

IoT時代に資するユニケージ開発手法の普及啓発に関する研究 (2) POSIX 中心主義と情報科学教育

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

ソフトウェア開発は年を重ねるごとに高度・複雑化すると共に、高い性能も求められてきており、さまざまな OS や言語、ライブラリ、ミドルウェアが日々発表、あるいは更新されている。しかしそれらのほとんどは、効率や性能の高さを追求するばかりで、長年に渡る保守の容易さを追求していない。ゆえに非互換なバージョンアップやサポート打ち切りが行われ、一般ユーザやシステム管理者、アプリケーション開発者は維持管理に多大なコストを支払わされる。これは単純なコスト増大のみならず、システム障害という思わぬ事故を誘発する原因にもなる。

本稿は、維持管理作業を最小限に抑えるために有効な方針である POSIX 中心主義を述べ、将来のソフトウェア開発を担う人材教育における重要性を提言する。

2. POSIX 中心主義

2.1. POSIX の特徴

POSIX の目的は、“UNIX” と一括りに称されるものの独自の発展を遂げて互換性が低下してしまっていた各 OS に対し、UNIX 系 OS が準拠すべき最低限の仕様を定め、移植性の高いソフトウェア開発を実現することにある[1]。各 OS は POSIX の仕様を満たすことでユーザから UNIX と認められ、多くのユーザを獲得できる。このため、現在 UNIX 系と称している OS の大半が POSIX で規定されている仕様の大部分を満たしている。

POSIX はこのように、多くの OS ベンダが準拠に努めようとする存在であるため、逆に、1 つの OS ベンダの思惑で仕様が頻繁に改変されないという特徴を持つ。

2.2. POSIX 中心主義の定義

前節の特徴は、維持管理コストの低いソフトウェアを開発するうえで都合の良い性質である。そこで著者らは、ソフトウェア開発において POSIX の範囲でできることは最大限 POSIX の範囲

で実装する等の方針を制定し、ソフトウェアの維持管理コストを最低に抑えることを最大の目的とする開発方針を提案し、このソフトウェア開発方針を POSIX 中心主義と名づけた[2]。この開発方針の最大の特徴は、POSIX で規定されているという理由によりシェルスクリプトや (POSIX で規定されている) UNIX コマンドを多用していることである。

POSIX では C コンパイラコマンド (C99) も規定されているため、C 言語の利用も定義に矛盾しないが実現は難しい。なぜなら、C 言語は低水準言語であり、バイトオーダやハードウェア構造を意識するため、環境依存を起しやすいためである。一方、シェルスクリプトと UNIX コマンドを活用する場合、そのような環境の差異は UNIX コマンドが吸収しているため、意識する必要がない。また動作速度の面でも、気をつければそれほど劣性にならない。主要なデータ加工は UNIX コマンドという C 言語で書かれているプログラムに任せ、シェルスクリプトはそれらの制御をするという役割分担をコーディング上の方針にすれば、大量データの処理においても十分実用的なアプリケーションを作成できる。

3. POSIX 中心主義の実践

3.1. POSIX 中心主義プログラミングの例

本稿では、現在の主要な開発言語ではリレーショナルデータベース (RDBMS) によるデータベースを構築し、API を通じて SQL クエリを発行することが定番となっている処理を例示する。

今、会員 ID、会員名、よみがな、メールアドレスから成る会員情報テーブルと、諸般の事情でブラックリストに登録された会員 ID からなるテーブルがあり、ブラックリストに載っていない会員の会員名とメールアドレスをよみがな順に取得したいとする。

一般的な開発言語なら、RDBMS ミドルウェアを別途用意し、数行の SELECT 文を作成したうえで、RDBMS にクエリを送信して結果を受け取る。一方、これを POSIX 中心主義的に実現する場合、会員 ID、会員名、よみがな、メールアドレスの 4 つを半角空白区切りで 1 行とし、それが会員数分あるファイル “memberlist.txt” と、同様にブ

A Deployment of Unicage Software Development Method for IoT Generations (2) POSIX Centric System and Information Science Education, †MATSUURA Tomoyuki †ISHIYAMA Masami ‡OHNO Hiroyuki †TOUNAKA Nobuaki, †Universal Shell Programming Laboratory Ltd. ‡Information Media Center, Kanazawa University

ブラックリスト会員 ID が 1 人 1 行で記されたファイル “blacklist.txt” でデータを格納しておき (ただし両者とも会員 ID 順にソート済), 次のシェルスクリプトを実行すればよい。

```
cat memberlist.txt |
join -1 1 -2 1 -v 2 blacklist.txt - |
sort -k 3,3 |
awk '{print $2,$4}'
```

