工学基礎実験実習
ファイル操作とシェル 第 2 回レポート

氏名: 重近 大智 (SHIGECHIKA, Daichi)
学生番号: 09501527

出題日: 2019 年 5 月 21 日
提出日: 2019 年 5 月日
締切日: 2019 年 5 月 28 日

1　はじめに

ファイルリダイレクションとパイプ，シェルスクリプトやコマンドについて記述する. 機能及び具体例についても記述する.

2　ファイルリダイレクションとパイプ

2.1　リダイレクション

2.1.1　標準出力のリダイレクション

機能 一般的にコマンドは実行すると端末上に結果が表示されるが，>記号を使うことで，ファイルに結果を出力することができる. 具体的には以下のようにする.

具体例 `cat` コマンドによって出力される「solarsystem1」の表示結果を「datalist」に書き込む場合を考える.

```
$ cat solarsystem1 > datalist
$ cat datalist
Sun      0.000  1304000      1.41
Mercury  0.579        0.056  5.43
Venus    1.082        0.857  5.24
Earth    1.496        1.000  5.52
Mars     2.279        0.151  3.93
$
```

上記のようにすると，`cat` によって出力された「solarsystem1」の内容が「datalist」に出力される.

2.1.2 標準入力のリダイレクション

機能 逆に、<記号を用いることで，データなどをファイルから入力することができる．具体的には以下のようにする．

具体例 sort コマンドの並びかえの対象を「solarsystem1」から入力し，2 番目の項目を降順に並べる場合を考える．

```
$ sort -k 2 -nr < solarsystem1
Mars     2.279        0.151  3.93
Earth    1.496        1.000  5.52
Venus    1.082        0.857  5.24
Mercury  0.579        0.056  5.43
Sun      0.000  1304000      1.41
$
```

上記のようにすると，sort コマンドの並びかえの対象を「solarsystem1」から入力することができる．

2.2 パイプ

機能 あるコマンドの出力結果を別のコマンドに入力する場合，別のファイルを経由せず直接入力することができる．この時にパイプと呼ばれる機能を利用する．

具体例 「solarsystem1」というファイルの表示結果を sort コマンドの入力にする場合を考える．

```
$ cat solarsystem1 | sort -k 4 -nr
Earth    1.496        1.000  5.52
Mercury  0.579        0.056  5.43
Venus    1.082        0.857  5.24
Mars     2.279        0.151  3.93
Sun      0.000  1304000      1.41
$
```

上記のようにすると，cat コマンドによって出力された「solarsystem1」というファイルの表示結果が，sort コマンドに入力される．

3 コマンドと実行可能ファイルの関係

3.1 ソースファイルの作成

「Windows7」と表示する C 言語のプログラムのソースファイルを作ることを考える．ファイル名は「Windows7.c」とし，内容は以下の通りである．

```
1 #include<stdio.h>
  2 int main(void)
  3 {
  4     printf ("Windows7\n");
  5     return 0;
  6 }
```

3.2 ソースファイルのコンパイルと実行可能ファイルの実行

gcc コマンドを用いてコンパイルすると，実行可能ファイル「a.out」が作成され，これを端末上で実行すると以下のようになる．

```
$ gcc Windows7.c
$ ./a.out
Windows7
```

./a.out を実行すると，結果として「Windows7」が表示される．

3.3 シェルスクリプト

実行可能形式のファイルは，C 言語のソースファイルをコンパイルするだけでなく，シェルスクリプトによって作成することもできる．ただしシェルスクリプトを実行するには，事前に chmod コマンドを用いて実行パーミッションを付加する必要がある．

ここでは実行すると「Windows10」と表示するシェルスクリプト「Windows10.sh」を作成する．スクリプトは以下の通りである．

```
#!/bin/sh
echo "Windows10"
```

実行パーミッションを付加するには，次のように入力する．

```
$ chmod 755 Windows10.sh
```

C 言語の実行可能ファイル「a.out」と同様にシェルスクリプトを実行する．実行するには，次のように入力する．

```
$ ./Windows10.sh
Windows10
```

./Windows10.sh を実行すると，結果として「Windows10」が表示される．

4 その他のコマンド

ここでは前回のレポートに記述しなかったいくつかのコマンドを紹介する．今回はオプションの確認に man コマンドも用いた．

4.1 ps コマンド

ps コマンドの概要は，以下の通りである．

機能 実行中のプロセスを表示する．

形式 ps (option)

