

BEEBUG

FOR THE
BBC MICRO &
MASTER SERIES

Graffer



- Select
 - 1: Demons
 - 2: Functi
 - 3: Time pl
 - 4: Time hi
 - 5: Input fr
 - 6: Suppressed zeros (default: keyboard)
 - 7: Full divisions
 - 8: Titles
 - 9: Manual Boundaries
- PRESS ϕ OR \langle P

• MINOTAUR'S MAZE

• MULTIPLYING THE MAIL

• CONVERTING NEWLINE CODES IN TEXT

• INTER-MAIL

FEATURES

Graffer	
Minotaur's Maze	
Putting Directories to Work (2)	13
First Course: File Handling (2)	17
Converting Newline Codes in Text	21
Mr Toad's Machine Code Corner	24
512 Forum	
Public Domain Software	
M-Base (3)	
Wordwise User's Notebook:	
Multiplying the Mail	41

REVIEWS

Inter-Mail	36
------------	----

REGULAR ITEMS

Editor's Jottings/News	4
Postbag	47
RISC User	49
Personal Ads	52
Hints and Tips	53
Subscriptions & Back Issues	54
Magazine Disc	55

HINTS & TIPS

Faster Basic	
Which day Is It?	
Testing OSBYTE Routines	
Invisible Boots	

PROGRAM INFORMATION

All listings published in BEEBUG magazine are produced directly from working programs. They are formatted using LISTO 1 and WIDTH 40. The space following the line number is to aid readability only, and may be omitted when the program is typed in. However, the rest of each line should be entered exactly as printed, and checked carefully. When entering a listing, pay special attention to the

difference between the digit one and a lower case l (L). Also note that the vertical bar character (Shift \) is reproduced in listings as |.

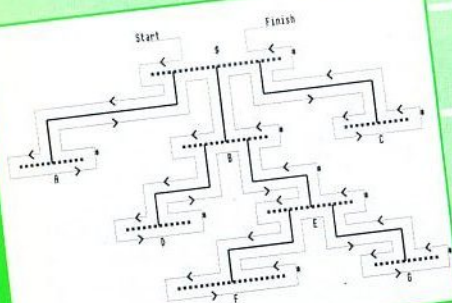
All programs in BEEBUG magazine will run on any BBC micro with Basic II or later, unless otherwise indicated. Members with Basic I are referred to the article on page 44 of BEEBUG Vol.7 No.2 (reprints

MENU

Select number keys as required

- 1: Demonstration plot
 - 2: Function plot
 - 3: Time plot
 - 4: Time histogram
 - 5: Input from file (default: keyboard)
 - 6: Suppressed zeros
 - 7: Full divisions
 - 8: Titles
 - 9: Manual Boundaries
- PRESS 0 OR <RTN> TO END SELECTION

Graffer



Putting Directories to Work

M-Base : No File

Options :

- 1) View records
- 2) Add records
- 3) Delete records
- 4) New or alter database
- 5) View information
- 6) Load file
- 7) Save file
- 8) Search for records
- 9) Status
- 0) Exit

Select 1-0 :

M-Base



Minotaur's Maze

INTER-MAIL

(C) Synectics 1991

- 1) Save letter on disc
- 2) Find old letter on disc
- 3) Find address
- 4) Enter new address
- 5) Change Options
- 6) Print letter
- 7) Start new letter 4.245
- 8) Start Mailshot
- 9) Utilities

14th November 1993
ESC INTER-WORD menu
<f0> To return here

Inter-Mail

Phone : 0011-112233

The Very Small Mammal Society,
23 Smogton Vale Crescent,
Hemelworth New Town,
Hertingfordshire,
H11 4HH.

20th April 1993

#Address

1993 VOLE APPEAL

Dear

Many thanks for your contribution to our 1993 special appeal. Although the amount raised, has yet to be worked out, it does appear that we will reach, if not pass, our target figure.

As you are aware, the Society is engaged in an extensive small mammal educational support programme. The money raised this year will enable us to provide better vocational training for voles, an area which has suffered severely from lack of funds following last year's cuts in Government spending. Funds

Multiplying the Mail

available on receipt of an A5 SAE), and are strongly advised to upgrade to Basic II. Any second processor fitted to the computer should be turned off before the programs are run.

Where a program requires a certain configuration, this is indicated by symbols at the beginning of the article (as shown opposite). Any other requirements are referred to explicitly in the text of the article.



