ソフトウェア基礎Ⅰ

Report#2

提出日　：2009年8月11日
所　属　：工学部情報工学科
学籍番号：095739K
氏　名　：當銘　孔太

1. UNIXにおける正規表現とは何か，使い方の例を挙げて説明しなさい.

1.1 正規表現とは？

正規表現（正則表現ともいう）とは、ある規則に基づいて文字列（記号列）の集合を表す方法の1つです。ファイル名表示で使うワイルドカードも正規表現の兄弟みたいなもの。

正規表現は、ワイルドカードのような簡単な使い方から複雑な文字列表現まで、幅広い利点を提供してくれる。

1.2 正規表現で使われるメタキャラクタ

メタキャラクタ	表現	メタキャラクタ	表現
.	任意の1文字	+	1回以上の繰り返し
^	行の先頭	?	あるかないか
$	行の終わり	\	タグ付き正規表現
[ ]	範囲内の任意の1文字	|	または
*	0回以上の繰り返し	( )	グループ化

1.3 簡単な正規表現の例

\\abc	「\abc」という文字列を意味する。
a.c	「aac」「abc」「acc」...という文字列を意味する。
a*c	「c」「ac」「aac」「aaac」などの文字列を意味する。
[abc]x	「ax」「bx」「cx」などの文字列を意味する。
[^abc]x	「dx」「ex」「fx」...などの文字列を意味する。
\([ab]c\)\1	「acac」「bcbc」という文字列を意味する。

1.4 オリジナル正規表現

```
^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.ac\.jp$
```

- ^ は行の先頭。
- [a-zA-Z0-9]+ は英小文字 or 英大文字 or 数字の1回以上の繰り返し。
- @ はそのまま @ を表す。
- \.ac\.jp は「.ac.jp」を表す。

よって、この正規表現は最後が「.ac.jp」で終わるメールアドレスの検索に使える。

2．次のスクリプトを script.sh で保存し，次のスクリプトを説明しなさい．

2.1 スクリプト : script.sh

```
#!/bin/sh

# キロバイトで与えられた入力を、キロバイト、メガバイト、ギガバイトのうちの
# いずれか適切な単位で出力する．
gmk()
{
      if [ $1 -ge 1000000 ] ;
         then echo "$(scriptbc -p 2 $1 / 1000000)Gb"
      elif [ $1 -ge 1000 ] ;
         then echo "$(scriptbc -p 2 $1 / 1000)Mb"
      else
        echo "${1}Kb"
      fi
}

if [ $# -gt 1 ] ;
     then echo "Usage: $0 [dirname]" >&2; exit 1
elif [ $# -eq 1 ] ;
     then cd "$@"
fi


for file in *
do
     if [ -d "$file" ] ;
        then size=$(ls "$file" | wc -l | sed 's/[^[:digit:]]//g')
        if [ $size -eq 1 ] ;
           then echo "$file ($size entry)|"
        else
           echo "$file ($size entries)|"
        fi
     else
        size="$(ls -sk "$file" | awk '{print $1}')"
        echo "$file ($(gmk $size))|"
     fi

done | \
     sed 's/ /^^^/g' | \
     xargs -n 2 | \
     sed 's/\^\^\^/ /g' | \
     awk -F\| '{ printf "%-39s %-39s\n", $1, $2 }'
exit 0
```

2.2 解説前に...

2.2.1 関数

シェルスクリプトでも関数を作ることができ、以下のような構文となる。

```
関数名()
{
        処理......
}
```

関数に引数を渡す場合はシェルと同様にスペース区切りで渡す。
関数に渡された引数を参照する場合はシェルと同様に「$1,$2,...」と参照する。

2.2.2 数値比較

変数等を数値として評価して比較したい場合は、以下のように記述する。

数値評価演算子	意味
数値1 -eq 数値2	数値1と数値2が等しい場合に真
数値1 -ne 数値2	数値1と数値2が等しくない場合に真
数値1 -gt 数値2	数値1が数値2より大きい場合に真
数値1 -lt 数値2	数値1が数値2より小さい場合に真
数値1 -ge 数値2	数値1が数値2より大きいか等しい場合に真
数値1 -le 数値2	数値1が数値2より小さいか等しい場合に真

