

パイプを使ったシェルコマンドの学習を支援する 初心者向けツールの提案

三枝泰士† 竹原一駿† 亀井仁志† 喜田弘司† 最所圭三†

香川大学†

1 はじめに

香川大学では、Linux シェルコマンドの使い方を学習する授業がある。単体のシェルコマンド（単一コマンド）の使い方は、初心者にとって授業や Web サイトの閲覧で容易に理解できていた。しかし、パイプに関しては、多くの初心者が授業中に質問を行っていたことから、パイプを用いたシェルコマンド（複合コマンド）の作成は、初心者にとって理解が難しいと考えた。

シェルコマンド学習支援ゲームの開発[1]では、シェルコマンド初学者を対象に、コマンドの記憶と基本的なファイル操作を修得する目的のゲームを実装している。また、Terminus[2]は、Web ブラウザ上で動作する、テキストアドベンチャーゲームである。ls 等基本コマンドの振る舞いを現実と比喻することで理解を促す。

本稿では、授業で Linux に触れたばかりの初心者に対して、パイプを用いた複合コマンドの作成の学習を支援するツール「enbicutter」の提案及び開発について述べる。

2 複合コマンドの理解に対する課題

本ツールでは、次に示す3つの課題を解決する。

課題①: 複合コマンドのパイプライン処理は、単一コマンドの標準出力を次の単一コマンドの標準入力として渡すことで実現する。複合コマンドの作成のためには、使用する単一コマンドとオプションを選択する必要がある。しかし、初心者は複合コマンドの作成に必要な単一コマンドとオプションを把握していないことが多い。また、Web サイトの閲覧により単一コマンドを調査できるが、複合コマンドに必要なオプションの選択が困難となる要因が、2つ考えられる。1つ目は、初学者はコマンドの調査が不得意と考えられること、2つ目は、オプションが多いコマンドが存在することである。

課題②: 複合コマンドは最後の単一コマンドの出力結果しか表示しない。そのため、各単一コ

マンドの入出力を予想しながら、作成する必要がある。複合コマンドの実行結果が予想と異なる原因の単一コマンドを特定するには、複合コマンドのどこまでが予想と一致しているか検証する必要があり、原因究明に手間がかかる。

課題③: 計算モデルの中には、C 言語などがとる要求駆動型、パイプライン処理などがとるデータ駆動型がある。データ駆動型は、要求駆動型と比べて授業で学習することが少なく、多くの初心者は不慣れである。初心者がデータ駆動型を容易く自習できる環境が無いことから、授業時間外の学習は難しい。

3 課題を解決する機能

ヒントの提示(課題①): 段階的にヒントを与える。まず、学習者がヒント1のボタンを押すと、複合コマンドの作成に必要な単一コマンドとオプションを提示する。次に、ヒント2のボタンを押すと、複合コマンドの設計方針など、大きく解答を助けるヒントを示す。何回も間違えた場合、解答例を提示するボタンを表示する。

複合コマンドの途中経過の表示(課題②): 複合コマンドの途中経過となる、部分的な複合コマンドの出力結果を表示することで、複合コマンドのパイプに渡される出力の流れを把握し、想定外の結果となる単一コマンドの位置を特定する支援を行う。

不正解の場合に学習者に提示する画面を図1に示す。学習者は、不正解の原因である単一コマンドを特定するため、注目する単一コマンドを選択し、複合コマンドの途中経過となる単一コマンドの出力を確認する。青背景の画面を見ると、grep コマンドではユーザ名の表示が来ているが、黒背景の画面を見ると、cut コマンドでユーザ名の右の列を抽出していると分かる。よって、cut コマンドのオプションに原因があると分かる。以上のことから、抽出位置を指定する f オプションの値を9から8に修正すればよいことを推測できる。

図1下部に示すように、途中経過の出力を青、黒、緑で分けた3つの画面に段階別に表示する。矢印は説明のために記入しているものである。

青背景のコマンドの標準出力は、青矢印で示す画面に表示される。黒、緑の背景のコマンドの標準出力に関しても同様に、各色の矢印で示す画面に表示される。黒背景のコマンドは、注目している単一コマンドである。prev/next ボタンを押すことにより、注目する単一コマンドを変更できる。

授業時間外の学習(課題③): Web サービスで提供することで、Web ブラウザ以外の環境を用意せずとも利用できる。これにより、初心者が授業時間外でも平易に学習できる。

