Open Computing “Hands-On”: “Wizard’s Grabbag” Column: March 1994: Listings

Listing 1. The `termprint` utility provides a simple interface for printing one or more specified files on the local printer attached to the user’s terminal.

A. Listing of the `termprint` Bourne shell program:

```
#!/bin/sh
# @(#) termprint  Print data on local-terminal printer
# Author: Greg Hinkel , July 1991
# Modified by Becca Thomas , November 1993
$DBG_SH                          # Dormant debugging directive (Apr. 92)

# Remind the user to put their printer online.
echo "Make sure your printer is online." >/dev/tty; sleep 3

# Constants:
CTRL_L=`echo X | tr X '\014'`    # Store formfeed (Control-L) character
PrintOn=""                       # Hard code if don't have tput
PrintOff=""                      # Hard code if necessary
Tmpfile=/tmp/`basename $0`$$     # File containing text to print

# Set trap to erase temp file and announce signal receipt:
trap 'rm -f $Tmpfile 2>/dev/null; exit' 0
trap 'echo "Interrupted by signal" >&2; exit' 1 2 3 15

# Figure out how to do echo without newline:
if [ -z "`echo 'hello\c' | sed 's/[^c]//g'`" ]; then
    N='';  C='\c'
else
    N='-n';  C=''
fi

# Process command-line argument(s):
case $# in
    0)  # Naive users might invoke without arguments and no pipe:
        [ -t 0 ] && echo "Usage: termprint file ...">&2 && exit 1
        # It might be appropriate to replace Control-L with a
        # Control-M to avoid ejecting blank pages:
        (echo $N "$CTRL_L$C" ; cat -) > $Tmpfile ;; # Store stdin
    *)  for File in $* ; do
            if [ ! -r $File ]; then
                echo "Cannot read \"$File\"" >&2
            else
                (echo $N "$CTRL_L$C" ; cat $File) >> $Tmpfile
            fi
        done ; shift ;;
esac

#  Exit if there are no files to print:
if [ ! -s "$Tmpfile" ]; then
    echo "Exiting, no readable files were located" >&2 ; exit 2
fi

# Get local-terminal printer-control codes if not already defined:
```

```
: ${PrintOn:="`tput mc5`"}      # Get printer-on and
: ${PrintOff:="`tput mc4`"} # printer-off capability

# Make sure the printer-on and printer-off capabilities exist:
if [ ! "$PrintOn" -o ! "$PrintOff" ]; then
    echo "Your terminal \"$TERM\" can't control your local printer" >&2
    exit 3
fi

# Output printer-on code, files, formfeed, printer-off code:
echo $N "$PrintOn$C"; cat $Tmpfile;
echo $N "$CTRL_L$C"; echo $N "$PrintOff$C"
```

