Ruby master - Feature #9842

system configuration variables (sysconf(), confstr(), pathconf() and fpathconf())

05/15/2014 11:45 AM - akr (Akira Tanaka)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		

Description

How about providing methods to obtain system configuration variables?

POSIX defines sysconf(), confstr(), pathconf() and fpathconf().
I implemented following methods in ext/etc.

- Etc.sysconf(name)
- Etc.confstr(name)
- IO.pathconf(name)
- IO#pathconf(name)

POSIX defines some names.
Various operating sysmtems define additional names.

They can be used as follows:

```
Etc.sysconf(Etc::SC_ARG_MAX) #=> 2097152
Etc.sysconf(Etc::SC_NPROCESSORS_ONLN) #=> 4
Etc.confstr(Etc::CS_PATH) #=> "/bin:/usr/bin"
Etc.confstr(Etc::CS_GNU_LIBC_VERSION) #=> "glibc 2.18"
IO.pathconf("/", Etc::PC_NAME_MAX) #=> 255
open("/") {|f| p f.pathconf(Etc::PC_TIMESTAMP_RESOLUTION) } #=> 1
```

I implemented them in ext/etc because I interpreted "etc" as
system configuration.

Any idea?

Associated revisions

Revision ea1a4f29 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45984 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45984 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

Revision 45984 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

Revision 45984 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

Revision 45984 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

Revision 45984 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

Revision 45984 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

Revision 82dbd097 - 05/19/2014 07:54 AM - nobu (Nobuyoshi Nakada)

etc.c: Etc.uname on Windows

- ext/etc/etc.c (etc_uname): add support for Windows using GetVersionExW(), GetSystemInfo(), and GetComputerNameExW() with ComputerNameDnsHostname. [Feature #9842]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46014 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46014 - 05/19/2014 07:54 AM - nobu (Nobuyoshi Nakada)

etc.c: Etc.uname on Windows

- ext/etc/etc.c (etc_uname): add support for Windows using GetVersionExW(), GetSystemInfo(), and GetComputerNameExW() with ComputerNameDnsHostname. [Feature #9842]

Revision 46014 - 05/19/2014 07:54 AM - nobu (Nobuyoshi Nakada)

etc.c: Etc.uname on Windows

- ext/etc/etc.c (etc_uname): add support for Windows using GetVersionExW(), GetSystemInfo(), and GetComputerNameExW() with ComputerNameDnsHostname. [Feature #9842]

Revision 46014 - 05/19/2014 07:54 AM - nobu (Nobuyoshi Nakada)

etc.c: Etc.uname on Windows

- ext/etc/etc.c (etc_uname): add support for Windows using GetVersionExW(), GetSystemInfo(), and GetComputerNameExW() with ComputerNameDnsHostname. [Feature #9842]

Revision 46014 - 05/19/2014 07:54 AM - nobu (Nobuyoshi Nakada)

etc.c: Etc.uname on Windows

- ext/etc/etc.c (etc_uname): add support for Windows using GetVersionExW(), GetSystemInfo(), and GetComputerNameExW() with ComputerNameDnsHostname. [Feature #9842]

Revision 46014 - 05/19/2014 07:54 AM - nobu (Nobuyoshi Nakada)

etc.c: Etc.uname on Windows

- ext/etc/etc.c (etc_uname): add support for Windows using GetVersionExW(), GetSystemInfo(), and GetComputerNameExW() with ComputerNameDnsHostname. [Feature #9842]

Revision 46014 - 05/19/2014 07:54 AM - nobu (Nobuyoshi Nakada)

etc.c: Etc.uname on Windows

- ext/etc/etc.c (etc_uname): add support for Windows using GetVersionExW(), GetSystemInfo(), and GetComputerNameExW() with ComputerNameDnsHostname. [Feature #9842]

History

#1 - 05/15/2014 01:38 PM - kosaki (Motohiro KOSAKI)

On Thu, May 15, 2014 at 7:45 AM, akr@fsij.org wrote:

> Issue #9842 has been reported by Akira Tanaka.
>
> ---
>
> Feature #9842: system configuration variables (sysconf(), confstr(), pathconf() and fpathconf())
> https://bugs.ruby-lang.org/issues/9842
>
> - Author: Akira Tanaka
> - Status: Open
> - Priority: Normal
> - Assignee:
> - Category:
>
> # * Target version:
>
> How about providing methods to obtain system configuration variables?
>
> POSIX defines sysconf(), confstr(), pathconf() and fpathconf().
> I implemented following methods in ext/etc.
>
> - Etc.sysconf(name)
> - Etc.confstr(name)
> - IO.pathconf(name)

- IO#pathconf(name)

POSIX defines some names.
Various operating sysmtems define additional names.

