

Profiling

V rámci profilingu byl využit profilovací nástroj VisualStudio. Profiling byl proveden pro vstupy o velikosti 10, 100 a 1000 hodnot

Výstup profileru pro 10 čísel

Název funkce	Počet volání	% uplynulého celkového...	% uplynulého výhradního...	Průměrný uplynulý celkový...	Průměrný uplynulý výhrad...
main	1	78,69 %	0,02 %	2 906 840,00	725,00
profiling	1	57,76 %	0,19 %	2 133 771,00	6 910,00
fopen	1	35,10 %	35,10 %	1 296 766,00	1 296 766,00
fscanf	11	21,24 %	0,13 %	71 326,00	447,00
MaafLajbrry::rad	1	0,86 %	0,22 %	31 641,00	8 226,00
MaafLajbrry::add	20	0,18 %	0,16 %	327,00	294,00
MaafLajbrry::pow	11	0,12 %	0,12 %	410,00	396,00
MaafLajbrry::divid	2	0,03 %	0,03 %	498,00	471,00
MaafLajbrry::MaafLajbrry	1	0,01 %	0,01 %	367,00	364,00
MaafLajbrry::mul	1	0,01 %	0,01 %	367,00	365,00
MaafLajbrry::sub	1	0,01 %	0,01 %	365,00	363,00
MaafLajbrry::~MaafLajbrry	1	0,01 %	0,01 %	365,00	362,00

Výstup profileru pro 100 čísel

Název funkce	Počet volání	% uplynulého celkového...	% uplynulého výhradního...	Průměrný uplynulý celkový...	Průměrný uplynulý výhrad...
main	1	82,98 %	0,01 %	6 154 075,00	643,00
profiling	1	67,77 %	1,25 %	5 026 511,00	93 035,00
fscanf	101	36,06 %	0,49 %	26 476,00	359,00
fopen	1	27,37 %	27,37 %	2 029 647,00	2 029 647,00
MaafLajbrry::add	200	1,85 %	0,78 %	685,00	288,00
MaafLajbrry::pow	101	0,60 %	0,51 %	437,00	375,00
MaafLajbrry::rad	1	0,57 %	0,13 %	42 629,00	9 753,00
MaafLajbrry::divid	2	0,03 %	0,03 %	986,00	966,00
MaafLajbrry::sub	1	0,02 %	0,02 %	1 191,00	1 191,00
MaafLajbrry::~MaafLajbrry	1	0,01 %	0,01 %	856,00	856,00
_RTC_CheckEsp	2	0,01 %	0,01 %	263,00	263,00
_RTC_CheckStackVars	1	0,01 %	0,01 %	438,00	438,00
MaafLajbrry::MaafLajbrry	1	0,01 %	0,01 %	390,00	390,00
MaafLajbrry::mul	1	0,00 %	0,00 %	363,00	363,00

Výstup profileru pro 1000 čísel

Název funkce	Počet volání	% uplynulého celkového...	% uplynulého výhradního...	Průměrný uplynulý celkový...	Průměrný uplynulý výhrad...
main	1	78,67 %	0,00 %	14 963 555,00	511,00
profiling	1	72,57 %	2,31 %	13 802 407,00	440 083,00
fscanf	1 001	55,59 %	2,27 %	10 562,00	431,00
fopen	1	7,50 %	7,50 %	1 426 693,00	1 426 693,00
MaafLajbrry::add	2 000	4,41 %	3,44 %	419,00	327,00
MaafLajbrry::pow	1 001	2,50 %	2,03 %	474,00	386,00
MaafLajbrry::rad	1	0,24 %	0,05 %	46 026,00	10 265,00
MaafLajbrry::divid	2	0,01 %	0,01 %	604,00	579,00
_RTC_CheckStackVars	1	0,00 %	0,00 %	442,00	442,00
MaafLajbrry::MaafLajbrry	1	0,00 %	0,00 %	388,00	388,00
MaafLajbrry::mul	1	0,00 %	0,00 %	374,00	374,00
MaafLajbrry::sub	1	0,00 %	0,00 %	373,00	373,00
MaafLajbrry::~MaafLajbrry	1	0,00 %	0,00 %	371,00	371,00

Nejvíce času tráví program funkcí “fscanf”, která je volána pro každé číslo a provádí konverzi z textové na číselnou hodnotu. Pro zrychlení by bylo lepší načítat hodnoty rovnou jako čísla z binárního souboru.

Z námi implementovaných funkcí je časově nejnáročnější odmocnina (rad), ale ta je vždy volána jen jednou, pro jakýkoli počet hodnot. Pro vstup více čísel by nemělo smysl ji optimalizovat. Naopak

volání funkcí mocnina (pow) a součet (add) jsou závislá na množství vstupních hodnot. Pro vstup 1000 čísel je vidět, že program spotřebuje sčítáním mnohem víc času než odmocňováním.

