

Programowanie II R

zestaw zadań 11

Biblioteka future, czyli programowanie asynchroniczne[w tle]

12 czerwca 2025

Strona wykładowcy <http://glach.wikidot.com/p2r>

Zadanie 1 z*: Liczenie liczby π metodą ciągu Leibniza

Wykorzystaj bibliotekę future

1. Napisz funkcję liczącą elementy Ciągu Leibniza od zadanego elementu k_1 do k_2
2. Sprawdź ile masz dostępnych wątków w procesorze (komenda `htop` na przykład)
3. Wyznacz maksymalny wyraz ciągu N i podziel ciąg na tyle części ile wątków.
4. Wykorzystując bibliotekę `futures` rozdziel wątki do procesowania w tle:
`async(launch::async,nazwa_funkcji,parametry oddzielone przecinkami)`
5. Zsumuj wszystkie wyniki i pomnóż razy 4: otrzymasz liczbę π , porównaj ją z oryginałem, oblicz błąd bezwzględny i względny[procentowo]

Podciągi całego ciągu możesz wrzucić do:

```
vector<future<double>> futures;
```

Za powyższe otrzymasz 2 punkty, dodatkowe punkty:

[+1]-implementacje innych metod liczenia π i porównanie czasu liczenia ich z ciągiem Leibniza [ciąg Chudnowski i Gregory]
[+1]-zaimplementuj sprawdzenie ile masz rdzeni żeby sprawdzało automatycznie i liczyło z maksymalną liczbą rdzeni, sprawdź jak dużo bajtów masz w pamięci RAM: jak potencjalnie dużą liczbę cyfr liczby π możesz w niej zmieścić?
Jak to zrobiło Google

Jak to zrobiło Linus Tech tips [obecny rekord 2025.06.12]

Na dalszą drogę w programowaniu/kodowaniu:

Mario 64: Kaze Emanuar Kanał na temat optymalizacji kodu wykorzystujący matematykę i znajomość budowy konsoli nintendo 64, bardzo dokładne opisy:

Jak efektywnie i tanio liczyć funkcję sinus i cosinus? oraz Tu filmik z jeszcze lepszą optymalizacją

Kanał, gdzie można znaleźć różne wyzwania w kodowaniu w tym fizyczne, raczej w javascript, ale bardzo fajnie tłumaczone

Symulacja podwójnego wachadła klasycznego

Tsoding: Kanał gdzie można znaleźć w zasadzie bardzo mocne skupienie na kodowaniu, niemniej w bardzo wielu językach, długie nagrania z live

Tu w c++ krzywe bézier inny ciekawy temat Hash tables

Kanał z różnymi filmikami ale skupiony w dużej mierze na C++: jak programować kolizje kul (cząstek)?:

tu macie więcej przykładów wymienionych, ale ich nie oglądałem

Powodzenia