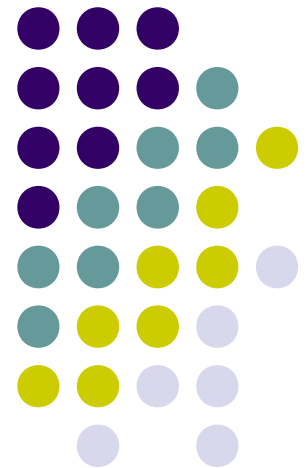


Średniozaawansowane programowanie w C++

Program #4
3 listopada 2016 r.
na
1 grudnia 2016 r.



Zadanie



Napisać program mnożący wielowątkowo macierze.

Zbadaj czas mnożenia macierzy w zależności od liczby tworzonych wątków.

Uruchomienie programu:

```
./program03 macierz1.mtrx macierz2.mtrx wynik.mtrx
```

Podział na jednostki kompilacji



1. main - dosyć długi

- Obsługa parametrów startowych programu
- Wczytanie macierzy z plików
- Wywołanie funkcji mnożącej
- Zmierzenie czasu trwania powyższej funkcji
- Zapisanie macierzy wynikowej do pliku

2. macierz

class Macierz – podstawowe konstruktory (brakuje kopiującego!),
destruktory, metody trywialne, zapis macierzy do strumienia

3. mnozenie_macierzy

class KontrolerMnozenia – kontroluje proces mnożenia macierzy
(**obiekt wspoldzielony przez wątki!**)
class MnozycielMacierzy – funktor wykorzystywany w wątkach
mnozacych macierze

main.cpp – fragmenty



```
const clockid_t ZegarID = CLOCK_REALTIME;
timespec start, koniec; // Struktury przechowujące czas

std::cout << "Mnozenie macierzy...";
std::cout.flush ();

clock_gettime (ZegarID, &start); // Zapisuje czas początku mnożenia

// Mnoży macierze a i b (liczba oznacza liczbę równoległych wątków)
std::auto_ptr<Macierz> wynik = mnozenie_macierzy (a, b, 2);

clock_gettime (ZegarID, &koniec); // Zapisuje czas końca mnożenia

std::cout << " zajelo " << interwał (start, koniec) << " s" << std::endl;
```

macierz.hpp



```
typedef std::pair <int, int> RozmiarMacierzy;
typedef std::pair <int, int> Położenie;

class Macierz
{
public:
    Macierz (std::istream &is);    // tworzy macierz na podstawie strumienia
    Macierz (int n, int m);        // tworzy pustą (niezainicjowaną) macierz n x m
    ~Macierz ();
    int element (int i, int j) const; // zwraca element macierzy (kopia przez wartość)
    int &element (int i, int j);    // zwraca ref. (można modyfikować!) do el. macierzy
    RozmiarMacierzy rozmiar () const;    // zwraca rozmiar macierzy

private:
    int n_;
    int m_;
    int *mac_;

    friend std::ostream &operator<< (std::ostream &os, const Macierz &m);
};
```

mnozenie_macierzy.hpp (1)



```
// Kontroluje proces mnożenia macierzy (obiekt współdzielony)
class KontrolerMnozenia
{
public:
    // Inicjuje składowe klasy
    KontrolerMnozenia (const Macierz &m1, const Macierz &m2, Macierz &m3);
    inline const Macierz &czynnik1 () const {return m1_;}
    inline const Macierz &czynnik2 () const {return m2_;}
    inline Macierz &wynik () const {return m3_;}

    // Zwraca współrzędne najbliższego elementu do obliczeń
    // lub kontener niezainicjowany, gdy obliczenia dobiegły końca
    boost::optional <Polozenie> coMamLiczyć ();

private:
    boost::mutex blokada_;           // blokada
    const Macierz &m1_;             // pierwszy czynnik (stała referencja)
    const Macierz &m2_;             // drugi czynnik (stała referencja)
    Macierz &m3_;                   // wynik mnożenia
};
```


Programowanie jest fantastyczne!!!

