

Unix シェルスクリプトによる企業システム構築

中村 和敬†

當仲 寛哲‡

‡ユニバーサル・シェル・プログラミング研究所

1 はじめに

本稿ではシステム開発手法、ユニケーシ開発手法を紹介する。ユニケーシ開発手法は、ユニバーサル・シェル・プログラミング研究所の提唱するシステム開発手法であり、多くの企業・大学での採用実績がある [1]。

現在、企業システムは、OS 以外にも、RDBMS やミドルウェア等のソフトウェアを使用して構築される事が多い。そのため、これらのソフトウェアの金銭面でのコスト、新しいソフトウェアの使用方法を学ぶ為の学習コスト、また、実行時の計算資源のコストがかかっている。

本手法は Unix 系 OS のシェルスクリプトによってシステムを構築する手法である。本手法は Unix の設計思想 [2] に強い影響を受けている。本手法は Unix 系 OS の、機能とコマンドのみを利用するので、少ないコストによりシステムを構築することが可能である。また、学習コストの低さから、企業システムの内製化にも適している。同時に、本手法は必要な Unix 系 OS の機能とコマンドのみを利用するので、オールインワンのソフトウェアを使用した場合と比べて、少ない計算資源で動作するシステムを構築する事ができる。これにより、安価で高速で柔軟なシステムを構築する事が可能である。

本手法は、Unix 系 OS の基本コマンドのみでシステムを構築する事ができるが、シェルスクリプトの記述を単純にし、またデータを高速に処理するため、いくつかの専用のコマンドを開発している。専用コマンドのセット、usp Tukubai コマンドは、ユニバーサル・シェル・プログラミング研究所が有償にて配布している。また、usp Tukubai コマンドの一部は Open usp Tukubai として無償で一般に公開されている [3]。本稿で例示するシェルスクリプトは、Open usp Tukubai コマンドを使用している。

2 テキストによるデータ表現

本手法はデータを中心に置いたシステム開発手法である。全てのデータを行指向のテキストデータとして保存する。テキストは一般的なデータ表現である。こ

れにより、システムの仕様変更に対応する事が可能であり、またデータを処理するソフトウェアの変更に影響を受けない。

例えば、小売店での商品販売データの一件あたりのレコードの項目としては、以下の要素がある。

- 販売店番号: 商品を販売した店の識別番号
- 販売日時: YYYYMMDDHHMMSS 形式の商品を販売した日時
- 商品番号: 販売した商品の識別番号
- 販売価格: 販売した商品の販売価格
- 販売数量: 販売した商品の数量

本手法ではこれらの項目を、以下のようなスペース区切りの行指向テキストデータとして表現する。

```
001 20150101123405 7714464893748 260 7
001 20150101123406 6461081761586 252 5
001 20150101123409 9419757962824 134 1
001 20150101123421 4179815504651 481 5
001 20150101123422 4736116438136 462 4
001 20150101123424 7177025238465 158 1
001 20150101123429 2016755240936 417 3
001 20150101123437 4881252263046 202 10
001 20150101123439 9640893236115 312 9
001 20150101123454 0458920590949 412 1
(以下略)
```

スペースで区切られた各フィールドは、左のフィールドから、販売店番号、販売日時、商品番号、販売価格、販売数量である。それぞれのレコードは、販売店番号と販売日時をキーとして識別する事ができる。このようなレコードを適当な単位で、例えば販売店毎、日毎の単位でファイルに保存する。それぞれのファイルは、sort コマンドの -k オプションを用いて、キーでソートしておく。

3 シェルスクリプトによるデータ処理

前節で述べたような行指向テキストデータを、Unix のコマンドを用いて処理を行う。各々のコマンドは単純

Enterprise System Construction by Unix Shell Script
†NAKAMURA Kazutaka ‡TOUNAKA Nobuaki
‡‡Universal Shell Programming Laboratory.Ltd