主要言語が RDBMS に頼っていたことも, POSIX の範囲だけで, 新たな言語やミドルウェアをインストールすることなく実現できる。そのうえ C 言語で実装された UNIX コマンドがデータ処理を担っているため, 遜色ない処理速度で動作する。

3.2. POSIX で実現不可能な作業の実現

POSIX ですべての処理が可能ではないため, POSIX 範囲外のソフトウェアの利用も必要になる。ただ, 単純に POSIX 外のソフトウェアの利用を許しては冒頭で記した既存のアプリケーションと同じ問題が生じる。そこで POSIX 中心主義においては「交換可能性」の担保を条件に POSIX 外のソフトウェアの利用を認めることとした。

「交換可能性」とは, 依存しているソフトウェア実装が使えなくなっても, 互換性のある別実装が使えるという性質と定義しており, POSIX 中心主義を推進する本質的な理由である。POSIX 仕様を満たす OS 実装やコマンド群には, さまざまな OS ベンダが開発した実装があるからである。

たとえば POSIX には Web アクセスを実現するコマンドがないので, curl, wget という複数の Web アクセスコマンドのうち 1 つがあれば動くようにすることを条件に利用を認めている。ただし, curl にある CGI 変数送信時の URL エンコーディング機能やファイルアップロード機能は利用してはならない。なぜなら, 担保先コマンドである wget にはない機能だからである。ただしそれらの機能は, awk 等 POSIX 範囲内のコマンドで自作でき, 問題とはならない。

3.3. Web アプリケーションの実現

前節の方針を認めれば, Web アプリケーションの作成も可能になる。まず, 特殊な機能に依存しなければ, 交換可能な Web サーバ実装は複数ある。またクライアント側では今や JavaScript の利用が不可欠だが, W3C が勧告する範囲の機能のみの利用を原則とし, その他ライブラリを利用する場合も交換可能性を担保すれば, POSIX 中心主義の目的は達成できるものと考えている。

この方針に基づき, 実際に Web アプリケーションも制作した (<http://metropiper.com/>)。これは東京地下鉄が 2014 年に開催したオープンソフト活用コンテスト [3] 応募作品である。

このコンテストは東京五輪に向けて優れたアプリケーションを募るために開催されたと言われているが, 本作品なら 2020 年までも極めて少ない維持管理コストで動き続けると予想している。一方, 長期互換性を重視していないプラットフォーム, 特に iOS や Android に依存した多くの応募作品は 2020 年でも動くのか疑問である。

3.4. 情報科学教育への提案

先に述べたように, POSIX 中心主義では各種処理を最大限 POSIX の範囲内で実現する。従って例示した URL エンコーディングも, RFC 等の技術資料を確認しつつ基本的には自力で実装する。

この作業は, 現在のソフトウェアや情報通信の基盤の学習, さらに各種仕様合理的に設計されている様子を理解するのに適した教材だと考えている。情報専門学科カリキュラム標準 (J07) との関連については, 第一報 [4] に記した。

4. まとめ

ソフトウェア開発・運用現場では, 冒頭に述べたような悩みが頻発しているにもかかわらず, ほとんどこの問題への解決策が議論されていない。特に, OS やミドルウェアといった開発基盤ではなく, アプリケーションを開発する立場の者にとってみれば, 存在する開発環境を選択するしかなく, 解決を図る機会そのものがなかなか得られない。このような不幸な現状を打開すべく, POSIX 中心主義の考え方を提案した。

POSIX 中心主義は, 誕生して日が浅く, どれほどの効果もたらされるのかといった定量的な議論も, この方針が効果をもたらす本質的要因は何かといった議論も始まったばかりである。本方針を学術的に考察することで, ソフトウェア維持管理コストを下げる必要性の認識や議論を, ソフトウェア業界全体で活発化させたい。

謝辞

POSIX 中心主義を発案するきっかけとなったユニケーj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] The Open Group. What is POSIX?. <https://collaboration.opengroup.org/external/pasc.org/plato/> (2016 年 1 月 6 日閲覧).
- [2] 松浦智之, すべての UNIX で 20 年動くプログラムはどう書くべきか, C&R 研究所, pp.3-12, 2015.
- [3] 東京メトロオープンデータ事務局. 東京メトロオープンデータ開発者サイト. <https://developer.tokyometroapp.jp/> (2016 年 1 月 6 日閲覧).
- [4] 中村和敬他, IoT 時代に資するユニケーj開発手法の普及啓発に関する研究 (1)ユニケーjシステムと情報科学教育, 情報処理学会第 78 回全国大会, 2016.