

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_gettime()*

Treść zadania

Napisać języku C program spełniający poniższe wymagania:

- program oblicza wartość liczby π korzystając z poniższego ciągu:

$$\pi = 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \dots$$

- 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 przetwarzanych przez kod i liczbę w wątków, w których odbędzie się przetwarzanie; zakładamy, że: $1 < n < 1000000$ oraz $1 < w < \min(100, n)$;
- 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 swojego zakresu 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 na *stdout* obliczoną wartość π z dokładnością do 20 miejsc po przecinku oraz czas jaki upłynął od włączenia pierwszego wątku do zakończenia ostatniego; wybierz adekwatny wariant wywołania funkcji *clock_gettime()*.
- następnie program jeszcze raz oblicza wartość π , tym razem bez użycia wątków, i na zakończenie wyświetla obliczone π oraz czas, jaki zajęło ponowne obliczenie.
- Przykładowe wyjście z programu:

```
$ ./p11 200000 3
Thread 7fbaf90fa6c0 size=6666 first==0
Thread 7fbaf88f96c0 size=6666 first==6666
Thread 7fbaf80f86c0 size=6668 first==13332
Thread 7fbaf90fa6c0 sum=0.14159265358894881914
Thread 7fbaf88f96c0 sum=0.00000000000073814677
Thread 7fbaf80f86c0 sum=0.0000000000007423135
w/Threads PI=3.14159265358976114157 time=0.000566
wo/Threads PI=3.14159265358975625659 time=0.000561
```

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.lab10.c` (np. `66666.so.lab10.c`),
- plik musi zostać wysłany z poczty uczelnianej (domena `zut.edu.pl`),
- temat maila musi mieć postać:
SO IS1 999X LAB10
gdzie 999X to numer grupy laboratoryjnej (np. SO IS1 210A LAB10),
- 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 LAB10
// 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.