Program needs at least one bank of sideways RAM.



Program is for Master 128 and Compact only.

Editor's Jottings

Welcome to this first issue of BEEBUG for 1994. As we stated at the start of volume 12, the decline in the BBC micro market had reached the point where we believed it would cease to be commercially viable for us to continue with the publication of BEEBUG beyond the issue for April 1994. Nothing has happened to change this view, and so there will be just two more issues of the magazine before we take our leave of you, our readers, for the last time.

As I have said before on this page, there is still plenty of life left in the many BBC micros which are still in frequent and widespread use. But it is not something which we can continue to support as a commercial venture. If support for the BBC micro is to continue at all, as I have no doubt it will, then it must return to the roots from which BEEBUG itself developed, in the form of user groups, run and organised by enthusiastic and dedicated users, with the support of users and for the benefit of users.

A number of such user groups have already come to my attention, some of them new some of them already established. This month I want to give some initial details. I must stress that BEEBUG has no connection with any of these groups, and the mention of any group on this page should not be interpreted as a recommendation on our part. We merely hope that the information may prove useful to readers in the future.

Mr.A.G.Nelson has initiated a new club called the **Beeb Developments User Group**, which you may have seen advertised recently in BEEBUG. The address to write to is 73 Spital Crescent, Newbiggin-by-Sea, Northumberland NE64 6SQ, and the telephone number is 0670 521055.

Then there is **8 Bit Software**, an established user group dedicated to all users of the BBC range. It offers low cost public domain software, and a monthly disc-based magazine. The group is run by Chris Richardson at 17 Lambert Park Road, Hedon, Hull HU12 8HF, tel. 0482 896868.

ByteBack is a similar user group, which publishes a monthly A5 sized magazine. This group is run enthusiastically by Paul Harvey, at 33 King Henry Mews, Enfield Lock, Middx EN3 6JS.

There is also a long-standing user group called **Solinet**, originally set up to provide support for users of Solidisc products, but for some years more of a general purpose BBC user group. This also offers a disc-based magazine. The contact is Ron Marshall, 41 Westbrook Drive, Rainworth, Mansfield, Nottingham NG21 0FB.

Another group just emerging into existence, goes under the name of **Destroyed Realities Disc Based Magazine**. The first issue was due for publication in January this year on a 40T single-sided DFS disc for about £1.80. The organiser is David Lowless of 82 Main Street, Pembroke, Dyfed, Wales SA71 4HH.

Remember all of these are privately run from private addresses. If you want more information then I am sure a stamped addressed envelope would be very much appreciated, as I am sure would be contributions of programs and articles and other offers of help.

Finally, if any of the organisations I have mentioned would like to update me on their activities, or any groups I have omitted let me know of their existence, then the relevant details can be included in either of our next two issues.

M.C.W.

Graffer (Part 1)

Some clever plotting by George Crossley

Graffer is a menu-driven graph and histogram plotting program for the BBC Master. It supports a number of data sources and presentation styles.

USING GRAFFER

Type in Graffer and save it to an ADFS disc. You will also need to create directories *F* and *T*.



The Graffer menu showing range of options available

The menu (Fig.1) gives you the choice of five types of graph or histogram, input from keyboard or file and four presentation options. Press the appropriate number keys for the configuration you want, then 0 or Return to end your selection. On ending the selection process, your chosen options are displayed; they are modified if you've made incompatible choices.

If you choose the Demonstration Plot the program plots a simple graph without further input, unless you have chosen option 8 or 9 as well. An example Function Plot is included, too; if you choose this option, answer 'Y' to the query 'Is the function defined?'. Any

other response ends the program with an invitation to define a function at line 1330. Options 3 and 4 generate graphs and histograms with time as the X-axis. You will be prompted to input Y coordinates and to choose Months or Years as the time co-ordinate. The default option generates general purpose graphs, with prompts for X and Y coordinates. Option 5, selected with one of the above options, draws graphs/histograms from data stored on file. File generation programs will be provided next month.

The rest of the options are meant to be used along with those already described. Option 6 gives Suppressed Zeros - the default is to plot X and Y axes, even if the graph doesn't touch or cross them. Option 7 plots dashed divisions at chosen intervals. Option 8 includes main, X-axis and Y-axis titles, all input from keyboard. Option 9 allows selection of graph boundaries, the default being for Graffer to select scales to fill the screen.

