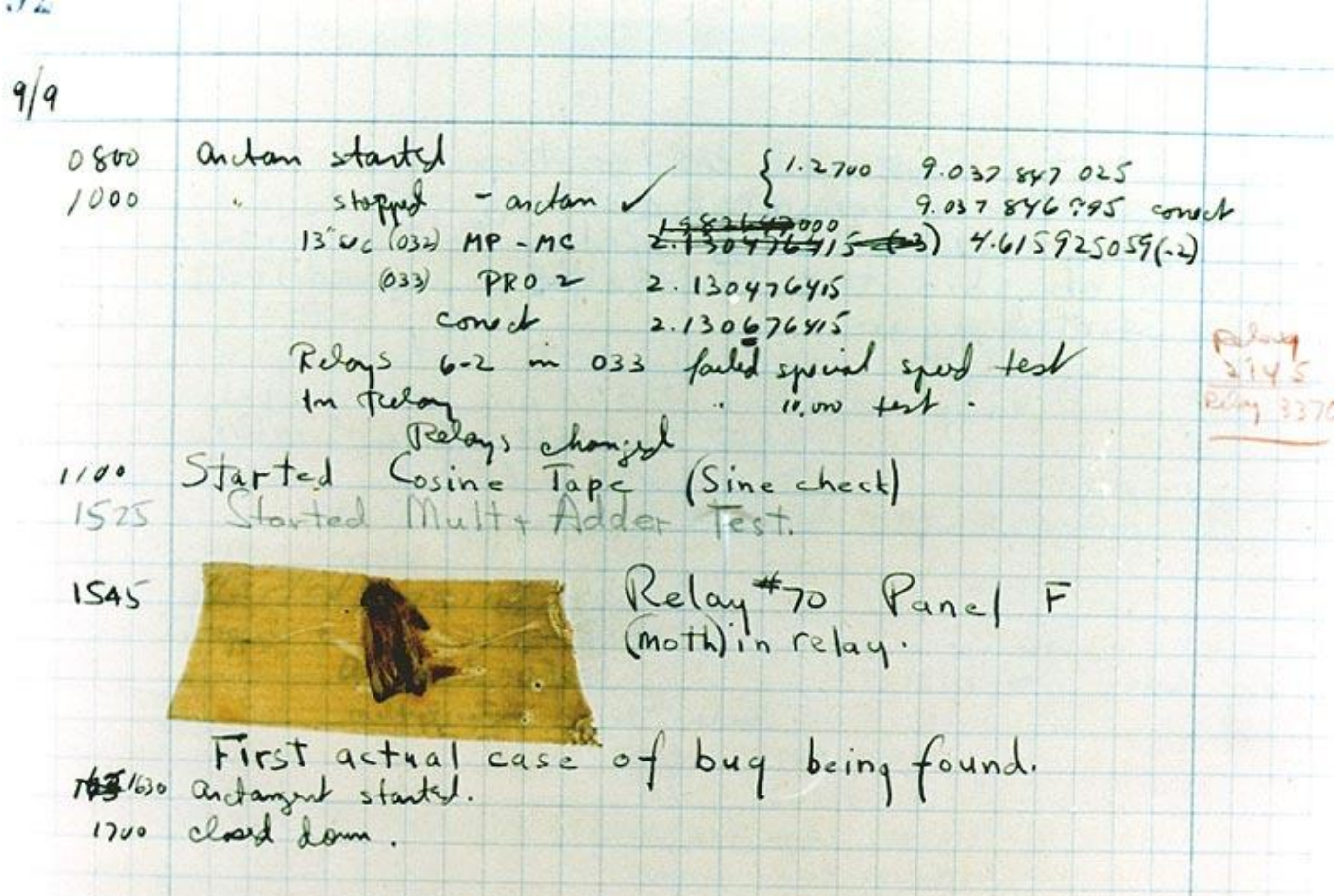
Presented 05 June 2019@CSRug Ruby Meetup (first Wednesday in the month)

Debugging in Ruby

IDE => puts => … ?