B. Sample usage:

```
$ termprint file

$ pr file | termprint
```

Listing 2. Source for Jerry Peek's `tcap` Program.

```
static char *rcsid = "$Header$";
/*
 **     tcap - termcaps for shell scripts.
 **     Usage: tcap [-a] cap
 */
#include <stdio.h>
#include <ctype.h>

extern char *tgetdefstr(), *tgetstr(), *tgoto();
extern int tgetdefnum(), tgetnum();

typedef struct tbl
{   char *str;
    short val;
} lookuptbl;

#define STRING  1
#define STR1ARG 2
#define STR2ARG 3
#define NUMBER  4
#define FLAG    5

lookuptbl tcaps[] = {
        "AL",  STR1ARG,
        "Bj",  STRING,
        "Bl",  STRING,
        "Br",  STRING,
        "CC",  STRING,
        "Cj",  STRING,
        "CM",  STR2ARG,
        "DC",  STR1ARG,
        "DL",  STR1ARG,
        "DO",  STR1ARG,
        "EP",  FLAG,
        "Ge",  STRING,
        "Gs",  STRING,
        "HD",  FLAG,
        "Hl",  STRING,
        "IC",  STR1ARG,
```

```
"K1",  STRING,
"K2",  STRING,
"K3",  STRING,
"K4",  STRING,
"K5",  STRING,
"LC",  FLAG,
"LE",  STR1ARG,
"Lj",  STRING,
"NL",  FLAG,
"OP",  FLAG,
"RI",  STR1ARG,
"Rj",  STRING,
"SF",  STR1ARG,
"SR",  STR1ARG,
"Tj",  STRING,
"Tl",  STRING,
"Tr",  STRING,
"UC",  FLAG,
"UP",  STR1ARG,
"Vl",  STRING,
"Xc",  STRING,
"ae",  STRING,
"al",  STRING,
"am",  FLAG,
"as",  STRING,
"bc",  STRING,
"bl",  STRING,
"bs",  FLAG,
"bt",  STRING,
"bw",  FLAG,
"cd",  STRING,
"ce",  STRING,
"ch",  STR1ARG,
"cl",  STRING,
"cm",  STR2ARG,
"co",  NUMBER,
"cr",  STRING,
"cs",  STR2ARG,
"ct",  STRING,
"cv",  STR1ARG,
"dB",  NUMBER,
"dC",  NUMBER,
"dF",  NUMBER,
"dN",  NUMBER,
"dT",  NUMBER,
"dV",  NUMBER,
"da",  FLAG,
"db",  FLAG,
"dc",  STRING,
"dl",  STRING,
"dm",  STRING,
"do",  STRING,
"ds",  STRING,
"ec",  STR1ARG,
"ed",  STRING,
"ei",  STRING,
"eo",  FLAG,
"es",  FLAG,
"ff",  STRING,
"fs",  STRING,
"gn",  FLAG,
```

```
"hc",  FLAG,
"hd",  STRING,
"ho",  STRING,
"hs",  FLAG,
"hu",  STRING,
"hz",  FLAG,
"ic",  STRING,
"if",  STRING,
"im",  STRING,
"in",  FLAG,
"ip",  STRING,
"is",  STRING,
"it",  NUMBER,
"k0",  STRING,
"k1",  STRING,
"k2",  STRING,
"k3",  STRING,
"k4",  STRING,
"k5",  STRING,
"k6",  STRING,
"k7",  STRING,
"k8",  STRING,
"k9",  STRING,
"kA",  STRING,
"kC",  STRING,
"kD",  STRING,
"kE",  STRING,
"kF",  STRING,
"kH",  STRING,
"kI",  STRING,
"kL",  STRING,
"kM",  STRING,
"kN",  STRING,
"kP",  STRING,
"kR",  STRING,
"kS",  STRING,
"kT",  STRING,
"ka",  STRING,
"kb",  STRING,
"kd",  STRING,
"ke",  STRING,
"kh",  STRING,
"kl",  STRING,
"km",  FLAG,
"kn",  NUMBER,
"ko",  STRING,
"kr",  STRING,
"ks",  STRING,
"kt",  STRING,
"ku",  STRING,
"l0",  STRING,
"l1",  STRING,
"l2",  STRING,
"l3",  STRING,
"l4",  STRING,
"l5",  STRING,
"l6",  STRING,
"l7",  STRING,
"l8",  STRING,
"l9",  STRING,
"le",  STRING,
```

```
"li",  NUMBER,
"ll",  STRING,
"lm",  NUMBER,
"ma",  STRING,
"mb",  STRING,
"md",  STRING,
"me",  STRING,
"mh",  STRING,
"mi",  FLAG,
"mk",  STRING,
"ml",  STRING,
"mm",  STRING,
"mo",  STRING,
"mp",  STRING,
"mr",  STRING,
"ms",  FLAG,
"mu",  STRING,
"nc",  FLAG,
"nd",  STRING,
"nl",  STRING,
"ns",  FLAG,
"nw",  STRING,
"os",  FLAG,
"pO",  STR1ARG,
"pb",  NUMBER,
"pc",  STRING,
"pf",  STRING,
"po",  STRING,
"ps",  STRING,
"pt",  FLAG,
"rP",  STRING,
"rc",  STRING,
"rf",  STRING,
"rp",  STR2ARG,
"rs",  STRING,
"sa",  STR2ARG,
"sc",  STRING,
"se",  STRING,
"sf",  STRING,
"sg",  NUMBER,
"so",  STRING,
"sr",  STRING,
"st",  STRING,
"ta",  STRING,
"tc",  STRING,
"te",  STRING,
"ti",  STRING,
"ts",  STR1ARG,
"uc",  STRING,
"ue",  STRING,
"ug",  NUMBER,
"ul",  FLAG,
"up",  STRING,
"us",  STRING,
"vb",  STRING,
"ve",  STRING,
"vi",  STRING,
"vs",  STRING,
"vt",  NUMBER,
"wi",  STRING,
"ws",  NUMBER,
```

```
        "xb",  FLAG,
        "xn",  FLAG,
        "xo",  FLAG,
        "xr",  FLAG,
        "xs",  FLAG,
        "xt",  FLAG,
        "xv",  FLAG,
        "xx",  FLAG,
};

putx(c) { putchar (c); }

main (argc, argv)
    int argc;
    char **argv;
{
    char tcbuf[1024];
    char *cp, *arg;
    int i, tgetnum(), tgetflag();
    static int aflag = 0;

    if (tgetent(tcbuf, getenv ("TERM")) <= 0) {
        (void) fprintf (stderr, "tcap: cannot find termcap for %s.\n",
                getenv ("TERM"));
        exit (1);
    }
    cp = (char *) malloc (256);
    argv++;
    while (*argv) {
        arg = *argv++;
        if (strcmp (arg, "-a") == 0) {
            aflag = 1;
            continue;
        }
        i = lookup (arg, tcaps);
        if (i < 0) {
            (void) fprintf (stderr, "tcap: no such termcap: %s\n", arg);
            exit (2);
        }
        switch (tcaps[i].val & 017) {
        case STRING:
            if (aflag)
                show (tgetdefstr (arg, &cp));
            else
                tputs (tgetdefstr (arg, &cp), 1, putx);
            break;
        case STR1ARG:
            if (aflag)
                show (tgetstr (arg, &cp));
            else
                if (argv != NULL && isdigit (**argv))