When you've made your menu selections Graffer plots a graph or histogram on the shadow screen. Pressing any key replaces the plot with another menu, together with some rubbish at the top and bottom of the screen. This is caused by program workspace encroaching on screen memory; your graph is preserved in shadow RAM.

The second menu gives you a choice of five options:

Run Program destroys your graph and starts the program again.

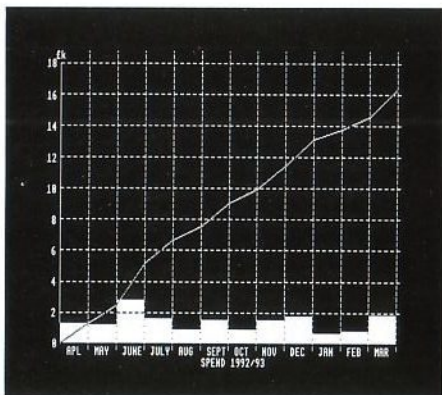
Graffer

Plot Another Graph on Same Axes prompts for further input and adds a second graph to the original one.

Save Graph to File invokes the Savescreen program provided in part 2.

Print Graph invokes Dump2, again, this will be in part 2. You can, however, provide your own mode 128 screen dump at line 2260.

Quit ends the program, having first reset text and graphics windows.



Combining line graph and bar chart

Fig.2 is a graph of cumulative spend superimposed on a time histogram of monthly spend. The histogram was plotted first, by selecting main menu options 4, 7, 8 and 9, specifying Months as the time axis, 12 as the number of months and 4 (April) as the start month. The Y-axis monthly spend data and titles were typed in and the Y-axis boundary set to a value which would accommodate the cumulative graph. After checking that the histogram looked OK, option 2 was selected from the second menu and the cumulative spend figures were entered.

PROBLEMS

If the main menu responds in an odd way to your choices, try doing a hard break and reloading the program. This is necessary after using many screen dump programs.

NEXT MONTH

Subsidiary programs which allow data files to be generated and saved for use with the main program and dump the final graph to printer or save it to file will be provided next month.

```
10 REM Program Graffer
20 Version B1.0
30 REM Author George Crossley
40 REM BEEBUG Jan/Feb 1994
50 REM Program Subject to Copyright
60 :
100 MODE 135
110 ON ERROR IF ERR=17 THEN RUN ELSE P
ROError
120 PRINT TAB(8,12) CHR$(131);CHR$(141
);"WELCOME TO GRAFFER":PRINT TAB(8,13) C
HR$(131);CHR$(141);"WELCOME TO GRAFFER"
130 TIME=0:REPEAT UNTIL TIME=400:CLS
140 DIM T(51,1)
150 REPEAT
160 FILE=0:STUB=1:TITL=0:FPLT=0:DEMO=0
:SUPP=0:TIM=0:HIST=0:INPT=0:SCAL=0
170 MORE=0:N=2:XT$="":YT$="":TI$="":XT
LEN=0:YTLEN=0:TILEN=0:R=0:S=0:Z=0
180 MODE 135
190 PRINT TAB(14,2) CHR$(131);CHR$(141
);"MENU":PRINT TAB(14,3) CHR$(131);CHR$(
141);"MENU"
200 PRINT 'CHR$(136);CHR$(130);" Selec
t number keys as required"
210 PRINT "" 1: Demonstration plot""
2: Function plot"" 3: Time plot"" 4
: Time histogram"" 5: Input from file
(default: keyboard)"
220 PRINT "" 6: Suppressed zeros"" 7
: Full divisions"" 8: Titles"" 9: Ma
nual Boundaries"
230 PRINT 'CHR$(134);" PRESS 0 OR <RTN
> TO END SELECTION"
```

```

240 REPEAT
250 X=GET
260 IF X=49 THEN DEMO=1
270 IF X=50 THEN FPLT=1
280 IF X=51 THEN TIM=1
290 IF X=52 THEN HIST=1
300 IF X=53 THEN FILE=1
310 IF X=54 THEN SUPP=1
320 IF X=55 THEN STUB=0
330 IF X=56 THEN TITL=1
340 IF X=57 THEN SCAL=1
350 Y=X-48:IF 0<Y AND Y<10 THEN PRINT
TAB(0,2*Y+3) CHR$(130);Y
360 UNTIL X=13 OR X=48
370 MODE 128
380 IF DEMO=1 THEN TIM=0:FILE=0:FPLT=0
:HIST=0
390 IF HIST=1 THEN TIM=1:SUPP=0
400 IF TIM=1 OR FILE=1 THEN FPLT=0
410 IF FILE=0 AND FPLT=0 AND DEMO=0 AN
D TIM=0 THEN INPT=1
420 PRINT""Your selection is:""
430 IF INPT=1 THEN PRINT"Input from ke
yboard" ELSE IF FILE=1 THEN PRINT"Input
from file"
440 IF DEMO=1 THEN PRINT"Demonstration
plot" ELSE IF FPLT=1 THEN PRINT"Funcio
n plot"
450 IF TIM=1 THEN PRINT"Time plot"
460 IF HIST=1 THEN PRINT"Histogram"
470 IF STUB=0 THEN PRINT"Full division
s"
480 IF SUPP=1 THEN PRINT"Suppressed ze
ros"
490 IF TITL=1 THEN PRINT"Titles (input
from keyboard)"
500 IF SCAL=1 THEN PRINT"Manual graph
boundaries"
510 PRINT""Accept this selection? (Y
or N)":O=GET
520 UNTIL O=89 OR O=121
530 IF INPT=1 THEN PROCinput
540 IF FILE=1 THEN PROCfile
550 IF FPLT=1 THEN PROCfplt
560 IF TIM=1 THEN PROCtim
570 IF DEMO=1 THEN PROCdemo ELSE PROCc
alc
580 VDU29,ZEROX;ZEROY;
590 IF TITL=1 THEN PROCtitle

