

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 akceptuje dokładnie dwa argumenty wywołania; oba są liczbami całkowitymi i oznaczają odpowiednio: pierwszy – liczbę n danych rzeczywistych przetwarzanych przez kod i liczbę w wątków, w których odbędzie się przetwarzanie; zakładamy, że: $0 < n < 1000000$ oraz $0 < w < n$;
- program tworzy tablicę danych typu *float* o rozmiarze n i wypełnia ją danymi według poniższego schematu:

```
srand(0);  
for(int i = 0; i < n; i++)  
    t[i] = 1000. * rand() / RAND_MAX;
```

- program powołuje do życia w wątków i przekazuje każdemu z nich wskaźnik na inny fragment tablicy t ;
- jeżeli n nie dzieli się bez reszty przez w , to $n-1$ pierwszych wątków otrzymuje równe co do rozmiaru fragmenty tablicy, a wątek ostatni dostaje fragment powiększony o resztę z tego dzielenia;
- każdy wątek rozpoczynając pracę wypisuje na *stdout* swój identyfikator i liczbę elementów w swoim fragmencie tablicy;
- każdy wątek oblicza **sumę** elementów w swoim fragmencie tablicy, a następnie aktualizuje wspólną, globalną zmienną, która w efekcie powinna zawierać sumę elementów całej tablicy (**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 częściową;
- program czeka na zakończenie wszystkich wątków, a następnie wypisuje wartość wspólnej zmiennej globalnej (czyli łącznej sumy elementów) oraz czas jaki upłynął od włączenia pierwszego wątku do zakończenia ostatniego;

- następnie program jeszcze raz sumuje dane w tablicy, tym razem bez użycia wątków, i na zakończenie wyświetla ponownie przeliczoną sumę i czas jaki zajęło ponowne sumowanie.
- Przykładowe wyjście z programu:

```
C:\>p10 1000 3
Thread #2196 size=333
Thread #2204 size=334
Thread #2196 sum=171071.296875
Thread #2204 sum=172265.265625
Thread #2180 size=333
Thread #2180 sum=176582.937500
w/Threads: sum=519919.500000, time=2.939sec
wo/Threads: sum=519919.500000, time=5.88sec
519919.500000
```

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.lab11.c` (np. `66666.so.lab11.c`),
- plik musi zostać wysłany z poczty uczelnianej (domena `zut.edu.pl`),
- temat maila musi mieć postać:
`SO IS1 999X LAB11`
gdzie `999X` to numer grupy laboratoryjnej (np. `SO IS1 210A LAB11`),
- 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 IS1 210A LAB11
// 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 uruchamiać, będzie traktowane jako brak programu i skutkowało otrzymaniem za niego oceny niedostatecznej.