Deprecating and removing the broken Process.clock_getres

03/28/2020 11:46 AM - Eregon (Benoit Daloze)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</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></td>
</tr>
<tr>
<td>ruby -v:</td>
<td></td>
</tr>
<tr>
<td>Backport:</td>
<td>2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN</td>
</tr>
</tbody>
</table>

Description

clock_getres(2) is incorrect (doesn't match the actual resolution for many clocks), buggy (e.g. can return negative values) on almost all platforms and for almost all clock_id.

This is an observation based running specs for it:
https://github.com/ruby/spec/blob/ec844797a51a017ebc93af833e421362b4b24a17/core/process/clock_getres_spec.rb
https://github.com/ruby/spec/blob/ec844797a51a017ebc93af833e421362b4b24a17/core/process/fixtures/clocks.rb#L19-L59

See how many exceptions there are.
And that's not all, even CLOCK_MONOTONIC on Linux is buggy: https://github.com/ruby/actions/runs/395166997#step:16:155

My conclusion is this API is unusable for any purpose.
If people want to know the resolution for clock_gettime(), they need to call it repetitively and see how precise it is.
I will remove the "matches the clock in practice" specs for clock_getres() as I'm tired to maintain them.
But I think we should remove this impossible-to-use correctly API, because it's fundamentally broken on all platforms.

So I propose to deprecate and remove this API, because I believe it harms more than it helps for Rubyists.

Kernel hackers which want to use that function can always call it via Fiddle/FFI, but it shouldn't be a proper Ruby API since it's so broken at the OS level.

Related issues:
Related to Ruby master - Feature #8809: Process.clock_getres

History

#1 - 03/28/2020 11:46 AM - Eregon (Benoit Daloze)
- Related to Feature #8809: Process.clock_getres added