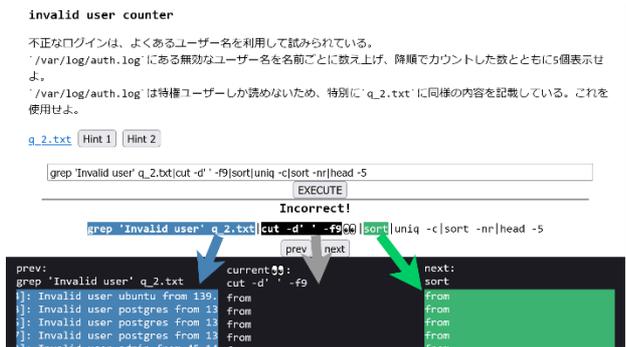


図 1 学習者に提示する画面

4 実装

4.1 システム構成

に本ツールの構成図を示す。構築を容易にするため、Docker コンテナを使用している。SSL/TLS で安全性を高めた、出題・解答を行うためのリバースプロキシ、授業時間外の学習を実現するための Web サーバ (API サーバも兼ねる)、問題と解答を記録するためのデータベースサーバの 3 つのコンテナで構築している(課題③)。学習者は Web サーバにアクセスすると、Web ブラウザに複合コマンドを用いる問題、解答入力欄、ヒントを表示するボタンが表示される(課題①)。解答として複合コマンドを入力し実行ボタンを押す。解答はデータベースに記録する。その後、複合コマンド実行用コンテナ (JCECON: Json Command Execute CONtainer) が起動される。複合コマンドを JCECON に送り、実行する。は、`grep '/bin/bash' /etc/passwd | cut -d ':' -f1` を実行したときの JCECON の出力例である。

JCECON では、複合コマンドの途中経過を出力するために、単一コマンドの標準出力を tee コマンドに渡し、それぞれをファイルに保存する。各ファイルの内容を一括して JSON 形式でブラウザに返却する。返却後、JCECON は削除される。ブラウザでは、JSON 形式をパースして複合コマンドの実行結果や途中経過を表示する。

4.2 問題とヒントの登録

出題者の問題の登録には、YAML 形式の設定

ファイルを用いる。設定ファイルには、問題のタイトル、問題文、解答例、実行結果、ヒントを記述する。問題の登録 API にアクセスし、設定ファイルの内容をデータベースに登録する。

4.3 学習者に提示する画面

図 1 に示すように、問題や解答入力欄と合わせて、複合コマンドの途中経過となる、部分的な複合コマンドの出力結果を表示する(課題②)。3 節に示したように、複合コマンドと、その出力を表示する画面を、青、黒、緑を背景色とする 3 つの画面に分ける。両目の絵文字によって、注目している単一コマンドの位置を学習者に分かりやすく示す。学習者は prev ボタンや next ボタンを押すことで、注目したい単一コマンドを変更する。変更と同時に、青、黒、緑が背景となる画面の内容も変更される。

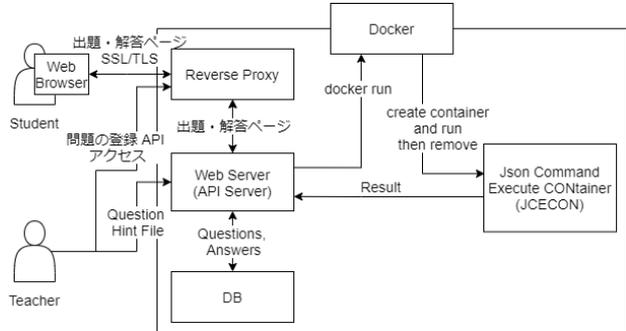


図 2 enbicutter の構成

```
{
  "stdout": [
    {
      "0": [
        "root:x:0:0:root:/root:/bin/bash",
        "ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash"
      ]
    },
    {
      "1": [
        "root",
        "ubuntu"
      ]
    }
  ]
}
```

図 3 JCECON の出力結果

5 おわりに

本稿では、初心者の複合コマンドの学習を支援するためのツール「enbicutter」を提案した。

今後の課題は、標準エラーの内容の表示や解答を間違えた原因をツールから指し示す仕組みの検討である。

参考文献

[1] 佐藤将真, 西田誠幸. “シェルコマンド学習支援ゲームの開発”, 第 77 回全国大会講演論文集. Vol.2015, No.1, pp.949–950, 2015.
 [2] “Terminus”. <https://web.mit.edu/mprat/Public/web/Terminus/Web/main.html>, (access on 2021-12-28).