

UNIX/Linux②基礎編講習会
演習問題の解答

2020年9月版

明治大学
生田メディア支援事務室

1. UNIX/Linux②基礎編

1.1. 演習問題 1 解答

(1) 現在作業しているディレクトリ内にある全てのファイルを確認してください。

```
$ ls -a
```

(2) ディレクトリの移動方法を確認しましょう。

(ア) /bin ディレクトリへ移動してからホームディレクトリへ移動する。

```
$ cd /bin
```

```
$ cd
```

(イ) pwd コマンドでホームディレクトリの場所を確認。その後..を使い、/ディレクトリまで上がってからホームディレクトリまで下りてくる。

```
$ pwd
```

```
$ cd ..
```

※/ディレクトリまで繰り返し上がる

```
$ cd ..
```

※ホームディレクトリまで繰り返し下りる。..はカレントディレクトリ直下の任意のディレクトリ名

(3) ファイルの作成、削除を試みましょう。

(ア) sample ディレクトリを作成し、その中に sample1 ファイルを作成する。

```
$ mkdir sample
```

```
$ cd sample
```

```
$ gedit sample1
```

※保存して終了する

(イ) sample1 ファイルをホームディレクトリにコピーする

```
$ cp sample1 ../
```

(ウ) ホームディレクトリに移動して、sample ディレクトリを削除する

```
$ cd
```

```
$ rm -r sample
```

(エ) sample1 ファイルを隠しファイルに変更する。(ファイル名の最初に.が付くと隠しファイル)

```
$ mv sample1 .sample1
```

(オ) 隠しファイルの.sample1を確認し、.sample1を削除する

```
$ ls -a
```

```
$ rm .sample1
```

1.2. 演習問題 2 解答

- (1) `date` コマンドを使って、本日の日付を「西暦/月/日」の形式で `test/test8` ファイルに書き込んでください（リダイレクト機能を使ってファイルに書き込むことができます）。

```
$ mkdir _test
```

```
$ date _+"%Y/%m/%d" _>_test/test8
```

- (2) `find` と `grep` コマンドを組み合わせ、`test` ディレクトリ内に"2"の文字を含むファイルを検索してください。

※`grep` コマンドに渡す際、`xargs` を最初に付け足してください。

コマンド書式： `find コマンド | xargs grep コマンド`

```
$ find _test/* _-name "*" | _xargs _grep _"2"
```

1.3. 演習問題 3 解答

ワイルドカードである“*”は、特定の文字列を指定し、その文字列を含むファイルを一度に全選択することができます。意図しないファイルにも影響を与える可能性があるため、使用する際には十分に気を付けてください。

例として、“test”という文字列を含むファイルをすべて消去することを考えます。カレントディレクトリに以下のファイルが用意されているとします。

`Readme.md`, `test_c1.c`, `testc2.c`, `test_py1.py`, `test_py2.py`

このとき、

```
$ rm _test*
```

と入力すると、カレントディレクトリには `Readme.md` のみが残ります。

また、ワイルドカードは拡張子を指定することも可能です。

例として、拡張子が“.c”であるファイルをすべて消去することを考えます。

このとき、

```
$ rm *_c
```

と入力すると、カレントディレクトリには `Readme.md`, `test_py1.py`, `test_py2.py` が残ります。

- (1) ホームディレクトリに `/sample` ディレクトリを作成し、その中に `fuga1.c`, `fuga2.c`, ..., `fuga5.c` を作成してください。（ファイルの中身の内容は任意で構いません。）

```
$ mkdir _~/sample
```

```
$ cd _sample
```

```
$ echo _c1_>_fuga1.c
```

```
$ echo _c2_>_fuga2.c
```

```
$ echo _c3_>_fuga3.c
```

```
$ echo _c4_>_fuga4.c
```

```
$ echo _c5_>_fuga5.c
```

