

オペレーティングシステム 第1回レポート課題

21T2166D 渡辺大樹

2024年5月31日

演習資料 2

問 16 - *という名前のファイルの消去

通常 shell 上において*はワイルドカードとして扱われるため、`rm *` とすると全てのファイルが消去されてしまう。そのため、*という名前のファイルを消去するためには、以下のようにする。

ソースコード 1 問 16 の解答

```
1 $ rm '*'
```

また、*という名前のファイルを作成する際には

ソースコード 2 問 16 の解答

```
1 $ touch '*'
```

を実行すればよい。

問 17 - ディレクトリの削除

指定した名前のディレクトリを削除するとき子ディレクトリやファイルもすべて削除するには、以下のようにする。

ソースコード 3 問 17 の解答

```
1 $ rm -r <ディレクトリ名>
```

問 18 - mv コマンドのオプション

`mv` コマンドにおいて第 2 引数にディレクトリ名を指定すると、第 1 引数のファイルが第 2 引数のディレクトリに移動することになる。

ソースコード 4 問 18 の解答

```
1 $ mv <ファイル名> <ディレクトリ名>
```

すなわち、例えばファイル a.txt をディレクトリ b に移動する場合は以下のようにする。

ソースコード 5 問 18 の解答

```
1 $ mv a.txt b
```

こうすることでディレクトリ b 以下に a.txt が移動される。

またこの場合、第 1 引数にファイルを複数指定することができる。

ソースコード 6 問 18 の解答

```
1 $ mv a.txt b c.txt d
```

この場合、a.txt と c.txt がディレクトリ b に、d がディレクトリ c に移動される。

問 19 - ln コマンドについて

ln コマンドはリンクを作成するコマンドである。リンクとはファイルに別の名前をつけることである。ln コマンドには -s オプションがあり、これをつけることでシンボリックリンクを作成することができる。シンボリックリンクとは、リンク先のファイルのパスを保持しているファイルのことである。

ソースコード 7 問 19 の解答

```
1 $ ln -s <リンク先のファイル> <リンク名>
```

例えば、ファイル a.txt にリンク b.txt を作成する場合は以下のようにする。

ソースコード 8 問 19 の解答

```
1 $ ln -s a.txt b.txt
```

この場合、b.txt は a.txt へのシンボリックリンクとなる。

問 20 - cp と ln の違い

問 20 では a.txt を cp,ln,ln -s でコピーした場合の違いを可視化する課題となっている。cp と ln はどちらもファイルをコピーするコマンドである。

ソースコード 9 問 20 の解答

```
1 $ echo aaaa > a.txt
2 $ cp a.txt b.txt
3 $ ln a.txt c.txt
4 $ ln -s a.txt d.txt
```

以上のコードを実行すると、a.txt,b.txt,c.txt,d.txt の内容はすべて aaaa となる。続いて、

ソースコード 10 問 20 の解答

```
1 $ echo bbbb > a.txt
```

を実行すると、a.txt の内容が bbbb に変更される。このとき、b.txt の内容は aaaa のままであるが、c.txt,d.txt の内容は bbbb に変更される。これは、cp はファイルの内容をコピーするのに対して、ln はファイルのリンクを作成するためである。また、ln -s はシンボリックリンクを作成するため、リンク先のファイルの内容が変更されるとリンク元のファイルの内容も変更される。

また続いて

ソースコード 11 問 20 の解答

```
1 $ rm a.txt
```

を実行すると、a.txt が削除される。このとき、b.txt は aaaa、c.txt は bbbb のままであるが、d.txt はリンク先のファイルが削除されたため、cat: d.txt: No such file or directory と表示される。また ls d.txt を実行すると、d.txt が赤文字で表示されるようになる。

問 21 - ディレクトリの cp

ディレクトリを対象に cp を行くと、ディレクトリ以下のファイルやディレクトリをコピーすることができる。

ソースコード 12 問 21 の解答

```
1 $ cp -r <コピー元ディレクトリ> <コピー先ディレクトリ>
```

問 22 - grep コマンド

ソースコード 13 問 22 の解答

```
1 $ cat *.txt | grep tanaka > tanaka.data
```

上記のコマンドは、全ての.txt ファイルの内容を結合し、その中から tanaka という文字列を含む行を抽出し、tanaka.data というファイルに出力するコマンドである。

問 23 - grep のパターン

grep コマンドでは、正規表現を用いてパターンを指定することができる。

ソースコード 14 問 23 の解答

```
1 $ grep '[0-9]' <ファイル名>
```

上記のコマンドは、ファイル名の中で数字が含まれる行を抽出するコマンドである。また、ほかにも

ソースコード 15 問 23 の解答

```
1 $ grep '[a-z]' <ファイル名>
```

のようにして小文字のアルファベットが含まれる行を抽出することもできる。また、数字や文字は繰り返しの指定も可能である。

ソースコード 16 問 23 の解答

```
1 $ grep '[0-9][0-9]' <ファイル名>
```

上記のコマンドは、ファイル名の中で 2 桁の数字が含まれる行を抽出するコマンドである。

問 24 - grep のオプション

1. メールアドレスの抽出

ソースコード 17 問 24-1 の解答

```
1 $ grep -E '[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}' test > result1
```

