

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
VYSOKÉ UČENÍ TECHNICKÉ

Profiling implementácie tried fib-sequence.rb a fib-sequence\_slow.rb v Ruby

Ciel': Profiling implementácie tried fib-sequence.rb a fib-sequence\_slow.rb v Ruby

Nástroj: ruby rprofile

Poznámka: Profiling bol prevedený u 20. a 20 000. člena Fibonacciho postupnosti

## 20. člen Fibonacciho postupnosti

Dáta z nástrojov rprofile a time pre triedu fib-sequence.rb a fib-sequence\_slow.rb

ruby -rprofile fib-sequence.rb								ruby -rprofile fib-sequence_slow.rb							
6765	%	cumulative	self	self	total			6765	%	cumulative	self	self	total		
time	seconds	seconds	calls	ms/call	ms/call	name		time	seconds	seconds	calls	ms/call	ms/call	name	
0.00	0.00	0.00	2	0.00	0.00	IO#set_encoding		0.00	0.00	0.00	2	0.00	0.00	IO#set_encoding	
0.00	0.00	0.00	1	0.00	0.00	Class#inherited		0.00	0.00	0.00	1	0.00	0.00	Class#inherited	
0.00	0.00	0.00	6	0.00	0.00	Module#method_added		0.00	0.00	0.00	6	0.00	0.00	Module#method_added	
0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#reset		0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#reset	
0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#initialize		0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#initialize	
0.00	0.00	0.00	1	0.00	0.00	Class#new		0.00	0.00	0.00	1	0.00	0.00	Class#new	
0.00	0.00	0.00	23	0.00	0.00	Fixnum#<		0.00	0.00	0.00	23	0.00	0.00	Fixnum#<	
0.00	0.00	0.00	1	0.00	0.00	Fixnum#<=		0.00	0.00	0.00	1	0.00	0.00	Fixnum#<=	
0.00	0.00	0.00	40	0.00	0.00	Fixnum#+		0.00	0.00	0.00	211	0.00	0.00	Fixnum#+	
0.00	0.00	0.00	62	0.00	0.00	Array#[]		0.00	0.00	0.00	24	0.00	0.00	Array#[]	
0.00	0.00	0.00	2	0.00	0.00	Kernel.nil?		0.00	0.00	0.00	2	0.00	0.00	Kernel.nil?	
0.00	0.00	0.00	21	0.00	0.00	FibonacciSequence#next		0.00	0.00	0.00	21	0.00	0.00	FibonacciSequence#next	
0.00	0.00	0.00	19	0.00	0.00	NilClass#nil?		0.00	0.00	0.00	19	0.00	0.00	NilClass#nil?	
0.00	0.00	0.00	38	0.00	0.00	Fixnum#-		0.00	0.00	0.00	19	0.00	0.00	Fixnum#-	
0.00	0.00	0.00	19	0.00	0.00	Array#[]=		0.00	0.00	0.00	19	0.00	0.00	Range#each	
0.00	0.00	0.00	1	0.00	0.00	Fixnum#>=		0.00	0.00	0.00	19	0.00	0.00	Array#[]=	
0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#current		0.00	0.00	0.00	1	0.00	0.00	Fixnum#>=	
0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#[]		0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#current	
0.00	0.00	0.00	1	0.00	0.00	Fixnum#to_s		0.00	0.00	0.00	1	0.00	0.00	FibonacciSequence#[]	
0.00	0.00	0.00	1	0.00	0.00	IO#write		0.00	0.00	0.00	1	0.00	0.00	Fixnum#to_s	
0.00	0.00	0.00	1	0.00	0.00	Kernel.print		0.00	0.00	0.00	1	0.00	0.00	IO#write	
0.00	0.01	0.00	1	0.00	10.00	#toplevel		0.00	0.00	0.00	1	0.00	0.00	Kernel.print	
								0.00	0.01	0.00	1	0.00	10.00	#toplevel	
real	0m0.014s							real	0m0.015s						
user	0m0.009s							user	0m0.015s						
sys	0m0.006s							sys	0m0.001s						