- (2) カレントディレクトリ内に `/c_files` のディレクトリを作成してください。

```
$ mkdir _c_files
```

- (3) カレントディレクトリ内で `mv` コマンドを用いて `fuga1.c`, `fuga2.c`, ..., `fuga5.c` を同時に `/c_files` に移してください。ただし、ファイル名を直接個別に指定せず、ワイルドカード機能 `*` を使用すること。

```
$ mv *_c _c_files
```

- (4) `rename` コマンドを用いて、`/c_files` に移した `fuga1.c`, `fuga2.c`, ..., `fuga5.c` を `fuga1.txt`, `fuga2.txt`, ..., `fuga5.txt` に変更してください。ただし、ファイル名を直接個別に指定せず、ワイルドカード機能 `*` を使用すること。

`rename` コマンドはファイル名の変更を行うコマンドであり、複数のファイル名を一括で変更可能という点で `mv` コマンドと異なります。`rename` コマンドは次のように使います。

```
$ rename _置換前の文字列_置換後の文字列_対象ファイル名
```

```
$ rename _c_.txt _c_files/*
```

1.4. 演習問題 4 解答

- (1) `grep` コマンドを用いて、全ての `yml` ファイル内に”MAX”という文字列を含む行を出力してください。

```
$ grep MAX *.yml
```

実行結果

```
dynamic_param.yml:MAX_ACCELERATION: 2.0
dynamic_param.yml:MAX_YAWRATE: 1.0
dynamic_param.yml:MAX_D_YAWRATE: 6.28
dynamic_param.yml:MAX_WHEEL_ANGULAR_VELOCITY: 11.6
volumetric_param.yml:MAX_SEATING_CAPACITY: 2
```

- (2) `grep` コマンドを用いて、全ての `yml` ファイル内に”MAX”という文字列を含む行を出力してください。ただし、パイプコマンド「|」で `cat` コマンドを併用して、`grep` コマンドによって該当ファイル名がターミナル上の行頭には出力されないようにすること。

```
$ cat *.yml | grep MAX
```

実行結果

```
MAX_ACCELERATION: 2.0
MAX_YAWRATE: 1.0
MAX_D_YAWRATE: 6.28
MAX_WHEEL_ANGULAR_VELOCITY: 11.6
MAX_SEATING_CAPACITY: 2
```

1.5. 演習問題 5 解答

- (1) kouka.txt には、”shi”と’si’の表記ゆれがあります。sed コマンドを使って、表記を”shi”に統一したものを kouka.txt に上書きしてください。また、変更点がわかるようにバックアップを作成しましょう。

```
$ sed -i.bak 's/si/shi/g' kouka.txt
```

実行例

```
shirakumo nabiku surugadai  
mayu hiidetaru wakoudo ga  
tsukuya jidai no akenokane  
bunnka no ushio michibikite  
togeshi ishinn no haeninou  
oh meiji sononazo warera ga bokou  
oh meiji sononazo warera ga bokou
```

- (2) (1)で上書きした kouka.txt 内の文字列”si”が”shi”に置換されているか、grep コマンドを用いて確認しましょう。(“si”を含む列がないか確認してみましょう。)

```
$ grep si kouka.txt
```

※実行後、何も表示されなければ正解

- (3) awk コマンドを使って、同様に”si”を”shi”に置換した文字列をコマンドラインに表示しましょう。

```
$ awk '{gsub("si","shi");print $0}' kouka.txt
```

※実行例は、(1)と同じ

- (4) awk コマンドはデフォルトではスペース(空白文字)を区切り文字としています。この区切り文字を変更するには”-F”オプションを使います。seiseki.csv には、(名前,点数,評定)の順でデータが格納されています。この csv ファイルのデータを awk コマンドと”-F”オプションを使って、「名前:”name”、点数:”score”、評価:”grade”」(それぞれの...”には csv ファイル中のデータが入ります。) というようにコマンドラインに表示しましょう。

```
$ awk -F ',' '{print "名前: ",$1,"点数: ",$2,"点","評価: ",$3}' seiseki.csv
```

実行例

名前: Tatsuo 点数: 87 点 評価: A

名前: Kozo 点数: 92 点 評価: S

名前: Isao 点数: 83 点 評価: A