

Windows Subsystem for Linux (WSL) によりプログラムの互換性と持続性を高める手法

松浦 智之[†]大野 浩之[‡]當仲 寛哲[†]ユニバーサル・シェル・プログラミング研究所[†]金沢大学 総合メディア基盤センター[‡]

1. はじめに

2016年8月2日、Microsoft社は“Windows 10 Anniversary Update”と称すWindows 10向け更新プログラムを公開した^[1]。その中にはLinuxに準じた構造を持つ“Windows Subsystem for Linux” (WSL と略される) と呼ばれるコンポーネントが搭載された。

WSLは、Linuxに準じているゆえ、Linuxが基本としているPOSIXが定める仕様も多く取り入れている。したがって、WSLはPOSIXに準じた環境であるといえる。しかもWindowsにとって、これまでで最も本格的かつ実用的な公式のPOSIX環境である。

筆者らはこれまで、POSIXに準ずる環境向けに、プログラムの互換性と持続性を高めるための開発指針「POSIX中心主義」^[2]を提唱してきたが、WSLにもこの開発指針が適用できることを確認できたため報告する。

2. WSL

2.1. 現状

WSLはBash on Ubuntu on Windowsとも呼ばれている (“Bash on” と “Ubuntu on” は片方が呼称から略されることもある) ように、Windows上にCanonical社と共同開発したUbuntuサブシステムが搭載され、さらにその上にシェルとしてbashが搭載された環境である。Ubuntu向けにコンパイルされた実行ファイルをWindowsカーネルが直接解釈・実行でき、かつUbuntuの標準パッケージ管理システムであるAPTも搭載されているため、過去のWindowsに搭載されていたUNIX環境 (Subsystem for UNIX-Based Application等) より実用性が高くなっている。

2017年1月13日現在、WSLはまだbeta版という位置付けではあるが、Insider Preview登録者に配布される最新のビルド15002においては、最初のバージョンでは対応していなかった日本語入力・表示やポートリスン (SSHやHTTP等のホストとして振る舞える)、Win32実行ファイ

ル・プロセスの制御も行えるようになっている。

NTFSとUFSというファイルシステムの違いに起因してファイルパーミッションの扱いが一般的なUNIX環境とは若干異なる等の課題は残っているが、Microsoft社はAnniversary Update規模の大型更新を2017年に2回実施すると予告しており、それら課題も早い段階で解決していくものと思われる。

2.2. POSIX環境としてのWSL

本研究では、WSLをPOSIXに準ずる環境として捉えた場合の、その完成度の高さに着目した。著者らはプログラムの互換性と持続性を高めるための開発指針「POSIX中心主義」を提唱しているが、Windows上でもこの指針に従ったプログラミングができるようになりつつある。これは、OSのバージョンアップ時に問題となっていた既存ソフトウェアの互換性や持続性の低さを、Windowsにおいても解決できるようになりつつあることを意味する。

3. POSIX中心主義

POSIX中心主義は、開発現場においてソフトウェアの互換性や持続性の低さに悩まされてきた著者らがその問題を克服するために考案したプログラム開発指針である。具体的には次の3つの指針から構成される。

3.1. POSIX準拠

第一の指針はPOSIX規格への準拠である。POSIXで規定されているsh (Bourneシェル) やコマンドだけを使ってプログラムを書く。

POSIXに準拠すれば、理論上、POSIX準拠を謳うOS全てで動くプログラムが書けることになる。

また、POSIXには1988年の初出時の仕様の大部分が、2017年現在においても互換性を失わない形で維持されているという特徴があり、今後もその性質が持続すると予想される。各OSは、POSIX準拠を謳う限り、たとえバージョンアップする時でもPOSIXの仕様は維持する。従ってPOSIXに準拠したプログラムは長期持続性も兼ね備えることになる。

3.2. 交換可能性の担保

第二の指針は、POSIXで用意されたコマンドでは難しい処理を実現するためにPOSIX外のソフ

The Method to Make Programs More Compatible and Sustainable with “Windows Subsystem for Linux”, †MATSUURA Tomoyuki ‡OHNO Hiroyuki †TOUNAKA Nobuaki, †Universal Shell Programming Laboratory Ltd. ‡Information Media Center, Kanazawa University

トを使わざるを得ない場合のルールである。

例えば POSIX には Web API にアクセスするためのコマンドが用意されていない。そこで、それが可能なコマンドとして curl, wget という 2 つを用意し、どちらがあっても動くようなプログラムを書く。こうすることで、一つの依存ソフトウェアが使えなくなっても別ものに交換すれば動くという性質が得られる。

