

How to update Les Bell's year-old telephone directory program.

A Better Mousetrap...

By Lloyd R Borrett

AFTER TAKING delivery of my IBM-PC microcomputer, I had plenty to do. There were manuals to read and examples to practise with. Finally, I sat down and attempted to program the beast.

Unfortunately, the only programming language available was BASIC, and the IBM-PC dialect was new to me. After searching through magazines for a suitable example, I decided to implement an indexed telephone directory program, written by *Your Computer* editor Les Bell. The program had been developed for the "BASIC for Birdwatchers" tutorial in the July 1982 edition.

It proved to be a simple task to convert the program for the IBM-PC. However, I wasn't pleased with the program's structure and, once running, some operational limitations became obvious.

I proceeded to create an improved version of the program by changing to a more efficient sort algorithm, modifying the user interface, making allowance for duplicated names and re-structuring the program to use more sub-routines.

Code that is written in BASIC can be difficult to understand and maintain at the best of times. Effective use of the GOSUB statement is one of the easiest ways to improve the structure of a BASIC program.

The GOSUB statement can also reduce the amount of duplicated code. For example, the code to display a file record is duplicated in four places in the original program.

The major sub-routines I've used are:

```
620 ' *** Add a name to file ***
1090 ' *** Sorted list of file to screen ***
1170 ' *** Display specified names ***
1360 ' *** Delete name from file ***
1590 ' *** Find comment ***
```

The minor sub-routines used are:

```
1740 ' *** Routine to display a record ***
1850 ' *** Binary Search ***
2040 ' *** Sort KEYS$ and POSN ***
2180 ' *** Pack pointer arrays and file ***
2400 ' *** Find next record with same name ***
2510 ' *** Hold or Quit display ***
2580 ' *** Put uppercase 1$ in UPPER$ ***
```

The RETURN statement is only used as the last executable statement of these "sub-routines". This forces a 1-in, 1-out control structure.

In his original version of the program, Les Bell made no allowance for duplicated names. The binary search would always locate the same entry if there was more than one with the same name.

I have added some code which, upon finding a match, steps back through the index to find the first name that matches (lines 1940 to 1960). A pointer to that name is returned. It was then necessary to add a "sub-routine" to return a pointer to the next name that matches (lines 2400 to 2500). These "sub-routines" are used in the sections which find surnames (lines 1170 to 1350) and delete names (lines 1360 to 1580).

Les Bell described the sort used in the original version as a standard Shellsort. While it worked, it was s-l-o-w...

After consulting *Algorithms Plus Data Structures Equals Programs* (Niklaus Wirth, Prentice-Hall, 1976) and *The Art of Computer Programming, Volume 3, Sorting and Searching* (Donald Knuth, Addison-Wesley), I decided to try a shuttle interchange sort.

I set up an array of 250 randomly ordered names and ran the two procedures over them. The shuttle interchange sort took 493 seconds, and the Shellsort took 560 seconds. Only a minor improvement.

But in this program, the typical case is a few names being sorted into the existing list. So I ran a procedure which would add five names to a list of 250 sorted names.

The shuttle interchange sort took 20 seconds, and the Shellsort took 506 seconds. Now, that's some difference! I decided to use the shuttle interchange sort!

I can think of ways to speed up the sort even further, and it's possible that another algorithm may also prove faster, but for the time being I shall stick with what I've got.

While there are many ways in which this program could be further improved, I've achieved the goals I set for myself when I started modifying it. So, for the time being, I'll concentrate on using it, instead of playing with it.

While working on a program, there comes a point in time when you feel any extra effort on your part can't be adequately returned. Les Bell probably felt he'd reached that point when he finished the original version.

I've reached that point now. How much further can you take it?