                    tputs (tgoto (tgetstr (arg, &cp),
                        atoi (*argv++), 0), 1, putx);
                else {
                    (void) fprintf (stderr, "tcap: error in arguments to %s\n",
                            arg);
                    exit(3);
                }
            break;
        case STR2ARG:
            if (aflag)
```

```
                show (tgetstr (arg, &cp));
            else
                if (argv != NULL && argv[1] != NULL &&
                    isdigit (**argv) && isdigit (*(argv[1])))
                    tputs (tgoto (tgetstr (arg, &cp),
                        atoi (*argv++), atoi (*argv++)), 1, putx);
                else {
                    (void) fprintf (stderr, "tcap: error in arguments to %s\n",
                                arg);
                    exit(3);
                }
            break;
        case NUMBER:
            if (aflag)
                (void) printf ("%d\n", tgetdefnum (arg));
            break;
        case FLAG:
            if (aflag)
                (void) printf (tgetflag (arg) ? "true\n" : "false\n");
            break;
        }
    }
    exit (0);
}



#define MAXLUPN 3      /* longer than longest possible name */

/* Lookup name in table.  Will take nonambiguous abbreviations.  If
   you want to insist that a certain table entry must be spelled
   out, enter it twice in the table.  Table entries must be sorted
   by name, and a name which is a substring of a longer name comes
   earlier in the table.  Accepts upper or lower case if table entry
   is lower case.

   Returns:
    > 0 table entry index
     -1 not found
     -2 ambiguous
*/
int lookup (name, table)
    char *name;
    lookuptbl *table;
{
    register char  *namptr,
                   *tblptr;
    int ind;
    int value = 0;
    short length;
    short longest = 0;
    int ambig = 0;
    char lname[MAXLUPN];

    if (name == NULL)
        return -1;
    namptr = name;
    tblptr = lname;
    for (;;) {
        if ((*tblptr++ = isupper (*namptr)? tolower (*namptr++): *namptr++)
            == '\0')
```

```
            break;
        if (tblptr >= &lname[MAXLUPN])
            return -1;
    }

    for (ind = 0; (tblptr = table->str) != 0; table++, ind++) {
        namptr = lname;
        for (; *tblptr == *namptr; tblptr++, namptr++) {
            if (*tblptr == '\0')
                break;
        }
        if (*namptr == '\0') {  /* end of name or exact match */
            length = namptr - lname;
            if (longest < length) {
                longest = length;
                ambig = 0;
                value = ind;
                if (*tblptr == '\0')
                    break;          /* exact match */
            }
            else /* longest == length */
                ambig = 1;
        }
        else if ( *namptr < *tblptr )
            break;
    }
    if (ambig)
        return -2;
    if (longest)
        return value;
    return -1;
}

show (str)
    char *str;
{
    char c;

    if (str == (char *)0) {
        (void) printf ("NULL\n");
        return;
    }
    while (c = *str++ & 127)
        if (c < ' ') {
            if (c == 27)
                (void) printf ("\n");
            (void) printf ("^%c", c + '@');
        } else if (c == 127)
            (void) printf ("^?");
        else
            (void) printf ("%c", c);
    (void) printf ("\n");
    return;
}

char *
tgetdefstr (cap, ptr)
    char *cap, *ptr;
{
    char *x;
    static char *bs = "\b",
```

```
                *ht = "\t",
                *nl = "\n",
                *cr = "\r",
                *ff = "\f",
                *nu = "\0",
                *bl = "\007",
                *nlcr = "\n\r";

    if ((x = tgetstr (cap, ptr)) != (char *) 0)
        return (x);
    if (strcmp (cap, "pc") == 0)
        return (nu);
    if (strcmp (cap, "bc") == 0)
        return (bs);
    if (strcmp (cap, "ta") == 0)
        return (ht);
    if (strcmp (cap, "nl") == 0)
        return (nl);
    if (strcmp (cap, "cr") == 0)
        return (cr);
    if (strcmp (cap, "ff") == 0)
        return (ff);
    if (strcmp (cap, "nw") == 0)
        return (nlcr);
    if (strcmp (cap, "bl") == 0)
        return (bl);
    if (strcmp (cap, "ho") == 0)
        return (tgoto (tgetstr ("cm", ptr), 0, 0));
    if (strcmp (cap, "ll") == 0)
        return (tgoto (tgetstr ("cm", ptr), tgetnum ("li") - 1, 0));
    if (strcmp (cap, "rs") == 0)
        return (tgetstr ("is", ptr));
    if (strcmp (cap, "is") == 0)
        return (tgetstr ("rs", ptr));
    if (strcmp (cap, "rf") == 0)
        return (tgetstr ("if", ptr));
    if (strcmp (cap, "if") == 0)
        return (tgetstr ("rf", ptr));

    return ((char *) 0);
}


tgetdefnum (cap)
    char *cap;
{
    int x;

    if ((x = tgetnum (cap)) != -1)
        return (x);
    if (strcmp (cap, "ws") == 0)
        return (tgetnum ("co"));
    return (0);
}
```