```

```

600 PROCaxes
610 IF HIST=0 THEN PROCplot ELSE PROCh
ist
620 PRINT"?";:O=GET:PRINT CHR$(127)
630 PROCopt
640 END
650 :
1000 DEF PROCinput
1010 REPEAT:INPUT "No. of data points (
50 max.) "N:UNTIL N>1 AND N<51
1020 REPEAT:FOR A=1 TO N:PRINT"INPUT X"
;A", Y";A;:INPUT " " T(A,0), T(A,1):NEX
T A
1030 PRINT"Accept these? (Y or N)":P=GE
T
1040 UNTIL P=89 OR P=121
1050 ENDPROC
1060 :
1070 DEF PROCfile:LOCAL X
1080 IF TIM=0 THEN *DIR F
1090 IF TIM=1 THEN *DIR T
1100 CLS:*
1110 INPUT "Input file name: " A$
1120 X=OPENIN A$
1130 INPUT#X,N,TIM
1140 FOR A=0 TO N
1150 INPUT#X,T(A,0):INPUT#X,T(A,1)
1160 NEXT A
1170 CLOSE#X
1180 *DIR^
1190 ENDPROC
1200 :
1210 DEF PROCfplt
1220 PRINT"Is the function defined (Y o
r N)? ":Q=GET
1230 IF Q=89 OR Q=121 THEN INPUT "Input
XMIN, XMAX " XMIN, XMAX ELSE PRINT"Def
ine single-valued FNx(X) at line 1330":E
ND
1240 N=51
1250 CALCIN=(XMAX-XMIN)/(N-1)
1260 T(1,0)=XMIN:T(1,1)=FNx(XMIN)
1270 FOR A=1 TO N-1
1280 PT=XMIN+(A*CALCIN)
1290 T(A+1,0)=PT:T(A+1,1)=FNx(PT)
1300 NEXT A
1310 ENDPROC
1320 :
1330 DEF FNx(X)=COS(X)*EXP(-X/10)

