

情報工学実験I-2  
シェルスクリプトプログラミング  
修正版

055702B 池野谷克俊  
担当教員:當間愛晃

実験日 2006年5月1日  
提出期限日 2006年5月8日  
提出した日 2006年5月10日

## 1 問題

### Level 1

指定したディレクトリ・ファイルの内容を表示するコマンド `myls` を作成せよ。但し、ディレクトリと通常のファイルは分けて表示しディレクトリの後ろには`/"`を表示すること

実行結果例

```
$ ls -FC /foo
dir1/  file1  dir2/  file2
$ myls /foo
dir1/
dir2/
file1
file2
```

### Level 2

ファイル名を一括で小文字から大文字 (`myupper`), 拡張子の変更 (`mymv`) が可能なコマンドを作成せよ。

実行結果例

```
% ls
file1.txt file2.tex file3.dat
% myupper file1.txt file2.tex
% ls
FILE1.TXT FILE2.TEX file3.dat

% ls
file1.ps file2.ps file3.ps
% mymv eps *ps
% ls
file1.eps file2.eps file3.eps
```

### Level 3

thumbnail3.sh をベースに, 1 ページ内に納める写真数 (photo\_num) を指定し, 自動的にページ分割するように拡張せよ. (下表参考)

photo_num	pagename (photos)
photo_num=1	page1.html (photo1)
	page2.html (photo2)
	page3.html (photo3)
	page4.html (photo4)
	page5.html (photo5)
photo_num=2	page1.html (photo1, 2)
	page2.html (photo3, 4)
	page3.html (photo5)
photo_num=3	page1.html (photo1, 2, 3)
	page2.html (photo4, 5)

### Level 4.1

access\_log に記載されているアクセス総数をカウントせよ.

### Level 4.2

日毎のアクセス数, IP 毎のアクセス数をカウントせよ.

### Level 4.3

Level 4.2 で求めたカウント数を棒グラフ作成せよ (何れか一つを必須とし, 残りはオプションとする).

- 日毎アクセス数: 横軸日付, 縦軸アクセス数
- IP 毎アクセス数: 横軸 IP, 縦軸アクセス数 (アクセス数トップ 10 のみで良い)

## 2 解答及び考察

### (1) Level 1

- 解答

ソースコード-myIs.sh-

```
#!/bin/sh

if [ $# -lt 1 ] ; then          # 引数チェック
    echo "Usage: prg xx "
    exit
elif [ $# -gt 1 ] ; then
    echo "引数を 1 つにしてください"
    exit
fi

#引数で指定したディレクトリが存在するか判定
if [ ! -e ./$1 ] ; then
    echo "$1 は存在しません"
    exit
fi

#引数で指定したものがディレクトリか判定
if [ ! -d ./$1 ] ; then
    echo "$1 はディレクトリではありません"
    exit
fi

#引数で指定したものがディレクトリの場合
echo "----directry----"
for file in `ls -1 $1`
do
    if [ -d ./$1/$file ] ; then
        echo "$file/"
    fi
done

echo "-----file-----"
for file in `ls -1 $1`
do
    if [ -f ./$1/$file ] ; then
        echo "$file"
    fi
done

echo "----unknown?----"
for file in `ls -1 $1`
do
    if [ ! -d ./$1/$file -a ! -f ./$1/$file ] ; then
        echo "Not directory or file"
    fi
done
```

## 実行結果

```
[j05002@myls]% ls
FILE1.TXT file3.txt myls.sh* sample1/
FILE2.TXT file4.txt myls.sh~* test1/
[j05002@myls]% sh myls.sh ./
----directry----
sample1/
test1/
-----file-----
FILE1.TXT
FILE2.TXT
file3.txt
file4.txt
mysl.sh
mysl.sh~
----unknown?----
```

### ● 考察

1. "ls -l" は存在するファイルの一覧を作成するときに 1 行に 1 つのファイルを表示する. 実行すると, ファイルの一覧を縦に列挙して表示することになる.

## (2) Level 2

### ● 解答

#### ソースコード-myupper.sh-

```
#!/bin/sh

if [ $# -lt 1 ] ; then          # 引数チェック
echo "Usage: prg xx"
exit
fi

for filename in $*
do
if [ -f $filename ] ; then     # 引数がファイルか判定
after='echo $filename | tr 'a-z' 'A-Z''
#ファイル名変更
mv $filename $after
else
echo "$filename はファイルではありません"
fi
done
```

## 実行結果

```
[j05002@myupper]% ls
file1.txt file2.txt file3.txt file4.txt myupper.sh* myupper.sh~*
[j05002@myupper]% sh myupper.sh file1.txt file2.txt
[j05002@myupper]% ls
FILE1.TXT FILE2.TXT file3.txt file4.txt myupper.sh* myupper.sh~*
```

### ● 考察

1. tr コマンドは標準入力から読み込んだ文字列を変換及び削除する. (リダイレクトを使用することでファイル中の文字列の変換, 削除も行える)

tr コマンドの構文: tr [オプション] 文字列 1 [文字列 2]