メールアドレスは、ユーザ名@ドメイン名の形式であるため、ユーザ名とドメイン名を正規表現で指定することで抽出することができる。そのため、ユーザ名を `[a-zA-Z0-9._%+-]+`、ドメイン名を `[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}` として抽出している。

2. 電話番号の抽出

ソースコード 18 問 24-2 の解答

```
1 $ grep -E '(\+81-)?0\d{1,4}-\d{1,4}-\d{4}' test > result2
```

電話番号は、0 から始まる 11 桁の数字、もしくは +81- から始まる 10 桁の数字列であるため、このパターンで抽出することができる。

3. grep のパターン

■(a). 電話番号のパターン

ソースコード 19 問 24-3(a) の解答

```
1 (\+81-)?0\d{1,4}-\d{1,4}-\d{4}
```

電話番号のパターンは以上の正規表現で表される。

■(b). 郵便番号のパターン

ソースコード 20 問 24-3(b) の解答

```
1 \d{3}-\d{4}
```

郵便番号のパターンは以上の正規表現で表される。

問 25 - ファイルパーミッション

chmod コマンドで適当なファイルの権限を操作してみる。

ソースコード 21 問 25 の解答

```
1 $ touch test.txt
2 $ ls -l test.txt
3 -rw-r--r-- 1 user user 0 May 29 21:00 test.txt
4 $ chmod u-r test.txt
5 $ ls -l test.txt
6 --w-r--r-- 1 user user 0 May 29 21:00 test.txt
7 $ chmod g+w test.txt
8 $ ls -l test.txt
9 --w--rw-r-- 1 user user 0 May 29 21:00 test.txt
10 $ chmod 777 test.txt
11 $ ls -l test.txt
12 -rwxrwxrwx 1 user user 0 May 29 21:00 test.txt
```

またディレクトリに対しても行い、変化を見してみる。

ソースコード 22 問 25 の解答

```
1 $ mkdir test
2 $ ls -ld test
3 drwxr-xr-x 2 user user 4096 May 29 21:00 test
4 $ chmod u-r test
5 $ ls -ld test
6 d-wxr-xr-x 2 user user 4096 May 29 21:00 test
7 $ chmod g+w test
8 $ ls -ld test
9 d-wxrw-r-x 2 user user 4096 May 29 21:00 test
10 $ chmod 777 test
11 $ ls -ld test
12 drwxrwxrwx 2 user user 4096 May 29 21:00 test
```

chmod コマンドではこのようにファイルやディレクトリの権限を変更できる。試しに実行パーミッションをなくしたディレクトリに対して cd コマンドを実行してみる。

ソースコード 23 問 25 の解答

```
1    $ cd test
2    bash: cd: test: Permission denied
```

このように、実行パーミッションがないディレクトリに対して cd コマンドを実行すると Permission denied と表示される。

問 26 - chgrp コマンド

chgrp コマンドはファイルやディレクトリのグループを変更するコマンドである。

ソースコード 24 問 26 の解答

```
1    $ chgrp <グループ名> <ファイル名>
```

このようにして、ファイル名のグループをグループ名に変更することができる。

問 27 - グループの追加

あるユーザ foo をグループ wheel に追加するには、以下のコマンドを実行する。

ソースコード 25 問 27 の解答

```
1    $ usermod -aG wheel foo
```

問 28 - kill コマンド

kill コマンドがデフォルトで送るシグナルは TERM である。

演習資料 3

問 4 - シェル変数

ユーザが一時的に値設定をしたシェル変数を無効化するには、以下のコマンドを実行する。

ソースコード 26 問 4 の解答

```
1    $ unset <変数名>
```

問 5 - 文字列のスクリプト

以下のスクリプトは、実行すると「Hello, world!」と現在の日付が表示される。

```
1 #!/bin/bash
2 # Greeting Script
3 greeting='Hello, world!'
4 dateInf="today is $(date +%m/%d)"
5 echo "${greeting}, ${dateInf}"
```

このスクリプトでは、まず変数「greeting」に「Hello, world!」という文字列を代入している。次に、変数「dateInf」に「today is」と現在の日付を取得するコマンド「date +%m/%d」を組み合わせた文字列を代入している。最後に、変数「greeting」と「dateInf」を結合して表示するために、echo コマンドを使用している。

実行結果は、例えば「Hello, world!, today is 07/01」といった形式で表示される。

問 6 - シェルの関数