=)K



9/9

0800 Arctan started

1000 " stopped - arctan ✓ {1.2700 9.037 847 025

9.037 846 995 correct

13" sc (033) MP - MC ~~1.982147000~~ ~~2.130476415 (-3)~~ 4.615925059(-2)

(033) PRO 2 2.130476415

correct 2.130676415

Relays 6-2 in 033 failed special speed test

In relay " 11.000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545 Relay #70 Panel F

(moth) in relay.

First actual case of bug being found.

~~1525~~ 1630 Antangent started.

1700 closed down.

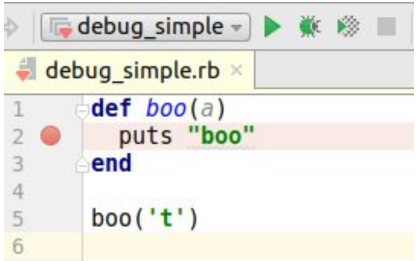
Relay 2145

Relay 3370

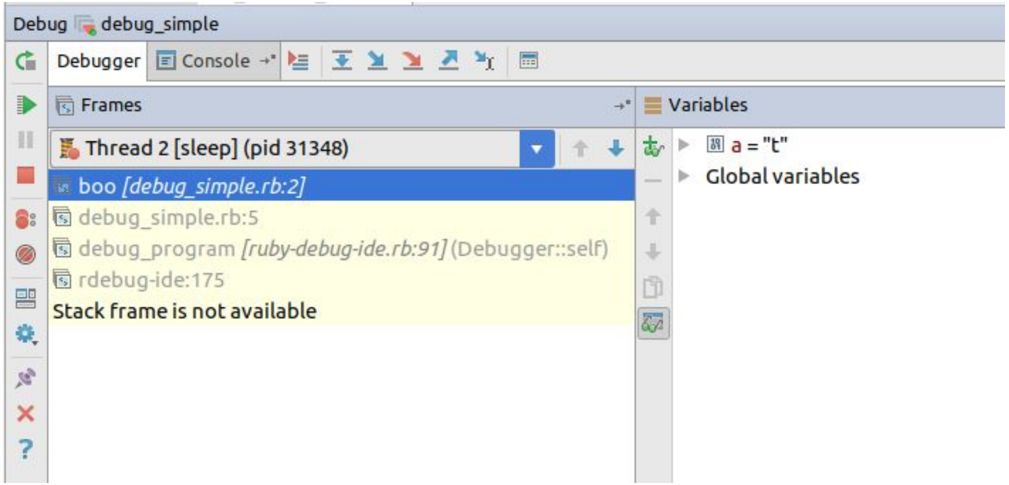
Courtesy of the Naval Surface Warfare Center, Dahlgren, VA., 1988. (https://commons.wikimedia.org/wiki/File:H96566k.jpg), „H96566k“, marked as public domain, more details on Wikimedia Commons:

https://commons.wikimedia.org/wiki/Template:PD-US

IDE (RubyMine 💜)



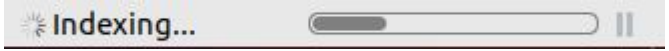
```ruby
def boo(a)
  puts "boo"
end

boo('t')

```



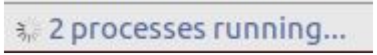
```
Thread 2 [sleep] (pid 31348)
boo [debug_simple.rb:2]
debug_simple.rb:5
debug_program [ruby-debug-ide.rb:91] (Debugger::self)
rdebug-ide:175
Stack frame is not available
```

```
a = "t"
Global variables
```

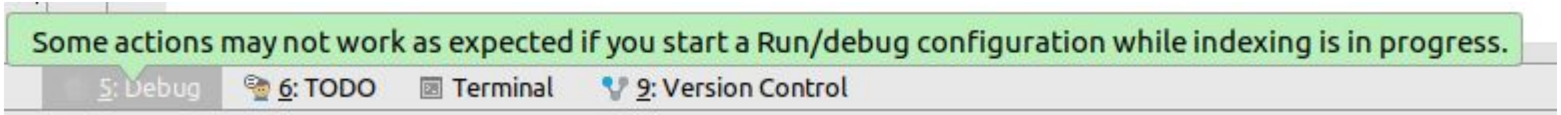
IDE (RubyMine ~~💜~~)



Indexing...

