Engineer

Bash

多くのスクリプトが対象とするGNUのシェル

どのようなものか

Bash、Bourne-Again SHellは、GNU Bashとして配布され、`bash`で起動するプログラムです。プロンプトに入力したり、ファイルに保存したりしたコマンドを読み、実行します。他のプログラムを起動して入出力をつなぎ、変数、ループ、条件判定、関数で、一連のコマンドが判断できるようにします。名前は、Bell LabsのResearch Unix第七版で`sh`を公開したStephen Bourneにちなむ言葉遊びです。マニュアルは、そのシェルにおおむね互換で、Korn shellとC shellの機能も取り入れ、POSIXのシェル部分への適合を意図すると説明します。現行は5.3で、5.3のマニュアルは2025年5月付です。

シェルは、動くウインドウと混同しやすいものです。端末は文字を描き、キー入力を渡し、シェルは、その向こうで解釈するプログラムです。対話的には、プロンプト、履歴、タブ補完を使い、非対話的には、スクリプトを先頭から最後まで、人の操作を待たずに読み進めます。多くの落とし穴は、後者にあります。

同じ役割を担うプログラムは、他にもあります。POSIXは移植可能な`sh`言語を定めますが、`/bin/sh`の実体はOSで異なります。DebianとUbuntuでは、Almquist shell由来の小さなPOSIXシェルdashです。Debian Policyは、bash、dash、kshをよく使うものに挙げます。macOSは10.15から、新規アカウントの標準ログインシェルをZshにしましたが、`/bin/bash`も残っています。スクリプトは、先頭行のshebangで、インタープリターを選びます。`#!/bin/sh`はOSのPOSIXシェル、`#!/bin/bash`や`#!/usr/bin/env bash`は、Bashを明示的に指定します。

概念を判断に生かす

「Bashとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 サーバー、メール、アカウント、更新を日々管理し、誰もが頼る仕組みを平穏に動かし続けます。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 UbuntuでPostgreSQL・MS SQLサーバー10台を運用 — 10 導入し正常に保ったDBサーバー

UbuntuでPostgreSQL・MS SQLサーバー10台を運用 お問い合わせ →

なぜ重要なのか

Bashは、Unixサーバーの処理をつなぐ言語です。ほぼすべてのLinuxにあり、入力するコマンドはすでにBashなので、数個をファイルに保存すれば、すでにある環境で動くスクリプトになります。ダンプ、圧縮、外部へのコピーのように既存プログラムをつなぐには、パイプ、リダイレクト、終了状態を標準の語彙とするため、特に短く書けます。

スクリプトが大きくなると、同じ性質が負担に変わります。値はすべて文字列で、算術は整数だけなので、JSON、日付、小数の金額など、構造のあるものを別のツールへ渡します。クォートしない展開は単語に分割され、その後ファイル名のパターンとして展開されるため、空白やアスタリスクでコマンドの意味が変わります。エラー処理は明示的に選ぶもので、標準では失敗後も続きます。`set -e` と `pipefail` にも、マニュアルが数段落を要する例外があります。データ構造、解析器、テスト一式が必要になったら、すでに一つのプログラムなので、Pythonなどのほうが保守しやすい、というのが実用的な目安です。

移植性も選択です。Debian Policyは、`/bin/sh` のスクリプトをPOSIXと少数の拡張に限定し、それ以上が必要なら、`#!/bin/bash` などで明示するよう求めます。`checkbashisms` や、poshのような厳格なシェルで確認し、迷えば `/bin/bash` を勧めています。標準の変化は遅く、Bashは3.0から `pipefail` を提供しても、POSIXが加えたのは2024年版です。

シェルはテキストをコマンドへ変えるため、外部のテキストが届く場所ではセキュリティ境界になります。2014年9月のCVE-2014-6271、Shellshockは、その広さを示しました。Bashは4.3まで、環境変数内の関数定義の後のコマンドを実行していました。ApacheのCGI、OpenSSHのForceCommand、一部のDHCPクライアントなど、攻撃者が制御できるテキストを環境変数へ入れてからBashを起動するものが、コード実行の経路になりました。最初の修正は不完全で、残りにCVE-2014-7169が付けられました。日々の作業での教訓はもっと身近です。外部のテキストは引数として渡し、`eval` や `sh -c` に渡す文字列の一部にはしません。