2.2.3 コマンド

- sed コマンド

  構文： sed [オプション] [コマンド] [ファイル名]

  コマンド

d	行を削除
-s///	各々の行で最初に一致した文字列だけ置換（s/パターン/置換文字列/）
-s///g	全体を置換（s/パターン/置換文字列/g）
-s///数値	各々の行で指定した数番目の文字列だけ置換（s/パターン/置換文字列/数値）

- awk コマンド

  構文： awk [オプション] [プログラム] [ファイル名]

  機能： 読み込んだテキスト中に指定されたパターンがないか照合し、一致するパターンが見つかった場合、指定された処理をする。

- xargs コマンド

  構文： xargs [オプション] [コマンド（引数）]

  機能： 標準入力から引数を読み込み、指定のコマンドを実行する。

2.3 解　説：script.sh

1行目
#!の後にシェルのパスを書くことで、シェルコマンドとして認識される。

3-4行目
コメント文。# の後の文は実行する際無視される。

5-14行目
関数gmk()を定義。

7-8行目
if文。引数の値が1000000より大きいか等しい場合、then以下の処理をする。
- 引数を1000000で割った値にGbをつけて値を返す。

9-10行目
elif文。引数の値が1000より大きいか等しい場合、then以下の処理をする。
- 引数を1000で割った値にMbをつけた値を返す。

11-12行目
else文。引数にKbをつけた値を返す。

16-17行目
シェルに与えられた引数の数が1より大きかったらthen以下の処理をする。
- echoで、シェルのファイル名とディレクトリ名をエラー出力とし、シェルの実行を終了。

18-19行目
シェルに与えられた引数の数が1と等しかったらthen以下の処理をする。
- cdで、引数のディレクトリへ移動する。

23-37行目
for文。現在のディレクトリ内にあるファイルやディレクトリを引数とする。

25-31行目
if文。"$file"がディレクトリならthen以下の処理をする。
- ls "$file"をwc -lで行数を取得し、sedコマンドで、先頭が10進数字でないなら、削除(置換文字列がないので..)した結果を変数sizeに代入する。

- if文。$size(ファイル数)が1と等しかったらthen以下の処理をする。
  - echoコマンドで、$file ($size entry)|（ディレクトリ名 &ファイル数）を出力。

- else文。$size(ファイル数)が1より大きい場合は
  - echoコマンドで、"$file ($size entries)|を出力。

32-35 行目

else 文。

- ls -sk "$file"で、ファイルサイズを取得。awk コマンドで、ファイル名を除く。
  得た値を変数 size に代入する。

- echo コマンドで、ファイル名とファイルサイズを出力する。
  このとき、関数 gmk に size を引数として渡し、単位変換した値を出力する。

37-41 行目

for 文の処理結果を sed → xargs → sed → awk の順にパイプでつなぐ。

- sed コマンド： 空白を「^^^」に置換する。
- xargs コマンド： 2 つ引数を読み込み、次の処理(コマンド)を実行する。
- sed コマンド： 「^^^」を空白に置換する。
- awk コマンド： 左詰め 39 文字で引数 1、引数 2 を出力する。

3. 考　察

今回のスクリプトは、どのような処理をしているのかが分かり易くて理解しやすかった。

しかし、新しいスクリプトの構文や、コマンドが多くて解説する前に教科書や Web サイトで、構文やコマンドについて調べながらレポートを作成していたので、結構時間がかかりました。

今回のスクリプトでは、ファイルやディレクリのファイルサイズを取得し、単位変換という処理でしたが、このスクリプトを応用したら便利なスクリプトができると思うので、暇なときにでも考えてみたいです。

参考文献

- 新 The UNIX Super Text[上]　　山口和紀　古瀬一隆　[監修]
- シェルスクリプト入門
  http://www.k4.dion.ne.jp/~mms/unix/shellscript/index.html
- 正規表現メモ
  http://www.kt.rim.or.jp/~kbk/regex/regex.html#CLASSNAME
- UNIX コマンド
  http://www.k-tanaka.net/unix/