raz oblicza wartość π , tym razem bez użycia wątków, i na zakończenie wyświetla ponownie przeliczony i pomnożoną przez cztery sumę oraz czas, jaki zajęło ponowne mnożenie.
- Przykładowe wyjście z programu:

```
C:\>p8.exe 10000000 3
Thread #11776 size=33333333 first=0
Thread #14140 size=33333333 first=33333333
Thread #7416 size=33333334 first=66666666
Thread #14140 sum=-0.00000001125000011251
Thread #7416 sum=0.00000000125000003750
Thread #11776 sum=0.78539817089762087310
w/Threads: PI=3.14159264359048329140 time=1.87832s
wo/Threads: PI=3.14159264358932599492 time=4.8732s
```

Uwaga! Kod źródłowy programu (1 plik) musi zostać jako **załącznik** przesłany na adres sos1@wi.zut.edu.pl:

- plik z kodem źródłowym musi mieć nazwę: numer_indeksu.so.lab08.c (np. 66666.so.lab08.c),
- plik musi zostać wysłany z poczty uczelnianej (domena zut.edu.pl),
- temat maila musi mieć postać:
SO IN1 99X LAB08
gdzie 99X to numer grupy laboratoryjnej (np. SO IN1 20A LAB08),
- w pierwszych trzech liniach kodu źródłowego w komentarzach (każda linia komentowana osobno) musi znaleźć się:
 - informacja identyczna z zamieszczoną w temacie maila,
 - imię i nazwisko osoby wysyłającej maila,
 - adres e-mail, z którego wysłano wiadomośćnp.:

```
// SO IN1 20A LAB08
// Jan Nowak
// nj66666@zut.edu.pl
```

- e-mail nie może zawierać żadnej treści (tylko załącznik).

Dostarczone kody programów będą analizowane pod kątem wykrywania plagiatów. Niewysłanie wiadomości, wysłanie jej w formie niezgodnej z powyższymi wymaganiami lub wysłanie pliku, który nie będzie się kompilował i uruchamiał, będzie traktowane jako brak programu i skutkowało otrzymaniem za niego oceny niedostatecznej.