*teldir.bas", listed on 03-15-1983 at 13:26:13

```
10 ' *** Address / Telephone Directory ***
20 ' TELDIR Revision 1.00
30 ' Written by Lloyd Borrett, March 1983.
40 ' Based on a program by Les Bell, "Your Computer",
    July 1982.
50 ' "BASIC For Birdwatchers", Part IX, page 54.
60 '
70 ' *** Initialize ***
80 DEFINT A-Z
90 DEF FNLN$(L) = STRING$(L,95)
    ' Used to produce a string of L underscores.
100 MAXREC = 256 ' Maximum number of records per file
110 TDELAY = 500 ' Time delay constant
120 NONE = 0
    ' No records constant, not in range 1 to maxrec
130 DELETED = NONE ' Used to flag a record as deleted.
140 NO = 0 ' Constant value for NO
150 YES = 1 ' Constant value for YES
160 DIBO = ASC("0")
170 DIM KEYS$(256), POSN(256)

180 ' *** Open the files ***
190 ' Get the file name
200 CLS : LOCATE 3,1
210 INPUT "Enter the directory name: (TELDIR) ";FILE$
220 IF FILE$="" THEN FILE$="TELDIR"

230 ' Read the index file
240 ON ERROR GOTO 510
250 OPEN "I", #1, FILE$+".IND"
260 TOPREC = 1 ' Pointer to the next available record
270 IF EOF(1) THEN CLOSE 1 : GOTO 290
280 INPUT #1, KEYS$(TOPREC), POSN(TOPREC) :
    TOPREC = TOPREC + 1 : GOTO 270
290 LOCATE 5,1 :
    PRINT TOPREC-1;" entries in directory ";FILE$;".
300 FOR I = 1 TO TDELAY : NEXT I

310 ' Open the directory file
320 OPEN "R", #2, FILE$+".TDR", 128
330 FIELD #2, 20 AS FSNAME$, 20 AS FCNAME$, 30 AS FADDR$,
    20 AS FCITY$, 4 AS FPSTCD$, 15 AS FTEL$, 19 AS
    FCOMNT$

340 ' *** Display Menu ***
350 SCREEN 0,1 : COLOR 7,0 : CLS
360 LOCATE 3,1 :
    PRINT "TELDIR Revision 1.00 on ";DATE$;" at ";TIME$
370 LOCATE 5,1 :
    PRINT " *** Address / Telephone Directory *** :
    LOCATE 8,1
380 PRINT " 1. Find name (Search directory for
    a surname)"
390 PRINT " 2. Find comment (Search directory for
    a comment)"
400 PRINT " 3. View directory (Display the full
    directory)"
410 PRINT " 4. Add name (Create new directory
    entries)"
420 PRINT " 5. Delete name (Remove directory
    entries)"
430 PRINT " 6. Exit (Quit and return to
    DOS)"
440 PRINT : PRINT "Press choices: ";

450 ' Get and process the choice.
460 I$ = INKEY$: IF I$ = "" THEN 460
470 IF I$ < "1" OR I$ > "6" THEN BEEP : GOTO 460
480 ON ASC(I$)-DIBO GOSUB 1180,1600,1100,640,1370,570
490 GOTO 350

500 ' *** Handle errors ***
510 BEEP : IF ERR <> 53 THEN 540
520 CLOSE :
    PRINT "Directory ";FILE$;
    " doesn't exist. Creating it."
530 TOPREC = 1 : RESUME 320
540 LOCATE 23,20 :
    PRINT " *** Error:ERR;"in line";ERL;". : STOP

550 ' *** Exit to DOS ***
560 ' Write index file
570 OPEN "O", #1, FILE$+".IND"
580 FOR I = 1 TO TOPREC-1 : WRITE #1, KEYS$(I), POSN(I) :
    NEXT I : CLOSE
590 REM
600 CLS : SYSTEM

620 ' *** Routine to add a name to file ***
630 REM
640 ADDFLG = NO

650 ' Is there room for more ?
```

```

660 IF TOPREC <= MAXREC THEN 710
670 CLS: LOCATE 5,1:
