

IVS-Calculator - protokol k profilingu

Profiling programu stddev.cpp sme urobili pomocou gprof.

Profilovali sme pre 1,000,000 vstupov, čo malo viditeľnejší dopad na čas strávený vo funkciách.

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self us/call	total us/call	name
45.51	0.05	0.05	1000080	0.05	0.05	CustomMath::power(double, double)
27.31	0.08	0.03	3000000	0.01	0.01	CustomMath::addition(double, double)
27.31	0.11	0.03				main
0.00	0.11	0.00	2	0.00	0.00	CustomMath::subtraction(double, double)
0.00	0.11	0.00	2	0.00	0.00	CustomMath::multiplication(double, double)
0.00	0.11	0.00	2	0.00	0.00	CustomMath::division(double, double)
0.00	0.11	0.00	1	0.00	0.00	_GLOBAL__sub_I_2N10CustomMath8additionEdd
0.00	0.11	0.00	1	0.00	0.00	_GLOBAL__sub_I_main
0.00	0.11	0.00	1	0.00	0.00	__static_initialization_and_destruction_0(int, int)
0.00	0.11	0.00	1	0.00	0.00	__static_initialization_and_destruction_0(int, int)
0.00	0.11	0.00	1	0.00	3.95	CustomMath::root(double, double)

Usúdili sme, že funkcia power na výpočet mocniny, ktorú sme použili na výpočet druhej mocniny, zaberala značné množstvo času a teda by bola vhodným kandidátom na optimalizáciu. Možným spôsobom optimalizácie stddev by bolo namiesto funkcie power použiť funkciu multiplication, ktorá je časovo menej náročná.

```
#include <stdlib.h>
#include <iostream>
#include "../src/CustomMath.h"

int main() {
    double tmp = 0;
    double sum = 0;
    double sqrsum = 0;
    int n = 0;

    CustomMath u;

    while(!feof(stdin)) {
        std::cin >> tmp;
        sum = u.addition(sum, tmp);
        sqrsum = u.addition(sqrsum, u.power(tmp, 2));
        n = u.addition(n, 1);
    }

    double avg = u.division(sum, n);

    double result = u.subtraction(n, 1);
    result = u.division(1, result);
    double result2 = u.power(avg, 2);
    result2 = u.multiplication(n, result2);
    result2 = u.subtraction(sqrsum, result2);
    result = u.multiplication(result, result2);
    result = u.root(result, 2);

    std::cout << result << std::endl;

    return 0;
}
```