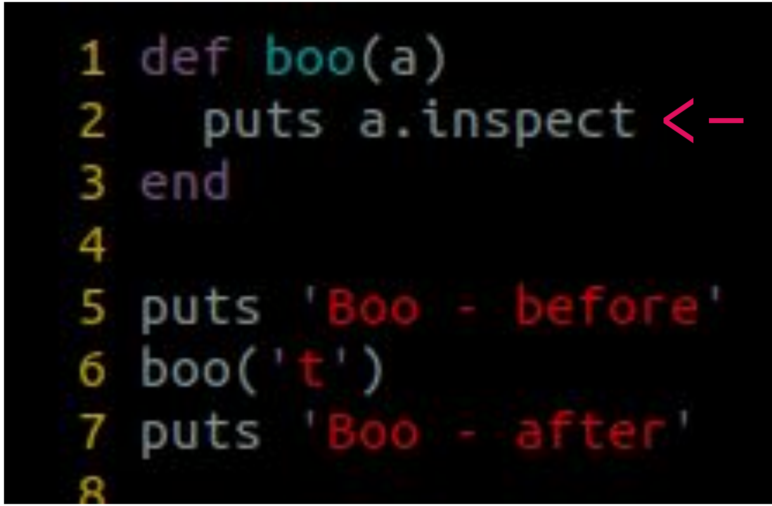


2 processes running...



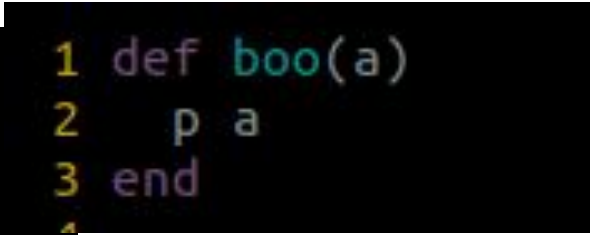
Some actions may not work as expected if you start a Run/debug configuration while indexing is in progress.

Puts



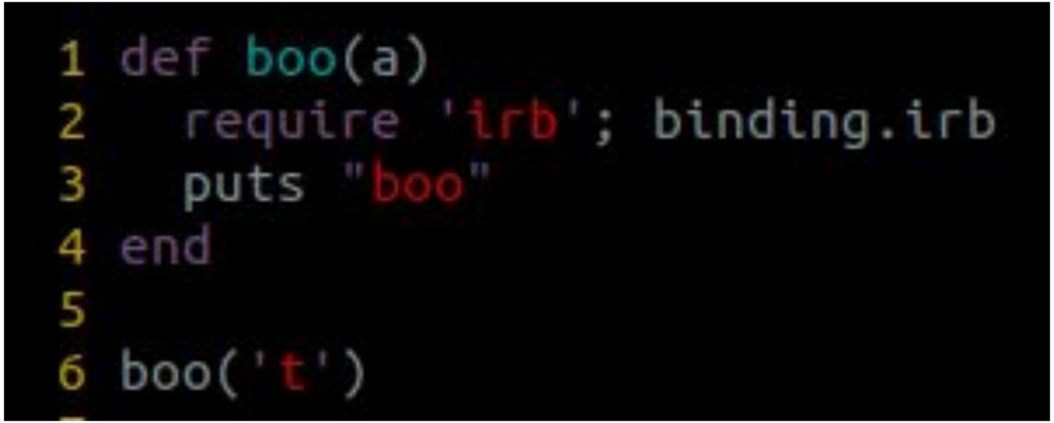
```ruby
def boo(a)
  puts a.inspect
end

puts 'Boo - before'
boo('t')
puts 'Boo - after'
```

