

Temat zajęć	Wątki w systemie Linux
Zakres materiału	Biblioteka <i>pthread</i> i jej wykorzystanie do tworzenia programów wielowątkowych

Materiał teoretyczny

- wątki w Linuksie
- muteksy w Linuksie
- typ *pthread_t*
- funkcje *pthread_create()*, *pthread_join()*, *pthread_self()*
- funkcje *mutex_lock()* i *mutex_unlock()*
- pomiar czasu, funkcja *clock()* i stała *CLOCKS_PER_SEC*

Treść zadania

Napisać języku C program spełniający poniższe wymagania:

- program oblicza wartość liczby π korzystając ze wzoru Wallisa:

$$\prod_{n=1}^{\infty} \frac{(2n)(2n)}{(2n-1)(2n+1)} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \dots = \frac{\pi}{2}$$

- 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 Wallisa 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 **iloczyn** wyrazów w swoim zakresie ciągu, a następnie aktualizuje wspólną, globalną zmienną, która w efekcie ma zawierać iloczyn 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 obliczony iloczyn;

- program czeka na zakończenie wszystkich wątków, a następnie wypisuje pomnożoną przez 2 wartość wspólnej zmiennej globalnej (czyli iloczynu 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 podwojony iloczyn oraz czas, jaki zajęło ponowne mnożenie.
- Przykładowe wyjście z programu:

```
$ ./p7 100000000 3
Thread #140203307648576 size=333333333 first=1
Thread #140203299255872 size=333333333 first=333333334
Thread #140203290863168 size=333333334 first=666666667
Thread #140203307648576 prod=1.57079631487859461281
Thread #140203290863168 prod=0.99999999087313684587
Thread #140203299255872 prod=0.99999999112312021321
w/Threads: PI=3.14159257319676354925 time=2.02345s
wo/Threads: PI=3.14159255774221746194 time=5.92134s
```

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.lab07.c` (np. `66666.so.lab07.c`),
- plik musi zostać wysłany z poczty uczelnianej (domena `zut.edu.pl`),
- temat maila musi mieć postać:
`SO IN1 99X LAB07`
gdzie `99X` to numer grupy laboratoryjnej (np. `SO IS1 20A LAB07`),
- 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 LAB07
// 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.