

Cel:

1. Nabycie umiejętności tworzenia i implementacji programów równoległych

Zajęcia:

1. Utworzenie katalogu roboczego (np. lab\_5), skopiowanie ze strony przedmiotu paczki pthreads\_suma.tgz, rozpakowanie w podkatalogu (np. zad\_1), uruchomienie i przetestowanie działania dla różnej liczby wątków (w miejsce dostarczonej biblioteki libpomiar\_czasu.a należy umieścić własną bibliotekę utworzoną w ramach laboratorium „pomiar\_czasu”) - sprawdzenie poprawności wyniku i czasu działania (wnioski powinny znaleźć się w sprawozdaniu)

2. Opracowanie (np. na bazie pliku pthreads\_suma.c) programu wykorzystującego bibliotekę wątków Pthreads i obliczającego na kilka sposobów przybliżoną wartość całki oznaczonej z zadanej funkcji metodą trapezów.

a) program powinien zawierać funkcję obliczającą całkę sekwencyjnie (można użyć np. kod:

```
x1 = a; c=0.0; dx = (b-a)/N;
for (i=0; i<N; i++){
    x2=x1+dx;
    c+=0.5*(f(x1)+f(x2))*dx;
    x1=x2;
}
```

uwaga: w kodzie każda wartość funkcji (z wyjątkiem skrajnych) jest liczona dwa razy, a mnożenie przez stałą podstawę odbywa się wielokrotnie w sumowaniu, czy można zmodyfikować kod, tak żeby liczyć każdą wartość tylko raz, a dodatkowo zmniejszyć liczbę operacji wyciągając mnożenie przed sumowanie?)

b) poza wariantem obliczania całki sekwencyjnie (w celu sprawdzenia poprawności kodu równoległego) program powinien zawierać dwa warianty równoległego obliczania całki: jeden związany z dekompozycją w dziedzinie problemu i drugi realizujący zrównoleglenie pętli (wariant pierwszy ma zakładać, że każdy wątek otrzymuje jako daną strukturę zawierającą parametry sobie przydzielonego przedziału, wariant drugi, że każdy wątek otrzymuje swój identyfikator, a dekompozycja odbywa się w sposób analogiczny jak w przykładzie sumowania na slajdach z wykładów). W jednym wariantcie można wykorzystać redukcję wielowątkową, a w drugim jednowątkową z wykorzystaniem globalnej tablicy wyników częściowych.

3. Uruchomienie procedury sekwencyjnej; sprawdzenie poprawności otrzymanego wyniku dla różnych dokładności zadawanych przez użytkownika (poprzez wielkość podstawy trapezów czyli także całkowitą liczbę trapezów w zadanym (globalnym) przedziale)

- należy zanalizować jaka jest dokładność całkowania (podstawa trapezów), dla każdego z wariantów całkowania – **rezultaty opisać we wnioskach sprawozdania**

4. W przypadku zrównoleglenia pętli opracowanie sposobu umożliwiającego równomierny podział iteracji między wątki. Jak rozwiązać problem, w przypadku gdy liczba trapezów jest niepodzielna przez liczbę wątków (nie zmieniając granic całkowania i nie zaniżając dokładności numerycznej)?

5. Uruchomienie programu równoległego (na np. 2, 3 i 4 wątkach); sprawdzenie poprawności działania.

Dalsze kroki dla podniesienia oceny:

1. Dodanie procedur pomiaru czasu i uzyskanie czasu wykonania dla 1,2,3,...,8 wątków oraz dla dwóch rozmiarów zadań:  $n=10000$  ( $10^4$ ) i  $n=10000000$  ( $10^7$ )

2. Stworzenie, dla obu rozmiarów zadania, wykresów zależności czasu wykonania od liczby wątków

Warunki zaliczenia:

1. Obecność na zajęciach i wykonanie kroków 1-5.

2. Oddanie sprawozdania z opisem zadania, kodem źródłowym programów i analizą wyników dla wszystkich wariantów oraz odpowiednimi wnioskami.