PRINT "There's no room for more names in directory
";FILE
680 BEEP:
PRINT "Unless you delete some of the existing
names."
690 FOR I = 1 TO TDELAY: NEXT I: GOTO 1070

700 ' Get another name
710 CLS: LOCATE 5,1:
PRINT "Create new directory entries." : LOCATE 7,1
720 PRINT "Surname": ;FNLN$(20)
730 PRINT "First Name": ;FNLN$(20)
740 PRINT "Street": ;FNLN$(30)
750 PRINT "Town/City": ;FNLN$(20)
760 PRINT "Postcode": ;FNLN$(4)
770 PRINT "Telephone": ;FNLN$(15)
780 PRINT "Comment": ;FNLN$(19)

790 ' Get the details
800 LOCATE 7,13: INPUT SNAME$: IF SNAME$ = "" THEN 1070
ELSE I$ = SNAME$: GOSUB 2590
810 LOCATE 7,13: PRINT "": SNAME$;SPACE$(20-LEN(SNAME$))
820 LOCATE 8,13: INPUT CNAME$
830 LOCATE 8,13: PRINT "": CNAME$;SPACE$(20-LEN(CNAME$))
840 LOCATE 9,13: INPUT ADDR$
850 LOCATE 9,13: PRINT "": ADDR$;SPACE$(30-LEN(ADDR$))
860 LOCATE 10,13: INPUT CITY$
870 LOCATE 10,13: PRINT "": CITY$;SPACE$(20-LEN(CITY$))
880 LOCATE 11,13: INPUT PSTCD$
890 LOCATE 11,13: PRINT "": PSTCD$;SPACE$(16)
900 LOCATE 12,13: INPUT TEL$
910 LOCATE 12,13: PRINT "": TEL$;SPACE$(15-LEN(TEL$))
920 LOCATE 13,13: INPUT COMNT$
930 LOCATE 13,13: PRINT "": COMNT$;SPACE$(19-LEN(COMNT$))

940 ' Write record to disk.
950 LSET F$NAME$ = SNAME$
960 LSET F$CNAME$ = CNAME$
970 LSET F$ADDR$ = ADDR$
980 LSET F$CITY$ = CITY$
990 LSET F$PSTCD$ = PSTCD$
1000 LSET F$TEL$ = TEL$
1010 LSET F$COMNT$ = COMNT$
1020 PUT #2, TOPREC

1030 ' Add record to index and go back to get more records
1040 KEYS$(TOPREC) = UPPER$: POSN(TOPREC) = TOPREC:
TOPREC = TOPREC + 1
1050 ADDFLG = YES: GOTO 660

1060 ' Finish off adding names. Sort and return to the
main menu.
1070 IF ADDFLG = YES THEN GOSUB 2050
1080 RETURN

1090 ' *** Sorted listing of file to the screen ***
1100 FOR POINTER = 1 TO TOPREC-1:
IF POSN(POINTER) = DELETED THEN 1150

1110 ' Get and display the record
1120 GET #2, POSN(POINTER): GOSUB 1750
1130 ' Hold or quit
1140 GOSUB 2520: IF QUIT = YES THEN 1160
1150 NEXT POINTER
1160 RETURN

1170 ' *** Display specified names ***
1180 CLS: LOCATE 3,1:
PRINT "Search the directory for a surname."
1190 LOCATE 5,1: INPUT "Enter surname to find: [Exit] ";I$:
IF I$="" THEN 1350

1200 ' Convert to all uppercase and search for it.
1210 GOSUB 2590: SNAME$ = UPPER$: GOSUB 1890:
IF POINTER = NONE THEN 1310

1220 ' Get and display the record
1230 GET #2, POSN(POINTER): GOSUB 1750
1240 LOCATE 20,20:
PRINT "Press SPACE BAR to continue, ESC to quit.";
1250 I$ = INKEY$: IF I$ = "" THEN 1250
1260 IF I$ = "" THEN 1290
1270 IF I$ = CHR$(27) THEN 1330 ELSE BEEP: GOTO 1250

1280 ' Any more with the same name?
1290 GOSUB 2440: IF POINTER = NONE THEN 1330 ELSE 1230

1300 ' Unsuccessful search
1310 LOCATE 7,15: BEEP: PRINT SNAME$;" not found.";
FOR I=1 TO TDELAY: NEXT I

1320 ' Go back for more names to display
1330 GOTO 1180

1340 ' Finished displaying names

