D-Shell: ディペンダブルな運用を支援するシェル処理系

関口　渚 †　　倉光　君郎 †

1. はじめに

近年，コンピュータ・システム障害が与える社会的な影響が大きくなり，システム運用をディペンダブルに行うことが重要となっている．我々は，ディペンダブルなシステム運用を支援するためのシェル処理系 D-Shell を開発している．D-Shell は例外処理機能を中心に，従来のシェルではサポートしていなかった高度な言語機能をサポートしている．本デモでは，D-Shell の例外処理機能を中心に，いくつかの機能を紹介する

2. 例外処理機能

従来のシェルスクリプトでは，コマンドの終了ステータスや trap コマンド[1] を用いることで異常系の処理を記述していた．しかしながら，従来の方法は，詳細なエラー情報を取得することができず，エラーの原因に応じた処理を記述するのが困難であった．また，コマンドを呼び出した関数の中でしかエラー処理を行うことができないといった問題があった．このような問題を解決するために D-Shell では，例外処理機能を用いてエラー処理を記述する．コマンドの処理に失敗した際，例外を発行する機能を持つ．これに加えて，コマンドが発行するシステムコールからコマンドのエラー原因を推定し，それに応じた例外を発行することができる．システムコールトレーサ[2] を用いることでコマンドが発行するシステムコール情報を取得する．その情報から終了直前に発生したシステムコールエラーを取得し，それに基づいた例外発行を行う．

3. その他の機能

D-Shell では例外処理機能の他に，複数台のサーバでのコマンドの並列分散実行をサポートする．また，例外発生時にファイルをロールバックすることもできる．

† 横浜国立大学
Yokohama National University

```
try {
    ls /hoge   ##例外処理を行うコマンド
}
catch(FileNotFoundException e) {
    ##ファイルが存在しない際の処理
}
catch(NotPermittedException e) {
    ##アクセス権限がない際の処理
}
```

図 1　例外処理の記述例

4. デモンストレーション

本デモでは，D-Shell の基本的な機能の実演と，コマンドの例外処理の実演を行う．いくつかのコマンドを D-Shell 上で実行する際，意図的にエラーを発生させる．このとき発行される例外の補足と，その種類に応じた例外ハンドラの実行が行われる様子を示す．

図 1 は D-Shell における例外処理（エラーハンドリング）の記述例を示したものである．try-catch 文を用いて例外処理を記述する．try 節中でコマンドの実行が失敗し例外が発行されると，その種類に応じた catch 節に処理が移る．

5. 今後の課題

現在，例外処理の実装で利用しているトレーサ（strace+）は ptrace システムコールを利用しているため，オーバーヘッドが非常に大きい．今後は，libc のラッパーライブラリを用いたトレーサを実装することによって，システムコールトレースのオーバーヘッドの削減と，エラー原因推定の精度向上を行う予定である．また，並列実行した複数のコマンドから発行される例外の補足や非同期実行コマンドの例外処理を実現するための言語機能（async/await[3]）を導入する予定である．

参 考 文 献

1) Newham, C.: *Learning the bash shell, third edition*, O'Reilly Media, Inc., 3 edition (2005).
2) : strace+: An improved version of strace that collects stack traces, `http://code.google.com/p/strace-plus/`.
3) : 非同期プログラミングのベスト プラクティス, `http://msdn.microsoft.com/ja-jp/magazine/jj991977.aspx`.