実際の活用例

仕事全体で、Bashは技術構成に示す役割を保っています。データ、自動化、サービスはPython、処理をつなぐ部分は、読みやすく書くBashです。

IPsec VPNの背後のデータウェアハウスからGoogle BigQueryへ流す六つのETL・ELTパイプラインでは、取得したアーカイブにParquet、CSV、.bakがありました。BashとPythonが展開し、種類と列構成で分類し、bcp、psql、SSISがステージングのスキーマへ読み込みました。その後、gzip、Google Cloud Storage、bqでBigQueryへ送りました。展開と分類をスクリプト化し、手動で三時間以上の処理を20分未満にし、週一回から毎日同期できるようにしました。

百の重要データの自動バックアップも同じ分担です。BarmanがPostgreSQL、Google Cloudが遠隔コピーを担い、BashとPythonが全体を調整・検証し、保持ルールでコピーを最新に保ち、整合性検査で存在だけでなく正常さを確かめます。既存ツールを順につなぎ、各段階の終了ステータスを確かめることにスクリプトの価値を置く、というのがBashに合う仕事です。

問題が起きるところ

空白のある名前が、二つの引数になります。 クォートしない `$file` は空白で分かれ、globとして展開され、 `report 2024.csv` は二つの名前として `rm` に届き、余分な `*` はディレクトリ全体に一致します。 `"$file"` のようにすべての展開をクォートし、引数一覧は `"$@"` で渡し、名前の一覧は配列に入れ、残りをShellCheckで検出します。

ダンプが失敗しても、バックアップのパイプラインが成功と報告します。 パイプラインの終了ステータスは最後のコマンドのものなので、 `pg_dump app | gzip > app.sql.gz` では、失敗後にわずかな入力だけを受けたgzipが成功します。最後の非ゼロ状態を返す `set -o pipefail` を有効にし、 `gzip -t` で、できれば復元テストで結果を確認します。

`set -e` **でも、失敗の先へ進みます。** `if` や `while` の条件、 `&&` ・ `||` でつないだコマンド列の最後以外、またそれらの場所から呼ぶ関数内では無視されます。そのため、下のダンプが失敗しても、成功と表示します。

```
#!/usr/bin/env bash
set -euo pipefail

backup() {
```

```
  pg_dump app > /var/backups/app.sql
  gzip -f /var/backups/app.sql
}

if backup; then # ここではbackup内のset -eが無視される
  echo "バックアップ完了"
fi
```

似た例が `local out=$(cmd)` で、`local` の終了ステータスを返し、`cmd` の失敗を隠します。関数内の重要な各コマンドには `|| return 1` を続け、変数は宣言してから、別の行で代入します。

一台で動いても、他では壊れます。 `#!/bin/sh` なのに `[[` や配列を使うと、あるOSではBash、別ではdashで動きます。後者では `[[` は不明なコマンドなので、`if` はエラーを出してelseへ進み、配列代入は構文エラーで終了します。Macの `/bin/bash` は3.2.57で、4.0で入った連想配列や `mapfile` より古く、`declare -A` は無効なオプションとして止まります。必要なシェルをshebangに書き、対象の最古の版でテストします。

後片付けが、はるかに多くを削除します。 `build_dir` が未設定だと、`rm -rf "$build_dir"/*` は `rm -rf /*` になります。`set -u` は未設定をエラーにし、`"${build_dir:?}"` なら、未設定または空文字列の際に、非対話スクリプトを止めます。

参考資料

- GNU Bash Reference Manual：5.3の展開、クォート、`set` 、`set -e` が無視される正確な条件です。
- Shell Command Language：`#!/bin/sh` の範囲となる、POSIX.1-2024の移植可能な `sh` です。
- Debian Policy：スクリプト：`/bin/sh` の必須機能、Bashを指定する条件、`set -e` の助言です。
- Bash NEWS：構文に必要な版を確認する、保守者の版別の追加機能一覧です。
- CVE-2014-6271：攻撃経路と後続の識別子を記載する、NVDのShellshock記録です。



用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「Bashとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/bash/