な処理のみを行うものであるが、それぞれのコマンドの入力と出力をパイプによって接続する事が可能であり、それによって複雑な処理を組み立てる事ができる。

そのため、一段づつ処理を追加して、入力に対する出力を確認しながら処理を組み立てて行く事が容易であり、経験の浅いプログラマも容易に正しいプログラムを作る事ができる。また、パイプによって入力と出力を結合されたコマンドは、それぞれ別のプロセスとして動作する。そのため、近年主流のマルチコアプロセッサの能力を容易に活用する事ができる。

例えば、先ほどの小売店の販売データから、販売店番号 001 番の新橋店の日毎の売上を算出する場合の処理は、以下のように記述される。

```
# 処理 1: 販売データの読み込み
cat 販売データ |
## 1:販売店番号 2:販売日時 3:商品番号
## 4:販売価格 5:販売数量
#
# 処理 2: 販売日時を販売日に修正し
#     必要なフィールドのみに絞り込み
self 1 2.1.8 4 5 |
## 1:販売店番号 2:販売日
## 3:販売価格 4:販売数量
#
# 処理 3: 新橋店のレコードを抜き出し
awk '$1=="001"' |
# 処理 4: 販売価格と販売数量から売上を算出
awk '{ print $1, $2, $3*$4 }' |
## 1:販売店番号 2:販売日 3:売上
#
# 処理 5: 日毎の売上を集計
#     販売店番号、販売日毎に
#     売上を合計
sm2 1 2 3 3
## 1:販売店番号 2:販売日 3:日毎売上
```

self コマンドと、sm2 コマンドは Open usp Tukubai コマンド [4] である。

4 レベルによるデータの管理とプロジェクト体制

企業システムは、非常に多くの種類のデータを、様々な対向システムから取得して、業務上の必要に応じて様々な出力を行うものである。また、企業システムは事業環境の変化に応じて、柔軟に変化する事が求められる。これらの問題を解決するために、本手法ではデータの整理を非常に重視している。

データは LV1 ~ LV5 に分類して整理して管理する。LV1 データは対向システムから取得したデータである。LV2 データは LV1 データを行指向テキストデータに変換したデータである。LV3 データは LV2 データを、業務や業務上必要な指標の単位で、時系列で整理したデータである。LV4 データは業務毎の処理に合わせて、前処理を施したデータである。LV5 データは最終的な出力データである。

本手法では、LV1 データから LV3 データまでを作成するチームと、LV3 データを使用して、LV4 データと LV5 データまでを作成するチームを分けてシステムを構築する。前者のチームには、ユーザ企業の業務の知識の豊富なメンバーを割り当て、業務に合わせて適切に整理された LV3 データを用意するようにする。

LV3 データを適切に用意しておく事で、必要なデータをすぐに取り出す事ができる。また、プログラムの性能はアルゴリズムに左右されるが、アルゴリズムはデータ構造によって決定される。したがって、LV3 データのデータ構造によって、それ以降の処理で使用するアルゴリズムが強制される。これにより、業務知識の少ない、経験の浅い開発者であっても、適切なアルゴリズムで処理を記述する事が出来る。

5 まとめ

本稿ではユニケース開発手法を紹介した。本手法は全てのデータを行指向のテキストデータによって保存する。データは、単純なコマンドをパイプによって組み合わせ、段階的に処理される。データは LV1 ~ LV5 という区分けで整理され、業務知識に乏しい開発者であっても素早く、高速に動作するアプリケーションを開発する事ができる。本手法は素早く、安価に、柔軟な企業システムを開発するための手法である。

参考文献

- [1] 有限会社ユニバーサル・シェル・プログラミング研究所. <http://www.usp-lab.com/works.html>.
- [2] 芳尾 桂 (翻訳) Mike Gancarz (著). UNIX という考え方 その設計思想と哲学, 2001.
- [3] Open usp Tukubai ダウンロード. <https://uec.usp-lab.com/TUKUBAI/CGI/TUKUBAI.CGI?POMPA=DOWNLOAD>.
- [4] Open usp Tukubai コマンドマニュアル, <https://uec.usp-lab.com/TUKUBALMAN/CGI/TUKUBALMAN.CGI?POMPA=LIST>.