これにより POSIX という長期持続性の担保が得られないコマンドにも同様の性質を担保させられる。

3.3. W3C 勧告準拠

2017 年現在は、プログラムのユーザインターフェースにいわゆる Web が頻繁に利用されるものの、Web は HTML/CSS/JavaScript 等を基礎としており POSIX の範囲ではない。しかし Web にも互換性確保のために策定された W3C 勧告という標準規格があるため、プログラムに Web を含める場合にはその標準に極力準拠する。

4. 有効性の検証

POSIX 中心主義を取り入れた場合に WSL 上のプログラムの互換性と持続性がどれほど改善されるかをそれぞれ検証した。

4.1. 互換性

著者らは「恐怖！小鳥男」と名付けた Twitter クライアントプログラム^[3]を作成した。これは Twitter の閲覧や投稿等の日常操作全般をコマンド上から行うものである。

プログラムは POSIX 中心主義に基づき、POSIX で規定された Bourne シェルと UNIX コマンド、および交換可能性を担保した一部の POSIX にないコマンドを用いている。

このプログラムには、各 OS 個別のコードを持たせていないにも関わらず表 1 に示す全ての環境で動作確認がとれた。

表 1. 小鳥男の動作確認がとれた環境

OS 名称	補足
WSL	Linux (Ubuntu) 系, Windows 10 Insider Preview 15002
CentOS	Linux (RedHat) 系, 5.9, 6.5, 7.2
Raspbian	Linux (Debian) 系, Jessie 2017-01-11 版
FreeBSD	9.1-R, 10.2-R, 11.0-R
Mac OS X	Mavericks (10.9)
AIX	7.1.0.0

同一のコード、かつ Twitter アプリケーションという一つの実用的なコードが Windows, Mac,

UNIX 上で動いたことであり、Windows で書いたプログラムが他機種へ持ち出し可能になったという意義は大きい。

4.2. 持続性

WSL は初出からまだ 1 年にも満たず、持続性検証の報告を行うには期間が短く、不十分である。

しかしながら、POSIX 中心主義に基づいて 2014 年 10 月に作成したプログラムが、WSL 上で当時のバージョンから一切の修正を加えずに、かつ全く問題なく動作することが確認できた。

これは、東京メトロが Web API を介して公開する車両の在線状況を取得し、Web 画面に表示させるアプリケーション^[4]であり、同年に開催された同社主催のコンテストへの応募作品である。

WSL という環境の登場を全く予期できなかった頃、2 年後に登場するその WSL 上で動くプログラムが既にかいたという事実であり、POSIX 中心主義に従えば WSL でも長期持続性のあるプログラムが作成可能であることを間接的に示している。

5. まとめ

Windows は現在、デスクトップ PC で世界シェア 9 割を超える OS であり、ゆえに WSL の登場前は POSIX 中心主義が適用できる OS は 1 割未満であった。しかし WSL のリリースを機に、既に WSL に対応していた Windows 10、それに Mac や Linux を合わせた POSIX 中心主義の通用する OS が世界で一気に 3 割に達した^[5]。Windows のライフサイクルを鑑みれば WSL に対応したバージョン、そして POSIX 中心主義が適用できる環境は今後数年で半分以上になると予想される。

このようにしてプログラムの互換性や長期持続性を高めるうえで有利な状況に今なりつつあるため、その目的が達成できるように今後も本研究を推進していきたい。

謝辞

POSIX 中心主義を発案するきっかけとなったユニークな開発手法を推進する USP 研究所の皆様、そして本手法を支持し、本論文投稿にあたり御指導くださった金沢大学の共同研究者の皆様、心より感謝を申し上げます。

参考文献

- [1] Windows Subsystem for Linux, <https://blogs.msdn.microsoft.com/wsl/>, (2017 年 1 月 13 日閲覧).
- [2] 松浦智之他, ソフトウェアの高い互換性と長い持続性を目差す POSIX 中心主義プログラミング, マルチメディア、分散、協調とモバイル(DICOMO2016)シンポジウム.
- [3] Twitter クライアント怪人「恐怖！小鳥男」, <https://github.com/ShellShoccar-jpn/kotoriotoko/>, (2017 年 1 月 13 日閲覧).
- [4] 東京メトロ各駅の今の接近情報を見る, <https://github.com/ShellShoccar-jpn/metropiper/>, (2017 年 1 月 13 日閲覧).
- [5] Market share for mobile, browsers, operating systems and search engines | NetMarketShare, <http://marketshare.hitslink.com/>, (2017 年 1 月 13 日閲覧).