ANDPAD

Ruby で学ぶ Windows システムプログラミング

Hiroshi SHIBATA @hsbt
株式会社アンドパッド フェロー
2026/11/21 仙台Ruby会議03

© 2026 ANDPAD All Rights Reserved.

自己紹介



Hiroshi SHIBATA

株式会社アンドパッド

フェロー

Profile | **経 歴**

X / GitHub: @hsbt

Ruby コミッタ

RubyGems と Bundler のリードメンテナ

ruby-lang.org の root 管理者

ANDPAD Fellow

© 2026 ANDPAD All Rights Reserved.

ANDPADとは

現場の効率化から経営改善まで一元管理できる

クラウド型建設プロジェクト管理サービス



社　内

営業 / 監督 / 設計

事務 / 管理職



ANDPAD



社　外

職人 / 業者

メーカー / 流通

案件管理　資料　工程表　写真　報告　チャット　黒板　図面　受発注　・・・

© 2026 ANDPAD All Rights Reserved.

みなさん Windows 使ってますか?

1 Windowsを毎日使っている

2 WindowsでRubyを使っている

3 WindowsでRubyそのものを開発している(not WSL2)

© 2026 ANDPAD All Rights Reserved.

AIに頼めるなら、

WindowsでもRubyを開発しよう

AIに任せる	環境の準備も調査も、指示して任せられるようになった
自分も試す	普段のmacOSに加え、WindowsでもRubyを開発したい
ところが	**最初にぶつかったのが、パスの長さだった**

© 2026 ANDPAD All Rights Reserved.

ファイルパス長

© 2026 ANDPAD All Rights Reserved.

260文字を超えると、Rubyはファイルを開けない

ユーザー領域の例

```
C:\Users\<名前>\AppData\Local\
C:\Users\<名前>\Documents\...\...
```

この下にgitリポジトリ、worktree、
テスト用の一時ファイルが加わる

上限は260文字

D:\ の3文字、本体256文字、
終端のNUL

AIの作業用コピーで、
一時ディレクトリが
269文字に届いた

記録：2026年7月のRubyGemsのテスト。本体の場所（25文字）では通り、AIのworktree（72文字）では後片付けに失敗した

© 2026 ANDPAD All Rights Reserved.

良い解決策が無いまま4年、他の実装を全部調べた

“That will be a huge endeavor, but worth it.”

Bug #18923の結び（2022年7月）。#18923と#18947は4年開いたままだった

AIで突き合わせた処理系

Go	CPython	Node	Deno	Bun	PHP	Lua
	7月			10月に広げた		

他にも Ruby に取り込めそうな改善策はあったが長パス問題はその一つ

出典：bugs.ruby-lang.org/issues/18923、18947（どちらも2026-08-19にClosed）

© 2026 ANDPAD All Rights Reserved.

Goを読んだら、本命は起動時のPEBだった

最初の AI の見立ては、fixLongPath(path) で 14〜15か所に \\?\ をおく変更

Goの現行実装（src/runtime/os_windows.go の initLongPathSupport）

```
const (
    IsLongPathAwareProcess = 0x80
    PebBitFieldOffset      = 3
)
// ... RtlGetVersion で 10.0.15063 未満なら何もしない ...

bitField := (*byte)(unsafe.Pointer(stdcall(_RtlGetCurrentPeb) +
    PebBitFieldOffset))
*bitField |= IsLongPathAwareProcess
```

よくよく調べさせると別のことをやっていた

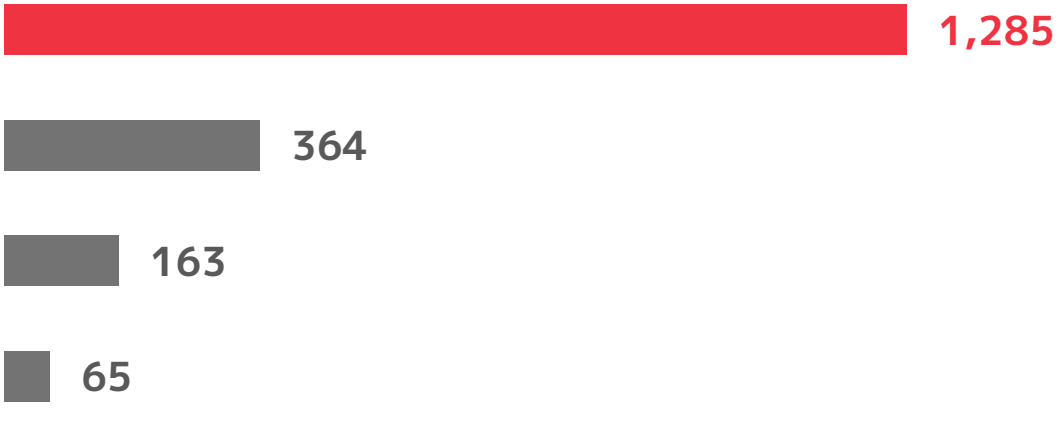
出典：golang/go の src/runtime/os_windows.go（initLongPathSupport）を抜粋。\\?\ はパスの前に付けて長さの制限を外す書き方

© 2026 ANDPAD All Rights Reserved.