<–



```ruby
def boo(a)
  p a
end
```

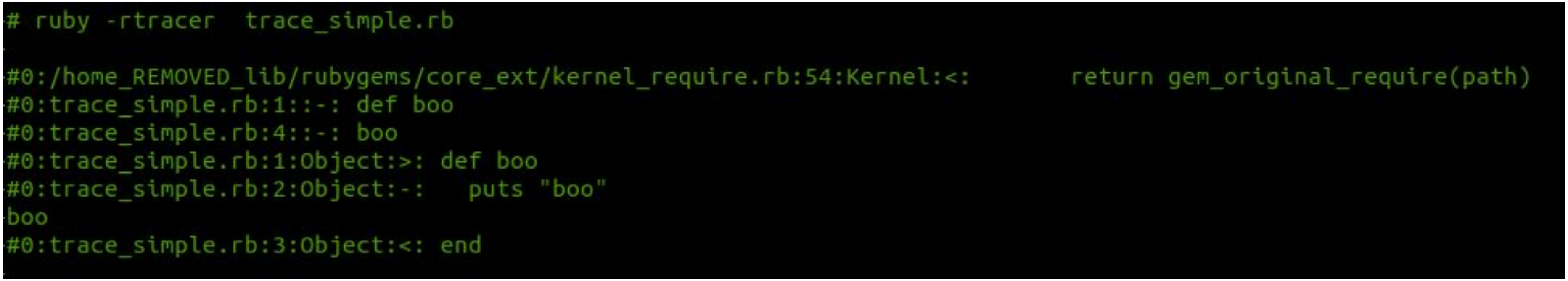
IRB (+PRY)



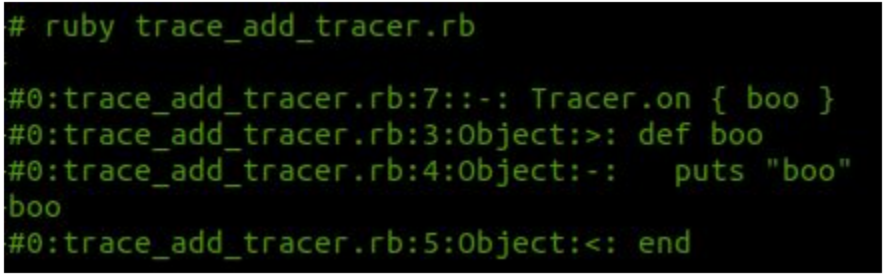
```ruby
def boo(a)
  require 'irb'; binding.irb
  puts "boo"
end

boo('t')
```

ruby 2.4.0, 2016-12-25

Tracer

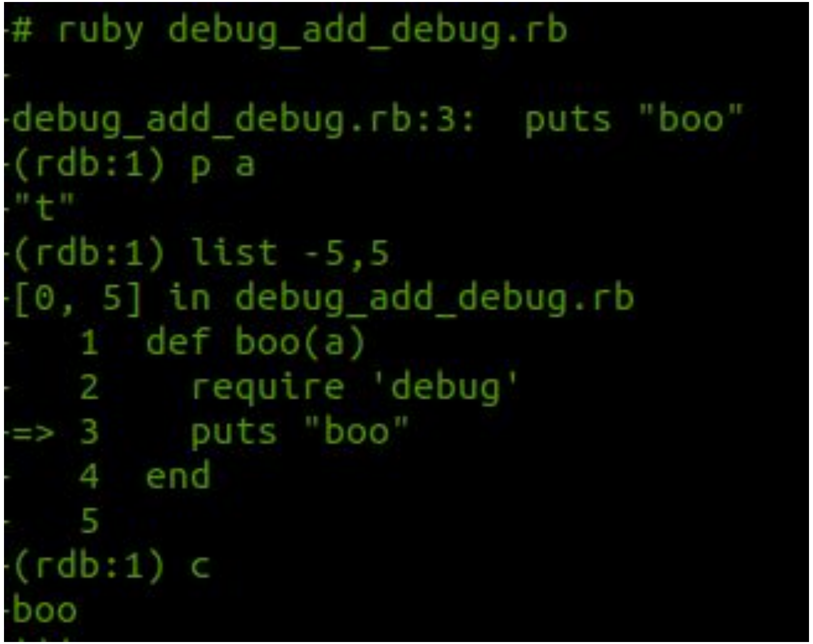


```
# ruby -rtracer  trace_simple.rb

#0:/home_REMOVED_lib/rubygems/core_ext/kernel_require.rb:54:Kernel:<:      return gem_original_require(path)
#0:trace_simple.rb:1::-: def boo
#0:trace_simple.rb:4::-: boo
#0:trace_simple.rb:1:Object:>: def boo
#0:trace_simple.rb:2:Object:-:   puts "boo"
boo
#0:trace_simple.rb:3:Object:<: end
```



```
# ruby trace_add_tracer.rb

#0:trace_add_tracer.rb:7::-: Tracer.on { boo }
#0:trace_add_tracer.rb:3:Object:>: def boo
#0:trace_add_tracer.rb:4:Object:-:   puts "boo"
boo
#0:trace_add_tracer.rb:5:Object:<: end
```

