

素因数分解シェルスクリプトの作成

総合情報処理センター 丹波澄雄

tanba@cc.hirosaki-u.ac.jp

1. はじめに

unix系OSのコマンドは非常に強力であり、フィルタの概念はプログラミングの方法を変えさせるほどの影響力があります。今回unix系OSのコマンドを用いてあまり役には立たないとは思いますが、素因数分解を行うシェルスクリプトを作成してみました。同等の機能をC言語を用いて書く場合よりは簡単に作成できます。

基本となったコマンドはfactorです。factorは引数を素因数分解し、その素因数を出力します。詳細はmanコマンドで確認してください。動作テストはHP-UXとLinuxで行いました。用いたシェルはcshです。コマンドの挙動が異なりましたので始めにHP-UXの場合を紹介し、次にLinuxで異なっている部分の紹介を行います。

目標とする出力フォーマットは「与えられた数値=素因数分解の結果」とします。

2. HP-UX上での作成

factorに120を引数として与えて実行させると、以下のように出力されます。

```
1 # factor 120
120
 2
 2
 2
 3
 5
```

1行目は引数の値ですので2行目以降を取り出します。

```
2 # factor 120 | tail +2
 2
 2
 2
 3
 5
```

最終行に空行が出力されていますので、この行を除外します。

```
3 # factor 120 | tail +2 | sed -e '$d'
 2
 2
 2
 3
 5
```

同じ数字の個数をカウントするためにuniqコマンドにオプション-cを付けて実行させます。

```
4 # factor 120 | tail +2 | sed -e '$d' | uniq -c
  3      2
  1      3
  1      5
```

個数と数字の順番に出力されていますので、順番を入れ替えて冪乗表現にします。

```
5 # factor 120 | tail +2 | sed -e '$d' | uniq -c | awk '{printf("%s ^ %s * ", $2, $1)}'
END {print ""}
2^3*3^1*5^1*
```

最後の乗算記号が不要であり、また1乗の部分も冗長なので削除します。

```
6 # factor 120 | tail +2 | sed -e '$d' | uniq -c | awk '{printf("%s ^ %s * ", $2, $1)}'
END {print ""}' | sed -e 's/ ^1//g' -e 's/..$//'
2^3*3*5
```

これで求めたい式の右辺が得られました。

次に左辺をfactorの出力から求めてみます。

```
7 # factor 120 | head -1
120
  等号記号=を付加して、さらに改行を抑制しておきます。
8 # factor 120 | head -1 | awk '{printf("%s = ", $1)}'
120 =
```

以上の結果を用いて、シェルスクリプトを作成します。コマンド名をfactorh.comとします。catコマンドで内容を表示させてみます。

```
9 # cat factorh.com
#!/bin/csh -f

if ($#argv != 1) then
  echo Usage : factor number
  exit 1
endif

set inv=$1

factor $inv | head -1 | awk '{printf("%s = ", $1)}'
factor $inv | tail +2 | sed -e '$d' | uniq -c | awk '{printf("%s ^ %s
* ", $2, $1)}' END {print ""}' | sed -e 's/ ^1//g' -e 's/..$//'
引数の個数が1個の時のみ動作するように引数の個数のチェックを行っています。
```

3. Linux上での作成

基本としたfactorコマンドの動作がHP-UXの場合とは異なっています。以下が実行結果です。

```
10 # factor 120
120 : 2 2 2 3 5
  素因数の部分だけを取り出します。
11 # factor 120 | cut -d : -f 2
2 2 2 3 5
```

素因数の個数を数えるためにuniqコマンドを用いたので1行に1つの数字が含まれるように行に分解します。

```
12# factor 120 | cut -d : -f2 | tr " " "\012"
```

```
2
2
2
3
5
```

1行目に空行が有りますので除外します。

```
13# factor 120 | cut -d : -f2 | tr " " "\012" | sed -e "1d"
```

```
2
2
2
3
5
```

uniqコマンドで数字の個数をカウントします。

```
14# factor 120 | cut -d : -f2 | tr " " "\012" | sed -e "1d" | uniq -c
```

```
3      2
1      3
1      5
```

この段階で4行目のコマンドの出力と同じになりました。あとは同じなので右辺は以下のようになります。

```
15# factor 120 | cut -d : -f2 | tr " " "\012" | sed -e "1d" | uniq -c | awk '{printf("%s ^ %s * ", $2, $1)} END {print ""}' | sed -e 's/ ^ 1//g' -e 's/..$//'
```

```
2 ^ 3 * 3 * 5
```

これが求めたい式の右辺となります。

次に左辺をfactorの出力から求めます。書式が異なりますので以下のようにします。

```
16# factor 120 | cut -d : -f1
```

```
120
```

これで7行目のコマンドの出力と同じになりました。あとは同じなので左辺は以下のようになります。

```
17# factor 120 | cut -d : -f1 | awk '{printf("%s = ", $1)}'
```

```
120 =
```

以上の結果から、Linux用のシェルスクリプトを作成します。コマンド名をfactorl.comとします。catコマンドで内容を表示させてみます。

```
18# cat factorl.com
```

```
#!/bin/csh -f
```

```
if ($#argv != 1) then
    echo Usage : factor number
    exit 1
endif
```

```
set inv=$1
```

```
factor $inv | cut -d : -f1 | awk '{printf("%s = ", $1)}'
```

```
factor $inv | cut -d : -f2 | tr " " "\012" | sed -e "1d" | uniq -c | awk
'{printf("%s ^ %s * ", $2, $1)} END {print ""}' | sed -e 's/ ^ 1//g' -e 's/..$//'
```

4. コマンドの統合

一応完成したのですが、unix系OSに依存せずに動作するように統合してみます。osの判別にはunameコマンドにオプション-sを指定して用います。コマンド名をfactorx.comとします。catコマンドで内容を表示させてみます。

```
19# cat factorx.com
#!/bin/csh -f

if ($#argv != 1) then
    echo Usage: factor number
    exit 1
endif

set inv=$1

set os=`uname -s`

if ("$os" == "Linux") then

factor $inv | cut -d: -f1 | awk '{printf("%s = ", $1)}'
factor $inv | cut -d: -f2 | tr " " "\012" | sed -e "1d" | uniq -c | awk
'{printf("%s ^ %s * ", $2, $1)} END {print ""}' | sed -e 's/ ^ 1//g' -e 's/..$//''

else if ("$os" == "HP-UX" || "$os" == "SunOS") then

factor $inv | head -1 | awk '{printf("%s = ", $1)}'
factor $inv | tail +2 | sed -e '1d' | uniq -c | awk '{printf("%s ^ %s * ",
$2, $1)} END {print ""}' | sed -e 's/ ^ 1//g' -e 's/..$//''

else
    echo can not execute.
endif
```

これで素因数分解シェルスクリプトは完成です。他のunix系OSでも動作させるためにはif文の中に動作するスクリプトを追加して下さい。SunOSの上でもテストしてみたところHP-UXと同じでしたのでSunOSでも動作するようにif文を修正してあります。

5. おわりに

スクリプトの便利さが感じられたならば、ぜひスクリプトで便利な小物を作成してみてください。くだらないプログラミングがかなり減少すると思います。最近パソコンは以前にもまして早くなってきており、かなりの作業がスクリプトやスクリプト言語を用いても十分な応答速度を持って実施できる様になってきています。とりあえずmanコマンドでunixのコマンド群でできることを確認してみるのが良いのではないのでしょうか。