```

```

1350 RETURN

1360 ' *** Routine to delete name from file ***
1370 DELFLG = NO:
1380 CLS: LOCATE 3,1: PRINT "Delete directory entries."
1390 LOCATE 5,1: INPUT "Enter surname to delete: [Exit] ";I$:
1400 IF I$ = "" THEN 1570
1410 GOSUB 2590: SNAME$ = UPPER$: GOSUB 1890:
IF POINTER = NONE THEN 1530

1420 ' Display it
1430 GET #2, POSN(POINTER): GOSUB 1750
1440 LOCATE 20,20: PRINT "Delete? (Y/N) "
1450 I$ = INKEY$: IF I$ = "" THEN 1450
1460 IF I$ = "Y" OR I$ = "y" THEN 1490
1470 IF I$ = "N" OR I$ = "n" THEN 1510 ELSE BEEP: GOTO 1450

1480 ' Flag the record as deleted?

1490 POSN(POINTER) = DELETED: DELFLG = YES

1500 ' Any more with the same name?
1510 GOSUB 2440: IF POINTER = NONE THEN 1550 ELSE 1430

1520 ' Unsuccessful search
1530 LOCATE 7,15: BEEP: PRINT SNAME$;" not found.";
FOR I=1 TO TDELAY: NEXT I

1540 ' Go back for more names
1550 GOTO 1380

1560 ' Finish off deleting names. Pack and return to the
main menu.
1570 IF DELFLG = YES THEN GOSUB 2200
1580 RETURN

1590 ' *** Find comment ***
1600 CLS: LOCATE 3,1:
PRINT "Search the directory for a comment."
1610 LOCATE 5,1: INPUT "Enter comment to search for:
[Exit] ";I$:
1620 IF I$ = "" THEN 1730 ELSE GOSUB 2590: COMNT$ = UPPER$

1630 ' Test every directory record for the comment
1640 CLEN = LEN(COMNT$)
1650 FOR POINTER = 1 TO TOPREC-1: GET #2, POINTER
1660 I$ = LEFT$(F$COMNT$,CLEN): GOSUB 2590

1670 ' If it matches, display it
1680 IF UPPER$ = COMNT$
THEN GOSUB 1750: GOSUB 2520: IF QUIT = YES
THEN 1710
1690 NEXT POINTER

1700 ' Go back for more comments to search for
1710 GOTO 1600

1720 ' Finished searching for comments
1730 RETURN

1740 ' *** Routine to display a record ***
1750 CLS: LOCATE 5,1: PRINT "Entry:
";POINTER;" Position: ";POSN(POINTER)
1760 LOCATE 7,1
1770 PRINT "Surname": ;F$NAME$
1780 PRINT "First Name": ;F$CNAME$
1790 PRINT "Street": ;F$ADDR$
1800 PRINT "Town/City": ;F$CITY$
1810 PRINT "Postcode": ;F$PSTCD$
1820 PRINT "Telephone": ;F$TEL$
1830 PRINT "Comment": ;F$COMNT$
1840 RETURN

1850 ' *** Binary Search ***
1860 ' Finds the first occurrence of SNAME$ in
KEYS$(MAXREC).
1870 ' The index position is returned in POINTER.
1880 ' Set upper and lower boundaries
1890 POINTER = NONE: LOWER = 1: UPPER = TOPREC - 1

1900 ' Try the middle of the current interval
1910 IF UPPER < LOWER THEN 2030 ELSE I = (LOWER + UPPER) \ 2
1920 IF KEYS$(I) < SNAME$ THEN LOWER = I + 1: GOTO 1910
1930 IF KEYS$(I) > SNAME$ THEN UPPER = I - 1: GOTO 1910

1940 ' We have a match. Get first of possible duplicates.
1950 IF I <= 1 THEN 1980
1960 IF KEYS$(I - 1) < SNAME$ THEN 1980
ELSE I = I - 1: GOTO 1950

1970 ' Find first record not deleted.
1980 IF POSN(I) < DELETED THEN 2020
1990 IF KEYS$(I) < KEYS$(I + 1) THEN 2030
2000 I = I + 1: IF I > TOPREC - 1 THEN 2030 ELSE 1980

2010 ' Put index position in POINTER and/or return
2020 POINTER = I
2030 RETURN

