Open Computing ‘‘Hands-On’’: ‘‘Answers to Unix’’ Column: January 1994: Listings

Listing 1: The script--originally published in the August 1993 ‘‘Answers to Unix’’ column--that converts numbers to English words.

```
 1  if echo $1 | grep -v '^[0-9]*\.[0-9]*$' > /dev/null
 2  then
 3      echo "argument must contain only numbers and a period"
 4      exit 1
 5  fi
 6  echo "$1" |
 7  awk '{dlrs = int($0)
 8        cents = int(($0 - dlrs) * 100 + .5)
 9        nbrs[0] = ""
10        nbrs[1] = "one"
11        nbrs[2] = "two"
12        nbrs[3] = "three"
13        nbrs[4] = "four"
14        nbrs[5] = "five"
15        nbrs[6] = "six"
16        nbrs[7] = "seven"
17        nbrs[8] = "eight"
18        nbrs[9] = "nine"
19        nbrs[10] = "ten"
20        nbrs[11] = "eleven"
21        nbrs[12] = "twelve"
22        nbrs[13] = "thirteen"
23        nbrs[14] = "fourteen"
24        nbrs[15] = "fiveteen"
25        nbrs[16] = "sixteen"
26        nbrs[17] = "seventeen"
27        nbrs[18] = "eighteen"
28        nbrs[19] = "nineteen"
29        nbrs[20] = "twenty"
30        nbrs[30] = "thirty"
31        nbrs[40] = "fourty"
32        nbrs[50] = "fifty"
33        nbrs[60] = "sixty"
34        nbrs[70] = "seventy"
35        nbrs[80] = "eighty"
36        nbrs[90] = "ninety"
37        if (dlrs > 1000000) {
38            print "Number greater than one million, too big"
39            exit
40        }
41        if (dlrs == 0)
42            nbrstr = "no"
43        if (dlrs >= 100000) {
44            nbrstr = nbrs[int(dlrs / 100000)] " hundred "
45            dlrs = dlrs % 100000
46            thousflag++   # should thousand be put in nbrstr
47        }
48        if (dlrs >= 1000) {
49            nbr = int(dlrs / 1000)
50            if (nbr > 20)
51               tens = int(nbr / 10) * 10
```

```
52            else
53               tens = nbr
54            if (ones = nbr - tens)
55                dash = "-"   # dash in twenty-one to ninety-nine
56            nbrstr = nbrstr nbrs[tens] dash nbrs[ones]
57            dlrs = dlrs % 1000
58            dash = ""
59            thousflag++
60        }
61        if (thousflag)
62            nbrstr = nbrstr " thousand "
63        if (dlrs >= 100) {
64            nbrstr = nbrstr nbrs[int(dlrs / 100)] " hundred "
65            dlrs = dlrs % 100
66        }
67        if (dlrs > 20)
68            tens = int(dlrs / 10) * 10
69        else
70            tens = dlrs
71        if (ones = dlrs - tens)
72            dash = "-"
73        nbrstr = nbrstr nbrs[tens] dash nbrs[ones]
74        printf "%s %2d/00\n", nbrstr, cents
75       }'
```

Listing 2. The `find.new.files` script keeps a list of every file on the system and the day it was created.

A. The source code for `find.new.files`:

```
 1  trap 'rm -f /tmp/$$ 2>/dev/null' 1 2 3 15
 2  today=`date "+%m/%d/%y"`
 3  newfilemask="Created on 99/99/99"        # used to mark new files
 4  STARTDIR=/
 5  FILELISTDIR=/usr/local/data
 6  if [ ! -f $FILELISTDIR/file.list ]; then # no file list, create one
 7      find $STARTDIR -fstype 4.2 -print |
 8      awk '{print "Created on '"$today"'\t" $0}' > $FILELISTDIR/file.list
 9  else
10      cp $FILELISTDIR/file.list $FILELISTDIR/file.list.old
11      (find $STARTDIR -fstype 4.2 -print |
12      awk '{print "'"$newfilemask"'\t" $0}'; \
13      cat $FILELISTDIR/file.list) | # combine find and data file together
14      sort -t'    ' +1 |       # sort file by path name
15      uniq -c -3 |             # count number of entries for each file name
16      egrep '^[ ]*2|^[ ]*1 '"$newfilemask"'   ' |  # ignore deleted files
17      sed 's/^.*Created/Created/
18           s.99/99/99.'$today. > /tmp/$$
19      mv /tmp/$$ $FILELISTDIR/file.list     # create new data file
20  fi
```

B. A `cron`-table entry that runs `find.new.files` every night at 11:30 pm:

```
30  23 *  *  *  /usr/local/bin/find.new.files
```

Listing 3: The `g2j` shell script that doesn't work properly.

```
# Convert GIF files to JPG format:
if ( $#argv == 0 ) then
    echo "syntax: g2j file1 [file2 ...]"
```

```
    exit 1
endif
foreach i (`ls $1`)
    echo -n "Converting " $i " ... "
    cjpeg $i > $i.JPG
    if ( $status ) then
        echo "GIF file error. Not removed."
        rm $i.JPG
    else
        echo "GIF to JPG successful."
        rm $i
    endif
end
```

Copyright © 1995 The McGraw-Hill Companies, Inc. All Rights Reserved.
Edited by Becca Thomas / Online Editor / UnixWorld Online / beccat@wcmh.com

Last Modified: Tuesday, 22-Aug-95 16:18:38 PDT