開発を始める前に道具の整備

AIがRubyの文脈を読めるように、MCPを4つ作った

7月からの呼び出し（claude の248セッション、約1,900回）



長パスでも、2022年のチケットを読み、matzに問い、承認を読んだのはbugsのMCP経由

Windows機の分は入っていない。gem-codesearchはakrさんのリポジトリにMCPを足した（PR #15）。bladeは65万通のMLとmatzの発言を引く

© 2026 ANDPAD All Rights Reserved.

nobucop：nobu のレビューや修正を、AIが模倣できるようにした

nobu の修正183件とPRのレビュー384件から、46のパターンを抜き出した

変更をpushする前に、このカタログをAIが当てて直す

いちばん多いパターン（§C-2、nobu の529d86a2c3）

```
- "unknown keywords: %s",
-   RSTRING_PTR(rb_ary_join(...))
+ "unknown keywords: %"PRIsVALUE,
+   rb_ary_join(...)
```

310回　9/16〜11/16に走った。うち200回はWindows機

カタログはnobuさん本人の見解ではなく、履歴からの推定（11-08時点）。書き換えはCodexに点検させてから戻すものを決める。回数は3台の記録の合計

© 2026 ANDPAD All Rights Reserved.

MS Learn MCP で読んだ公式の定義には、「Reserved2」としか書いていない

winternl.h の PEB（64ビット版の定義）

```
typedef struct _PEB {
    BYTE Reserved1[2];
    BYTE BeingDebugged;
    BYTE Reserved2[21];          // この先頭の1バイトを使う
    PPEB_LDR_DATA LoaderData;
    PRTL_USER_PROCESS_PARAMETERS ProcessParameters;
    ...
} PEB;
```

使うのは4バイト目。説明は「OS内部用」だけで、ビットの名前もソースも無い

出典：Microsoft Learn「PEB structure (winternl.h)」の64ビット版の定義を抜粋。コメントは発表用に足したもの

© 2026 ANDPAD All Rights Reserved.

決めているのは3つの層、効くのはPEBの1ビット

manifest
アプリが longPathAware を宣言

レジストリ
システムの LongPathsEnabled（HKLM）

PEB の 1 ビット
プロセスごと。実際に効くのはここだけ

ローダが決める

manifestを読み、
レジストリが1なら
ビットを立てる
素の機では立たない

レジストリを変えるには
管理者権限が要る
立つと、32のAPIの制限が外れる

© 2026 ANDPAD All Rights Reserved.

4年分の大仕事は、4行で終わった

win32/win32.c の init_long_path_support（抜粋）

```
enum {peb_bit_field_offset = 3, is_long_path_aware_process = 0x80};
/* ... RtlGetVersion で 10.0.15063 未満なら何もしない ... */

bit_field = (BYTE *)pRtlGetCurrentPeb() + peb_bit_field_offset;
*bit_field |= is_long_path_aware_process;
```

残る制限

子プロセスのcwdは今も260まで
名前1つは255字まで（NTFS）
未文書のビット（golang/go#66560）
組み込みホスト（ruby_setupで直した）

matzの承認理由（8/6）

効かなくなっても、今日の260文字に戻るだけで、壊れはしない

最初のコミットはbb847847ae。PR #18088（2026-08-19マージ）でwin32.cに39行、全体で72行、5ファイル

© 2026 ANDPAD All Rights Reserved.

ファイルサイズ

© 2026 ANDPAD All Rights Reserved.

2GBの壁は、3つの別物だった

2GB超のファイルのSHA-256が取れなかった

大きさを聞くと落ちる。丸ごと読むと、53.7秒後に落ちる

1 fd経由のfstat	CRTの構造体が32ビットで、2^31バイトでEOVERFLOW
2 ReadFileの長さ	DWORDに切り詰められ、2^32バイトのreadが0を返す
3 longの長さ	longが32ビットなので、StringとArrayが2GiB−1で止まる（仕様）

File.size(path)はパス経由の別の経路なので動く。fstatの形は少なくとも2017年（91de3d6539）から。53.7秒は2GBちょうどのFile.binreadの記録

© 2026 ANDPAD All Rights Reserved.

2つは、Goと同じく小さく直した

Rubyで直したこと

```
/* fstat を _fstat64 に（b72368a31b3） */
- struct stat tmp;
- int ret = fstat(fd, &tmp);
+ struct _stat64 tmp;
+ int ret = _fstat64(fd, &tmp);

/* 読み書きの長さを丸める（5f860723241） */
+ if (size > INT_MAX) size = INT_MAX;
```

Goも同じ理由で切っている

```
const maxRW = 1 << 30
```

1回の読み書きは1GBまで

直している途中で別のバグが5件見つかり、書き込み側の切り詰めは#19164で直した

出典：ruby/ruby #19075、#19164、golang/go の src/internal/poll/fd_windows.go（golang.org/issue/26923）

© 2026 ANDPAD All Rights Reserved.