```

```

2040 ' *** Sort KEYS$(maxrec) and POSN(maxrec)
2050 CLS: LOCATE 5,1:
PRINT "Sorting the directory. Please wait."
2060 J = 1
2070 IF J > TOPREC - 2 THEN 2160
2080 IF KEYS$(J) <= KEYS$(J+1) THEN 2150
2090 SWAP KEYS$(J), KEYS$(J+1): SWAP POSN(J), POSN(J+1):
PRINT ". ";
2100 I = J - 1
2110 IF I < 1 THEN 2150
2120 IF KEYS$(I) <= KEYS$(I+1) THEN 2140
2130 SWAP KEYS$(I), KEYS$(I+1): SWAP POSN(I), POSN(I+1):
PRINT ". ";
2140 I = I - 1: GOTO 2110
2150 J = J + 1: GOTO 2070
2160 PRINT
2170 RETURN

2180 ' *** Pack pointer arrays and file ***
2190 ' Pack pointer arrays
2200 CLS: LOCATE 5,1:
PRINT "Packing the directory. Please wait." : D = 1
2210 FOR I = 1 TO TOPREC - 1: IF POSN(I) = DELETED
THEN 2240
2220 IF D < I
THEN KEYS$(D) = KEYS$(I): POSN(D) = POSN(I)
2230 D = D + 1
2240 NEXT I

2250 ' Were any records deleted?
2260 IF D = TOPREC THEN 2390

2270 ' Yes. Pack file
2280 P = 1
2290 FOR J = 1 TO TOPREC - 1
2300 FOR I = 1 TO D-1: IF POSN(I) = J THEN 2340
ELSE NEXT I

2310 ' Record not needed
2320 GOTO 2360

2330 ' Record still needed
2340 IF P < J THEN GET #2, J: PUT #2, P: POSN(I) = P
2350 P = P + 1
2360 NEXT J
2370 IF P < D
THEN PRINT "ERROR:
File and pointer array do not match": STOP
2380 TOPREC = D
2390 RETURN

2400 ' *** Find the next record with the same name
***
2410 ' Name is in SNAME$, current position in index is
in POINTER.
2420 ' Returns POINTER =
NONE if no more records with the same name,
2430 REM otherwise POINTER points to a new position in the
index.
2440 I = POINTER: POINTER = NONE
2450 I = I + 1: IF I >= TOPREC THEN 2500
2460 IF SNAME$ < KEYS$(I) THEN 2500
2470 IF POSN(I) = DELETED THEN 2450

2480 ' Found one
2490 POINTER = I
2500 RETURN

2510 ' *** Hold or quit display ***
2520 QUIT = YES: LOCATE 20,20:
PRINT "Press SPACE BAR to hold, ESC to quit."
2530 FOR I = 1 TO TDELAY: I$ = INKEY$: IF I$ = ""
THEN 2560
2540 IF I$ = "" THEN I = 1: GOTO 2560
2550 IF I$ = CHR$(27) THEN 2570 ELSE BEEP
2560 NEXT I: QUIT = NO
2570 RETURN

2580 ' *** Put an all uppercase version of I$ in
UPPER$ ***
2590 UPPER$ = I$: IF LEN(I$) = 0 THEN 2640
2600 CONVERT = ASC("a") - ASC("A")
2610 FOR I = 1 TO LEN(I$): I1$ = MID$(I$,I,1)
2620 IF I1$ = "a" AND I1$ <= "z"
THEN MID$(UPPER$,I,1) = CHR$(ASC(I1$) - CONVERT)
2630 NEXT I
2640 RETURN

```