## 20 000. člen Fibonacciho postupnosti

```

ruby -rprofile fib-sequence.rb
% cumulative self      total
time seconds  seconds  calls ms/call ms/call name
56.98 1.02 1.02 20001 0.05 0.08 FibonacciSequence#next
12.29 1.24 0.22 1 220.00 1790.00 FibonacciSequence#[]
8.94 1.40 0.16 60002 0.00 0.00 Array#[]
5.59 1.50 0.10 19999 0.01 0.01 Array#[]
4.47 1.58 0.08 39998 0.00 0.00 Fixnum#-
3.35 1.64 0.06 19999 0.00 0.00 Fixnum#+
3.35 1.70 0.06 19999 0.00 0.00 NilClass#nil?
3.35 1.76 0.06 20001 0.00 0.00 Fixnum#<
1.68 1.79 0.03 20003 0.00 0.00 Fixnum#<=
0.00 1.79 0.00 1 0.00 0.00 Class#inherited
0.00 1.79 0.00 2 0.00 0.00 Kernel.nil?
0.00 1.79 0.00 6 0.00 0.00 Module#method_added
0.00 1.79 0.00 1 0.00 0.00 Fixnum#<=
0.00 1.79 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 1.79 0.00 1 0.00 0.00 Class#new
0.00 1.79 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 1.79 0.00 1 0.00 0.00 Fixnum#>=
0.00 1.79 0.00 1 0.00 0.00 FibonacciSequence#current
0.00 1.79 0.00 2 0.00 0.00 IO#set_encoding
0.00 1.79 0.00 1 0.00 0.00 #toplevel

real 0m1.889s
user 0m1.791s
sys 0m0.100s

```

```

ruby -rprofile fib-sequence_slow.rb
% cumulative self      total
time seconds  seconds  calls ms/call ms/call name
73.76 13.83 13.83 1999 6.92 9.29 Range#each
23.41 18.22 4.39 1821186 0.00 0.00 Bignum#+
1.97 18.59 0.37 179815 0.00 0.00 Fixnum#+
0.64 18.71 0.12 2001 0.06 9.35 FibonacciSequence#next
0.16 18.74 0.03 1 30.00 18750.00 FibonacciSequence#[]
0.05 18.75 0.01 2003 0.00 0.00 Fixnum#<
0.00 18.75 0.00 2 0.00 0.00 Kernel.nil?
0.00 18.75 0.00 1 0.00 0.00 Fixnum#<=
0.00 18.75 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 18.75 0.00 2004 0.00 0.00 Array#[]
0.00 18.75 0.00 1 0.00 0.00 Class#new
0.00 18.75 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 18.75 0.00 1999 0.00 0.00 NilClass#nil?
0.00 18.75 0.00 1999 0.00 0.00 Fixnum#-
0.00 18.75 0.00 6 0.00 0.00 Module#method_added
0.00 18.75 0.00 1999 0.00 0.00 Array#[]
0.00 18.75 0.00 1 0.00 0.00 Class#inherited
0.00 18.75 0.00 1 0.00 0.00 Fixnum#>=
0.00 18.75 0.00 1 0.00 0.00 FibonacciSequence#current
0.00 18.75 0.00 1 0.00 0.00 FibonacciSequence#current
0.00 18.75 0.00 2 0.00 0.00 IO#set_encoding
0.00 18.75 0.00 1 0.00 18750.00 #toplevel

real 0m19.609s
user 0m18.764s
sys 0m0.855s

```

**Zhodnotenie:** Profiling bol prevedený nad triedami fib-sequence.rb a fib-sequence\_slow.rb za účelom porovnanie efektívnosti výpočtov členov Fibonacciho postupnosti. V pomalšej metóde (v triede fib-sequence\_slow) výpočtov členov si prvky vypočítavame úplne odznova, nepoužívame tie, čo sme už vypočítali a sú v poli (rozdiel oproti normálnej – rýchlej metóde v triede fib-sequence). Ako je možné vidieť, pri získavaní 20. člena ešte časový rozdiel nebol citeľný, no v profilingu už si môžeme všimnúť vzrastajúci počet sčítaní, ktorý prebehol u pomalšej metódy. Počet volaní funkcie next sa v oboch implementáciách rovná. U 20 000. člena sa spomalenie a neefektivita prejavilo naplno, časový medzi metódami bol skoro 18 sekúnd. U pomalšej metódy sa taktiež uskutočnilo vyše milión sčítaní – oproti skoro 20 000 sčítaniam v normálnej implementácii. V triede fib-sequence je teda výpočet efektívnejší oproti triede fib-sequence\_slow, keďže sa znovu využívajú vypočítané členy Fibonacciho postupnosti.