DEBUG[GER] (+BYEBUG)



```
# ruby debug_add_debug.rb

debug_add_debug.rb:3:  puts "boo"
(rdb:1) p a
"t"
(rdb:1) list -5,5
[0, 5] in debug_add_debug.rb
   1  def boo(a)
   2    require 'debug'
=> 3    puts "boo"
   4  end
   5
(rdb:1) c
boo
```

☒ IDE

...?

❌ PUTS,P
❌ IRB (PRY AND BYEBUG)
❌ DEBUG(GER)

..."JUST FORCE DEBUG[GER] TO CONTINUE INSTEAD OF STOPPING AND STOP SOMEWHERE INSTEAD OF CONTINUING"...

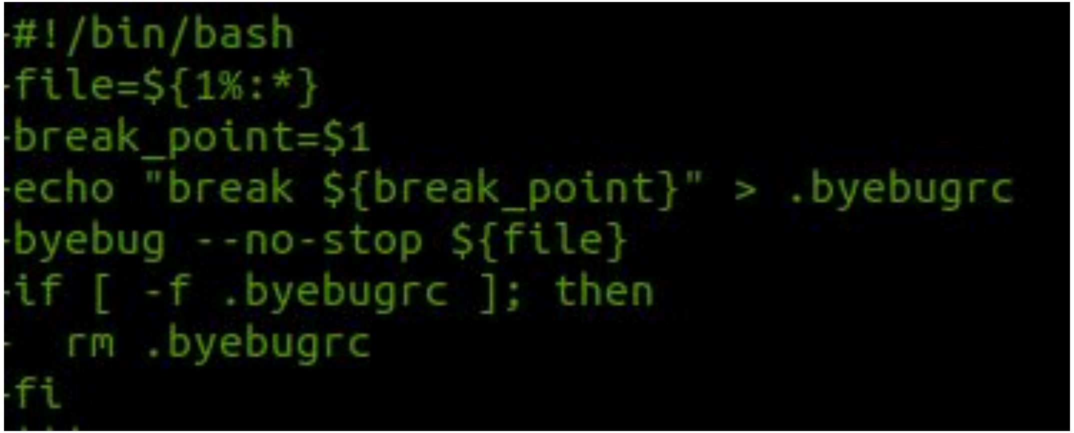
⛔

Byebug

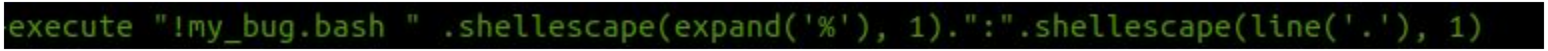
```
--no-stop
```

```
.byebugrc [~/, $PWD/]
```

MY_BUG.BASH



```bash
#!/bin/bash
file=${1%:*}
break_point=$1
echo "break ${break_point}" > .byebugrc
byebug --no-stop ${file}
if [ -f .byebugrc ]; then
  rm .byebugrc
fi
```



```vim
execute "!my_bug.bash " .shellescape(expand('%'), 1).":".shellescape(line('.'), 1)
```

=)K

@otaznik_net

https://github.com/otaznik-net/debug_rb