オプション
- -e： システム上の全てのプロセスを表示する．
- -u： root として実行されている全てのプロセスを表示する．

使用例 $ ps

```
  PID TTY          TIME CMD
 4064 pts/0    00:00:41 emacs
24252 pts/0    00:00:00 ps
31768 pts/0    00:00:00 bash
32556 pts/0    00:00:22 evince
$
```

ps コマンドを実行すると，上記のような結果が得られた．この結果より，エディタ (emacs)，ps コマンド，evince コマンドなどが実行されていることが分かる．ps -l とすると，次のような詳細な表示になった．

```
$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  3327  4064 31768  0  80   0 - 288841 poll_s pts/0    00:00:42 emacs
4 R  3327 31162 31768  0  80   0 - 38304 -      pts/0    00:00:00 ps
0 S  3327 31768 31756  0  80   0 - 28758 do_wai pts/0    00:00:00 bash
0 S  3327 32556 31768  0  80   0 - 359568 poll_s pts/0   00:00:23 evince
$
```

4.2 rmdir コマンド

rmdir コマンドの概要は，以下の通りである．

機能 ディレクトリを削除する．

形式 rmdir (option) [directory name]

オプション
- --ignore-fail-on-non-empty： ディレクトリが空でないため削除に失敗した場合，エラーだけを無視する．

使用例

```
$ rmdir 1
$
```

4.3 cd コマンド

cd コマンドの概要は，以下の通りである．

機能 カレントディレクトリを変更する．

形式 cd [directory name]

オプション [directory name] を入力しなければ，ホームディレクトリに戻る．

使用例
```
$ cd p1
$
```

5 ファイル操作のコマンド

5.1 cat コマンド

cat コマンドの概要は，以下の通りである．

機能 ファイル内容を連結したり，表示したりする．

形式 cat (option) [directory name]

オプション オプションは下記の通りである．

- -n： 行番号を付けて表示する．
- -v： 制御コードなどを含むファイルを表示する時に指定する．
- -b： 空行を除いて行番号を付け加える．-n より優先される．

使用例 $ cat -n bindec.c
```
 1 main(){
 2   int i,j,k,l,p=0;
 3   printf("2 進数\t10 進数\n");
 4   for(i=0;i<2;i++)
 5     for(j=0;j<2;j++)
 6       for(k=0;k<2;k++)
 7 for(l=0;l<2;l++){
 8   printf("%d%d%d%d\t%d\n",i,j,k,l,p++);
 9 }
10 }
```

5.2 less コマンド

less コマンドの概要は，以下の通りである．

機能 ファイル内容を 1 画面ごとに表示する．

形式 less [directory name]

オプション オプションは下記の通りである．ただし，ファイルを開いた状態でのみ使用できる．

- e： 1 行進む．
- y： 1 行戻る．
- f： 1 画面進む．
- b： 1 画面戻る．

使用例 $ less report1_09501527.tex

5.3 mv コマンド

mv コマンドの概要は，以下の通りである．

機能 ファイルを移動したり，ファイルの名前を変更したりする．

形式 mv (option) [directory name] [directory name]

オプション オプションは下記の通りである．ただし，-i，-f，-n を一つ以上使用した場合は最後のオプションが使用される．

- -i： 上書きの前に確認を行う．
- -f： 上書きの前に確認を行わない．
- -n： 既存のファイルには上書きしない．
- --backup： 上書き前にバックアップを作成する．

使用例
```
$ mv -i a.txt b.txt
mv: ‘b.txt’ を上書きしますか? y
  $
```

5.4 cp コマンド

cp コマンドの概要は，以下の通りである．

機能 ファイルを移動したり，ファイルの名前を変更したりする．

形式 cp (option) [directory name] [directory name]

オプション オプションは下記の通りである．

- -l： コピーの代わりにファイルのハードリンクを作成する．

- -a： ファイルの属性のみコピーする.
- --backup： 上書き前にバックアップを作成する.

使用例 $ cp a.txt p1
$

6 考察

本レポートの作成を通して，CentOS の端末下におけるいくつかのコマンドの使い方，オプションについて理解することができた. オプションを指定することで，動作が変わることに驚いた. 今までは，ls -a など，オプションも含めて 1 つのコマンドだと考えていた. 端末でコマンドを扱えるようになったことが嬉しく，もっと深く理解したくなった. 自分でも詳しく調べてみようと思う.

7 まとめ

本レポートでは，CentOS の端末下におけるいくつかのをコマンドをまとめた。またいくつかのオプションについては，端末から--help を用いて確認した.