オプション: -d 文字列 1 を削除, -s 文字列 1 の繰り返しをその文字 1 文字に変換

#### ソースコード-mymv.sh-

```
#!/bin/sh

if [ $# -lt 2 ] ; then                # 引数チェック
    echo "Usage: prg xx xx"
    exit
elif [ $# -gt 2 ] ; then
    echo "引数を 2 つにして下さい"
    exit
fi

before=$1
after=$2

list='ls -l ./'
for file in $list
do
    # $file が引数で指定した拡張子が判定
    if [ $file != `basename $file .$before` ] ; then
        newfile=`basename $file .$before`.$after
        mv $file $newfile
    fi
done
```

#### 実行結果

```
[j05002@mymv]% ls
mymv.sh* mymv.sh* sample1.eps sample2.eps
[j05002@mymv]% sh mymv.sh eps ps
[j05002@mymv]% ls
mymv.sh* mymv.sh* sample1.ps sample2.ps
```

#### ● 考察

1. basename コマンドは文字列ファイル名からディレクトリ部分を取り除き, その文字列を標準出力に表示する. さらに, 引数として拡張子が指定されていた場合, ファイル名の末尾がその拡張子にマッチすれば, その拡張子も取り除いた結果を標準出力に表示する.
2. \$file が引数で指定した拡張子が判定する部分について. if 文の条件を \$file != `basename \$file .\$before`にしたのは指定した拡張子のファイルのみを変更するためである. basename \$file .\$before というコマンドを実行すると, \$file の拡張子が\$before ならば\$file の拡張子を取り除く. ということはこのコマンドを実行した前後で, \$file が変化していれば, \$file は指定した拡張子のファイルで, 変化していなければ違う拡張子のファイルといことになり, うまく判定できる.

### (3) Level 3

- 解答

ソースコード-thumbnail4.sh-

```
#!/bin/sh

# file      : thumbnail4.sh
# synopsis: thumbnail4.sh SUFFIX photo_num
# comment   : SUFFIX で指定した拡張子を持つ画像ファイルの
#             縮小画像を生成し、1 ページに photo_num で指定
#             した数の画像を表示する HTML ファイルを生成。

# 引数チェック
if [ $# -lt 3 ] ; then
    echo "Usage: $0 SUFFIX photo_num"
    echo "  RESIZE: percent"
    echo "  example) thumbnail3.sh jpg 30 2"
    exit
fi

# リストアップ処理
SUFFIX=$1
RESIZE=$2
photo_num=$3
files='ls *.$SUFFIX'
count=0      #画像の数を調べるための変数

# 縮小画像生成
## 縮小画像保存用の一時ディレクトリ生成
DIR=resized
if [ -d $DIR ] ; then
    echo "temporal directory ($DIR) exists..."
else
    mkdir $DIR
    echo "temporal directory ($DIR) CREATED."
fi

## 一時ディレクトリに縮小画像を生成
cd $DIR
for filename in $files
do
    convert -resize $RESIZE%x$RESIZE% ../$filename ${filename}_s.$SUFFIX
    echo "convert $filename to $DIR/${filename}_s.$SUFFIX"
    count='expr $count + 1'
done
cd ..

page_count=1      #HTML のページ番号
flag=1            #HTML ファイル生成の判定に使用する
photo_count=1     #画像を貼る処理に使用する
resized_files='ls $DIR/*_s.$SUFFIX'

for filename in $resized_files
do

if [ $flag -eq 1 ] ; then
# HTML ファイル生成
## HTML 開始
echo "<html>" > page$page_count.html
echo "<head>" >> page$page_count.html
echo "<title>thumbnail page$page_count</title>" >> page$page_count.html
echo "</head>" >> page$page_count.html
echo "<body>" >> page$page_count.html

    flag=0
fi
```

ソースコードは次のページに続いています

## ソースコード-thumbnaill4.sh-

```
echo "<img src=\"\$filename\">" >> page$page_count.html

#画像数が指定した数に達したか判定
if [ $photo_count -eq `expr $photo_num \* $page_count` -o
$photo_count -eq $count ]; then

    flag=1
fi

if [ $flag -eq 1 ]; then
    ## HTML 終了
    echo "</body>" >> page$page_count.html
    echo "</html>" >> page$page_count.html
    echo "page$page_count is made"
    page_count=`expr $page_count + 1`
fi

photo_count=`expr $photo_count + 1`

done
```

## 実行結果

```
[j05002@thumbnail]% sh thumbnail4.sh jpg 40 5
temporal directory (resized) exsits...
convert DSC00030.jpg to resized/DSC00030.jpg_s.jpg
convert DSC00057.jpg to resized/DSC00057.jpg_s.jpg
convert DSC00076.jpg to resized/DSC00076.jpg_s.jpg
convert DSC00149.jpg to resized/DSC00149.jpg_s.jpg
convert DSC00160.jpg to resized/DSC00160.jpg_s.jpg
convert DSC00204.jpg to resized/DSC00204.jpg_s.jpg
page1 is made
page2 is made
[j05002@thumbnail]% cat page1.html
<html>
<head>
<title>thumbnail page1</title>
</head>
<body>





</body>
</html>
[j05002@thumbnail]% cat page2.html
<html>
<head>
<title>thumbnail page2</title>
</head>
<body>

</body>
</html>
```