They can be used as follows:

```
Etc.sysconf(Etc::SC_ARG_MAX) #=> 2097152
Etc.sysconf(Etc::SC_NPROCESSORS_ONLN) #=> 4
Etc.confstr(Etc::CS_PATH) #=> "/bin:/usr/bin"
Etc.confstr(Etc::CS_GNU_LIBC_VERSION) #=> "glibc 2.18"
IO.pathconf("/", Etc::PC_NAME_MAX) #=> 255
```

Please drop this. This is broken by design. If an attacker create a
symlink which point
to FAT file system on the same place, IO.patchconf(PC_NAME_MAX) may return very
small size and might lead to security issue.

http://womble.decadent.org.uk/readdir_r-advisory.html

Or, at least, we need looooong document why you need much care about IO.pathconf.

open("/") {|f| p f.pathconf(Etc::PC_TIMESTAMP_RESOLUTION) } #=> 1

```
I implemented them in ext/etc because I interpreted "etc" as
system configuration.

Any idea?


---Files--------------------------------
sysconf-confstr-pathconf.patch (13 KB)


--
https://bugs.ruby-lang.org/
```

#2 - 05/15/2014 02:06 PM - akr (Akira Tanaka)

Motohiro KOSAKI wrote:

> Etc.sysconf(Etc::SC_ARG_MAX) #=> 2097152
> Etc.sysconf(Etc::SC_NPROCESSORS_ONLN) #=> 4
> Etc.confstr(Etc::CS_PATH) #=> "/bin:/usr/bin"
> Etc.confstr(Etc::CS_GNU_LIBC_VERSION) #=> "glibc 2.18"
> IO.pathconf("/", Etc::PC_NAME_MAX) #=> 255
>
> Please drop this. This is broken by design. If an attacker create a
> symlink which point
> to FAT file system on the same place, IO.patchconf(PC_NAME_MAX) may return very
> small size and might lead to security issue.

I see. I agree to drop IO.pathconf.

Is there a problem with sysconf(), confstr() and fpathconf()?

#3 - 05/16/2014 01:23 PM - akr (Akira Tanaka)

- File sysconf-confstr-fpathconf.patch added

Updated patch attached.
It doesn't define IO.pathconf.

#4 - 05/16/2014 03:55 PM - usa (Usaku NAKAMURA)

I don't disagree with this feature, but can't you design a little more abstract interface?

#5 - 05/16/2014 04:15 PM - akr (Akira Tanaka)

Usaku NAKAMURA wrote:

> I don't disagree with this feature, but can't you design a little more abstract interface?

What the purpose of the abstraction?

For example, we can define Etc.getconf as a unified version of Etc.sysconf and Etc.confstr.
It is exepected to work as POSIX getconf command which can invoke sysconf and confstr.
getconf is an abstraction of sysconf and confstr because it includes the both functions.
Does this abstration help you?

#6 - 05/16/2014 04:35 PM - usa (Usaku NAKAMURA)

POSIX is the standard which should be respected obviously, but Ruby is operating not only on POSIX environment.

Real needs is not revealing the items which getconf (or etc.) provides but the information along to a user's purpose, isn't it?
Whether do you want the information along to some use cases or getconf itself?
If the latter case, you can simply call getconf via system, `` or etc.
If the former case, I believe that you can determine a natural interface.

#7 - 05/16/2014 04:56 PM - akr (Akira Tanaka)

I intend this feature (Etc.sysconf, etc.) as low level API.

One of the feature I want is Etc.confstr(Etc::CS_GNU_LIBC_VERSION).
I feel it is not a good idea to define a high level interface for it.

I like functions over command invocation.
Command invocation depends various fragile factors (PATH, etc.) and difficult to treat errors.

Of course, I agree some feature may be appropriate to be defined as high level interface.
People will find such feature more easily because
this feature makes low level features more visible.

#8 - 05/18/2014 12:33 AM - akr (Akira Tanaka)

matz accepted this issue at https://bugs.ruby-lang.org/projects/ruby/wiki/DevelopersMeeting20140517Japan

#9 - 05/18/2014 01:48 AM - akr (Akira Tanaka)

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset r45984.

- ext/etc/etc.c: Etc.sysconf, Etc.confstr and IO#pathconf implemented.
- ext/etc/extconf.rb: Check sysconf(), confstr() and fpathconf().
- ext/etc/mkconstants.rb: New file.

[ruby-core:62600] [Feature #9842]

#10 - 05/19/2014 02:18 AM - usa (Usaku NAKAMURA)

Akira Tanaka wrote:

> I intend this feature (Etc.sysconf, etc.) as low level API.
>
> One of the feature I want is Etc.confstr(Etc::CS_GNU_LIBC_VERSION).
> I feel it is not a good idea to define a high level interface for it.

I see.

> I like functions over command invocation.
> Command invocation depends various fragile factors (PATH, etc.) and difficult to treat errors.

Almost I can agree it.

> Of course, I agree some feature may be appropriate to be defined as high level interface.
> People will find such feature more easily because

this feature makes low level features more visible.

I was convinced by this explanation. Thank you.

Files

sysconf-confstr-pathconf.patch	13 KB	05/15/2014	akr (Akira Tanaka)
sysconf-confstr-fpathconf.patch	12.1 KB	05/16/2014	akr (Akira Tanaka)