UnixWorld Online: ‘‘Answers to Unix’’ Column: 1995 Number 1: Listings

Listing 1. Files can have more than one name (or link).

A. The output of the `ls -i` command:

```
$ ls -i
30466 footer.ans
30467 header.ans
30465 feb.95
88090 testdir
$ []
```

B. Creating and listing file links:

```
$ ls -l feb.95
-rw-r--r--   1 ray      100        10959 Dec  1 10:15 feb.95
$ ln feb.95 feb.95.link
$ ls -l feb.95*
-rw-r--r--   2 ray      100        10959 Dec  1 10:15 feb.95
-rw-r--r--   2 ray      100        10959 Dec  1 10:15 feb.95.link
$ ls -i feb.95*
30465 feb.95
30465 feb.95.link
$ rm feb.95.link
$ ls -l feb.95*
-rw-r--r--   1 ray      100        10959 Dec  1 10:15 feb.95
$ []
```

C. Example linking files from one subdirectory to another:

```
$ ls -l /usr/ray/hidden
-rw-r--r--   1 ray      100         5478 Dec  3 12:43 code1.c
-rw-r--r--   1 ray      100         2234 Dec  5 09:25 code2.c
$ ln /usr/ray/hidden/code1.c /usr/ray/shared/program1.c
$ ln /usr/ray/hidden/code2.c /usr/ray/shared/program2.c
$ ls -l /usr/ray/shared
-rw-r--r--   2 ray      100         5478 Dec  3 12:43 program1.c
-rw-r--r--   2 ray      100         2234 Dec  5 09:25 program2.c
$ []
```

Listing 2. Merging addresses from a data base with a letter to create form letters.

A. Sample lines from the address data base:

```
Decotex Inc:Dutchess Court Plaza:Pawling, NY 12564:Karl Kullling
Berkley Decision Systems:803 Pine Street:Santa Cruz, CA 95062:Ray Swartz
```

B. Listing of the `merge.awk` program:

```
 1  #! /bin/sh
 2  # @(#) merge.awk  Form-letter address merge program
 3  # Address-line fields: company:address:city, state, ZIP:recipient name
 4  #
 5  if [ $# -ne 2 ]; then
 6      echo "Usage: $0 form-letter-file address-file" >&2
 7      exit 1
 8  fi
 9  (cat $1; echo "@@@" ; cat $2) |
10  awk 'addrs == 1 { printf "%s\n%s\n%s\n%s\n\n", $4, $1, $2, $3
11                    for (i = 1; i <  nbrLines; i++)
12                        print letter[i]
13                    print "^L" }
14       addrs == 0 { letter[++nbrLines] = $0 }
15       /^@@@$/ { addrs  = 1
16                 FS = ":"}'
```

C. Listing of the `merge.perl` program:

```
 1  #!/usr/local/bin/perl
 2  # @(#) merge.perl  Form-letter address merge program
 3
 4  # Rough check for valid arguments:
 5  $#ARGV == 1 || die "Usage: $0 form-letter-file address-file";
 6
 7  # Open the form letter file or bust:
 8  open(LETTER, $ARGV[0]) || die "Can't open form letter file\n" ;
 9
10  # Read contents of form letter into "letter" array:
11  @letter = <LETTER> # all lines in one gulp
12
13  # Determine number of lines in "letter" file:
14  $letlen = $#letter; # Lines in "letter" file = array element count
15
16  # Open the address file or bust:
17  open(ADDRESS, $ARGV[1]) || die "Can't open address file\n";
18
19  # Prefix address-line fields to form letter, output:
20  while ($line = <ADDRESS>) { # loop once for each address
21      chop($line);
22      @f = split(/:/, $line); # split into fields on :
23      print "$f[3]\n$f[0]\n$f[1]\n$f[2]\n\n";
24      foreach $line (@letter) {   # print the form letter
25          print $line;
26      }
27      print "\f\n";       # Separate each letter with a form-feed character
28  }
```

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

Go To Content Search

Last Modified: Tuesday, 22-Aug-95 15:55:03 PDT