### ● 考察

1. flag という変数を用いれば、HTML を生成するか、画像を出力するか、HTML を終了するかを切り替えられる。



#### (4) Level 4.1 & Level4.2 & Level 4.3

- 解答

##### ソースコード

```
#!/bin/sh

#引数チェック
if [ $# -lt 1 ] ; then
    echo "引数がありません"
    exit
elif [ $# -gt 1 ] ; then
    echo "引数を 1 つにしてください"
    exit
fi

#引数で指定したファイルが存在するか判定
if [ ! -e $1 ] ; then
    echo "$1 は存在しません"
    exit
fi

#引数で指定したものがファイルかどうか判断
if [ ! -f $1 ] ; then
    echo "$1 はファイルではありません"
    exit
fi

#Level4.1 アクセス総数をカウント
echo -n "アクセス総数は "
echo `less $1 | wc -l`
echo ""

#Level4.2 日毎のアクセス数
echo "日毎のアクセス数"
cut -d ' ' -f 4 $1 | tr -d [ | cut -d ':' -f 1 | uniq -c
echo ""

#Level4.2 IP 毎のアクセス総数
echo "IP 毎のアクセス総数"
cut -d ' ' -f 1 $1 | sort | uniq -c
echo ""

#Level 4.3 の準備
##Level4.2 で求めた日毎のアクセス総数を
##ファイル"daily.data"に格納する
daily=`cut -d ' ' -f 4 $1 | tr -d [ | cut -d ':' -f 1 | uniq -c`
echo " " > daily.data

flag=1

for daily in $daily
do
    if [ $flag -eq 1 ] ; then
        data=$daily
        flag=0
    else
        echo "$daily $data" >> daily.data
        flag=1
    fi
done

# gnuplot でグラフの作成
PLOT=daily.gnuplot
echo "set terminal postscript eps" > $PLOT
echo "set output \"daily.eps\"" >> $PLOT
echo "set xdata time" >> $PLOT
echo "set timefmt \"%d/%b/%Y\"" >> $PLOT
echo "plot \"daily.data\" using 1:2 notitle with boxes" >> $PLOT
gnuplot < $PLOT
```

## 実行結果

```
[j05002@log]% sh count.sh access_log  
アクセス総数は 95
```

### 日毎のアクセス数

```
7 06/Dec/2004  
1 08/Dec/2004  
76 09/Dec/2004  
2 10/Dec/2004  
1 13/Dec/2004  
2 15/Dec/2004  
1 17/Dec/2004  
2 21/Dec/2004  
3 22/Dec/2004
```

### IP 毎のアクセス総数

```
1 133.13.48.235  
1 133.13.48.8  
1 133.13.49.18  
2 133.13.50.76  
1 133.13.52.37  
1 133.13.53.122  
2 133.13.54.156  
2 133.13.54.53  
3 133.13.57.226  
2 133.95.109.80  
1 210.139.250.215  
1 210.47.27.36  
1 218.113.200.138  
1 218.146.238.174  
1 219.160.252.142  
5 219.166.179.56  
3 220.109.8.237  
5 220.110.222.75  
3 220.20.70.41  
1 220.221.238.133  
2 220.221.239.196  
1 35.10.47.37  
3 60.34.136.173  
1 61.116.186.45  
2 61.199.170.156  
1 61.213.47.211  
6 61.78.61.166  
1 61.95.54.174  
1 64.53.90.33  
2 66.196.90.178  
1 66.196.90.207  
5 66.196.90.59  
1 66.196.90.92  
1 66.196.91.132  
1 66.196.91.175  
1 66.196.91.178  
1 66.196.91.199  
1 66.196.91.202  
1 66.196.91.203  
2 66.196.91.205  
1 66.196.91.206  
2 66.196.91.207  
2 66.196.91.216  
7 66.196.91.239  
1 66.196.91.32  
1 66.196.91.87  
1 68.121.94.147  
1 68.143.54.82  
4 82.67.110.170  
2 82.79.189.243
```



そして `cut -d ':' -f 1` で ':' (コロン) をフィールドの区切りとして、第一フィールドを取り出す。この時点でデータの抜粋は次のようになっている。

17/Dec/2004

最後に、`uniq -c` で同じ日付を一つにまとめ、その数をカウントして出力する。

### 3 感想

今回の課題は、ものすごい時間がかかった。いろんなアイデアを試しては失敗するということを何十回も繰り返して、やっとうまく実行できるようになった。

閃くか閃かないか、知ってるか知らないかで、かかる時間が大分変わる課題だと思う。得るものも大きかったが、それにしても難しい課題だった。

### 参考文献

- [1] 日経 Linux[Linux コマンド集]  
[http://homepage2.nifty.com/cs/linux\\_command/command/index.html](http://homepage2.nifty.com/cs/linux_command/command/index.html)
- [2] 基本的な文法-[シェルスクリプト リファレンス & Tips]  
<http://www.rhythm-cafe.com/shell/GrammerMenu.aspx>