Ruby master - Feature #1256
Add constant TAILRECURSION to let a program recognize if tail recursion optimization is implemented
03/09/2009 02:43 AM - WoNaDo (Wolfgang Nádasi-Donner)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
</tbody>
</table>

Description
=begin
Ruby 1.9 has the possibility to support tail recursion optimization for methods and proc objects. Due to technical problems it cannot be easy implemented for every VM or platform, which will support Ruby 1.9 functionality.

Tail recursion optimization is an implementation detail and not a language feature in general. I propose to supply a constant TAILRECURSION, which will be set to "true" if this feature is implemented, otherwise "false".
=end

History
#1 - 03/09/2009 03:13 AM - shyouhei (Shyouhei Urabe)
=begin
I'm curious, how you can be happy if you detect your VM supports tail-call optimization or not (from your script)?

I think you have to write without tail-calls anyway for compatibilities with those no-tail-call environment.
=end

#2 - 03/09/2009 03:39 AM - WoNaDo (Wolfgang Nádasi-Donner)
=begin
Shyouhei Urabe schrieb:

I'm curious, how you can be happy if you detect your VM supports tail-call optimization or not (from your script)?

I think you have to write without tail-calls anyway for compatibilities with those no-tail-call environment.

I don't always write portable programs, but I want a controlled end of such a programm, e.g. with a message "Programm cannot run under this environment".

It's the same as I do in the moment. When started using Ruby 1.8 they tell the user, that it will not work using Ruby 1.8.

This is better than a system error message.

Otherwise it is also possible to present a different implementation, which may be much slower, but works and inform the user via a warning about this fact.

Wolfgang Nádasi-Donner
=end

#3 - 03/09/2009 09:53 AM - nobu (Nobuyoshi Nakada)
=begin
Hi,

At Mon, 9 Mar 2009 02:41:55 +0900, Wolfgang Nádasi-Donner wrote in [ruby-core:22748]:

Tail recursion optimization is an implementation detail and not a language feature in general. I propose to supply a constant TAILRECURSION, which will be set to "true" if this feature is implemented, otherwise "false".
At Mon, 9 Mar 2009 02:41:55 +0900, Wolfgang Nádasi-Donner wrote in [ruby-core:22748]:

Tail recursion optimization is an implementation detail and not a language feature in general. I propose to supply a constant TAILRECURSION, which will be set to "true" if this feature is implemented, otherwise "false".

RubyVM::InstructionSequence.compile_option[:tailcall_optimization]

Thank you for this information. I saw that it still exists in Ruby 1.9.1-p0.

Yesterday I tried to compile Ruby 1.9.1-p0 with changes named by Nobuyoshi Nakada. In vm_opts.h I changed

#define OPT_TAILCALL_OPTIMIZATION    0

to

#define OPT_TAILCALL_OPTIMIZATION    1

Afterwards I rebuilt Ruby, but the stack overflow for both examples I posted before are still there.

Should I change additional things in the source?

Wolfgang Nádasi-Donner

As far as I understand this Ticket can be closed. If RubyVM::InstructionSequence.compile_option exists (defined? RubyVM::InstructionSequence.compile_option) and the value is "true" a program can expect a working tail recursion optimization, otherwise not.

Is this correct? - I yes, I think thee is no additional need for an extra constant.

Afterwards I rebuilt Ruby, but the stack overflow for both examples I posted before are still there.

Should I change additional things in the source?

Wolfgang Nádasi-Donner

trace_instruction option prevents the optimization.

$ cat ~/tmp/tailcall.rb
#!/usr/bin/ruby
src, file, line = < e
puts "#{e.class} #{e.backtrace.size}"
$ ./ruby ~/tmp/tailcall.rb
tailcall_optimization: false, trace_instruction: true
SystemStackError 8187

$ ./ruby ~/tmp/tailcall.rb true
tailcall_optimization: true, trace_instruction: true
SystemStackError 8187

$ ./ruby ~/tmp/tailcall.rb true false
tailcall_optimization: true, trace_instruction: false
ok

--
Nobu Nakada

=end

#7 - 03/10/2009 11:42 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Rejected

=begin
=end