Listing 3: A script can be used to manage `.rhosts` files by renaming them.

A. Listing of the script:

```
#! /bin/sh
```

```
# rhyes/rhno Enable/disable use of remote .rhosts file
# Author: Jerry Peek, May 1992
# set -x        # uncomment for debugging

# Get name of invoked script:
Thisprog="`basename $0`"

# Get account name of invoking user:
if [ "$USER" ]; then
    Username=$USER
elif [ "$LOGNAME" ]; then
    Username=$LOGNAME
else
    echo "Can't determine your account name." >&2; exit 2
fi

# Determine home directory for user:
if [ "$HOME" ]; then
    Homedir=$HOME
elif [ "$LOGDIR" ]; then
    Homedir=$LOGDIR
else
    echo "Can't determine your home directory name." >&2; exit 3
fi

# Handle invocation without a remote host name:
case $# in
    0)  # No argument specified, so rename local file:
        case "$Thisprog" in
            rhyes)  if [ ! -f $Homedir/.rhosts.moved ]; then
                        echo "Exiting, no $Homedir/.rhosts.moved" >&2
                        exit 2
                    fi
                    mv $Homedir/.rhosts.moved $Homedir/.rhosts ;;
            rhno)   if [ ! -f $Homedir/.rhosts ]; then
                        echo "Exiting, no $Homedir/.rhosts" >&2
                        exit 2
                    fi
                    mv $Homedir/.rhosts $Homedir/.rhosts.moved ;;
            *)  echo "$Thisprog: Invalid program name" >&2
                exit 1 ;;
        esac
        exit 0 ;;
esac

# Create proper file-name order for rename operation:
case "$Thisprog" in
    rhyes)  Cmdargs=".rhosts.moved .rhosts" ;;
    rhno)   Cmdargs=".rhosts .rhosts.moved" ;;
    *)      echo "$0: Help!  Shouldn't get here!" 1>&2
            exit 1 ;;
esac

# Perform operation for all remote hosts:
for host
    do
        echo "Connecting to $host ..."
        cat << ENDCMDS | ftp -n $host   # Don't use .netrc
    user $Username
    rename $Cmdargs
    quit
```

```
ENDCMD
done
```

B. Sample usage session:

```
%  rhyes snake
Password:   ...type snake password
%  rsh snake some-command
    ...
%  rhno snake
Password:   ...type snake password
%  rsh snake some-command
Permission denied.
% []
```

Edited by Becca Thomas / Online Editor / UnixWorld Online / beccat@wcmh.com

Go To Content   Search

Last Modified: Wednesday, 23-Aug-95 17:09:29 PDT