3つ目は、mswinでだけ長さの型を替えた

置き換える前の、longを含む行

3,654行

ruby/rubyの*.cと*.h（extとtestを含む）

1行の中にも混ざる（Array#sum）

```
// rb_ary_sum の long i, n;
rb_len_t i;   // 配列の位置
long n;       // 整数の合計
```

rb_len_t：mswinだけ64ビットの長さの型

人の手では1つずつ見分けるのがとても大変。AIにファイルごとの43コミットで置き換えさせた

ABIが変わるので、long *を渡す拡張の警告（C4133）をエラーにした

Feature #22400。3,654は置き換え前（03e33766434の親）のgit grep -w -c。置き換えは「Use rb_len_t」の43コミット。4.1.0-preview1は1段目のみ

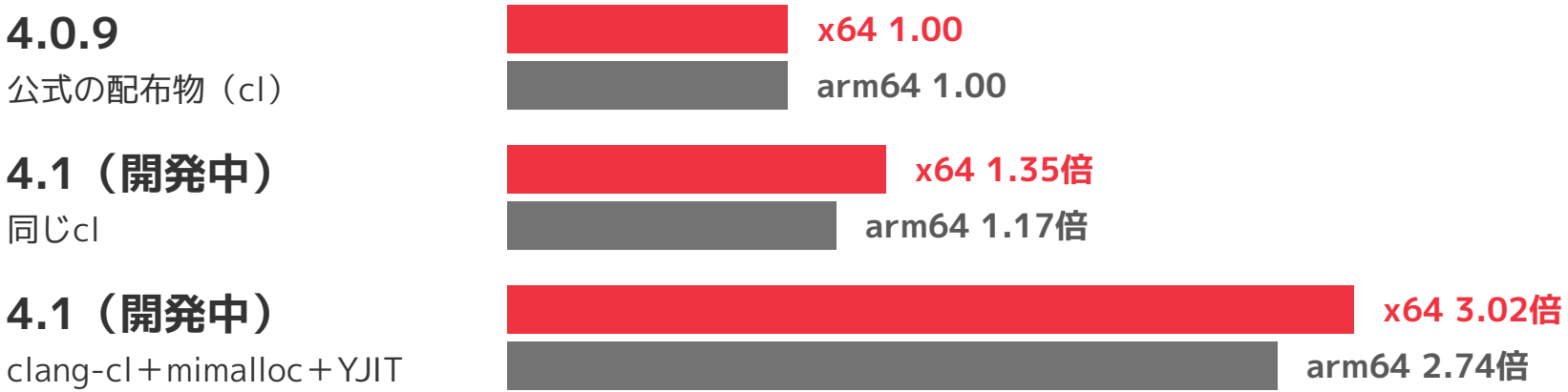
© 2026 ANDPAD All Rights Reserved.

速さ

© 2026 ANDPAD All Rights Reserved.

開発中の4.1は、4.0.9の2.7〜3倍速い

主要なベンチマーク16本の速さ（4.0.9を1とする）



x64はRyzen AI 9 HX 370、arm64はSnapdragon X2 Plus（どちらもmswin）。比べてよいのは各機械での4.0.9との比だけで、2台の速さそのものは比べない。
4.1はリリース前のmaster（d55b67dfa3）。ruby-benchの主要16本を各機械7回回した中央値の幾何平均

© 2026 ANDPAD All Rights Reserved.

Windowsで遅いところを、1つずつ直した

Dir.glob	**72.1ms → 14.9ms**	#19204
File.stat	27.5µs → 7.1µs	#18193
VMの命令の振り分け	実行時間 0.78〜0.80倍	#19229
Socket.gethostname	47µs → 2µs未満	#19701

Dir.globの直しは1行（#19204）

```
+#define IFTODT(m) (((m) & S_IFMT) >> 12)
```

数字は各PRの本文の実測。#19204はtest/の**/*（2887件）で同じビルドの有無を交互に回した最小値、#18193はx64-mswin64で5回の最良（Windows 11 24H2以降で効く）、#19229はx64のbenchmark/の19本とoptcarrotの幾何平均（2回）、#19701は1台での測定

© 2026 ANDPAD All Rights Reserved.

自分のために直すと、世界は勝手に良くなる

OSSの初心に戻って

「自分がWindowsでRubyを使いたい」をモチベーションにする

AI時代は、それを1人で続けられる

巡り巡って自分以外の誰かにとっても便利になればいいかなくらいでやっていく

© 2026 ANDPAD All Rights Reserved.