```

```

1340 :
1350 DEF PROCcalc
1360 IF TIM=1 THEN XMAX=T(0,0):YMAX=T(0
,1):XMIN=T(0,0):YMIN=T(0,1) ELSE XMAX=T(
1,0):YMAX=T(1,1):XMIN=T(1,0):YMIN=T(1,1)
1370 FOR A=1 TO N
1380 IF T(A,0)>XMAX THEN XMAX=T(A,0)
1390 IF T(A,1)>YMAX THEN YMAX=T(A,1)
1400 IF T(A,0)<XMIN THEN XMIN=T(A,0)
1410 IF T(A,1)<YMIN THEN YMIN=T(A,1)
1420 NEXT A
1430 IF TIM=1 THEN XMIN=0
1440 IF XMIN>0 AND SUPP=0 THEN XMIN=0
1450 IF XMAX<0 AND SUPP=0 THEN XMAX=0
1460 IF YMIN>0 AND SUPP=0 THEN YMIN=0
1470 IF YMAX<0 AND SUPP=0 THEN YMAX=0
1480 IF SCAL=1 THEN PROCscal
1490 XRG=XMAX-XMIN:YRG=YMAX-YMIN
1500 IF DEMO=0 THEN PRINT'"XRG "XRG,"Y
RG "YRG'
1510 IF DEMO=0 AND TIM=0 THEN REPEAT IN
PUT "Division intervals: X axis, Y axis
(min. XRG/10, YRG/10) "XINT, YINT:UNTIL
XINT>=XRG/10 AND YINT>=YRG/10
1520 IF TIM=1 THEN XINT=1:REPEAT INPUT"
Y axis division interval (min. YRG/10) "
YINT:UNTIL YINT>=YRG/10
1530 UNITX=1079/XRG:UNITY=903/YRG
1540 XI=UNITX*XINT:YI=UNITY*YINT
1550 XMI=UNITX*XMIN:XMA=UNITX*XMAX:YMI=
UNITY*YMIN:YMA=UNITY*YMAX
1560 ZEROX=1170-XMA:ZER0Y=982-YMA
1570 ENDPROC
1580 :
1590 DEF PROCplot
1600 PLOT 4,T(1,0)*UNITX,T(1,1)*UNITY
1610 IF TIM=1 THEN PLOT 4,0,T(0,1)*UNIT
Y:PLOT 6,T(1,0)*UNITX,T(1,1)*UNITY
1620 FOR A=2 TO N
1630 PLOT 6,T(A,0)*UNITX,T(A,1)*UNITY
1640 NEXT A
1650 ENDPROC
1660 :
1670 DEF PROCdemo *
1680 N=3
1690 T(1,0)=-100:T(1,1)=-100:T(2,0)=0:T
(2,1)=100:T(3,0)=100:T(3,1)=-100
1700 XINT=25:YINT=30

```

```

1710 PROCcalc
1720 PRINT'"XRG "XRG,"YRG "YRG
1730 PRINT"XINT"XINT,"YINT"YINT
1740 IF TITL=0 THEN XT$="XUnit":YT$="YU
nit":TI$="Demo Graph":XTLEN=5:YTLEN=5:TI
LEN=10
1750 TIME=0:REPEAT UNTIL TIME=200:CLS
1760 ENDPROC
1770 :
1780 DEF PROCtitle
1790 PRINT:INPUT LINE "Graph Title "TI$
1800 TILEN=LEN(TI$)
1810 REPEAT INPUT LINE "X Axis Title (5
letters or less) "XT$
1820 INPUT LINE "Y Axis Title (5 letter
s or less) "YT$
1830 XTLEN=LEN(XT$):YTLEN=LEN(YT$)
1840 UNTIL XTLEN<6 AND YTLEN<6
1850 ENDPROC
1860 :
1870 DEF PROCtim
1880 INPUT "Input Months or Years " M$
1890 IF FILE=0 OR PFLT=1 THEN PROCtimin
1900 S$="M":R$="Y":S=INSTR(M$,S$):R=INS
TR(M$,R$)
1910 IF R=1 THEN INPUT "Input first yea
r "YR:DIM Y(N):FOR A=1 TO N:Y(A)=YR:YR=Y
R+1:NEXT A
1920 IF S=1 THEN PROCmonths
1930 ENDPROC
1940 :
1950 DEF PROCtimin
1960 REPEAT PRINT"Number of "M$"? (max
12) ";;INPUT ""N:UNTIL N>0 AND N<13
1970 REPEAT FOR A=0 TO N:T(A,0)=A:PRINT
"INPUT Y";A;;INPUT " "T(A,1):NEXT A
1980 PRINT"Accept these? (Y or N)":P=GE
T
1990 UNTIL P=89 OR P=121
2000 ENDPROC
2010 :
2020 DEF PROCmonths
2030 REPEAT INPUT "Input start month (J
an=1) "START
2040 UNTIL START>0 AND START<13
2050 DIM MT$(N+START):DIM MTH$(N)
2060 FOR A=1 TO (START+N):READ MT$(A):N
EXT A

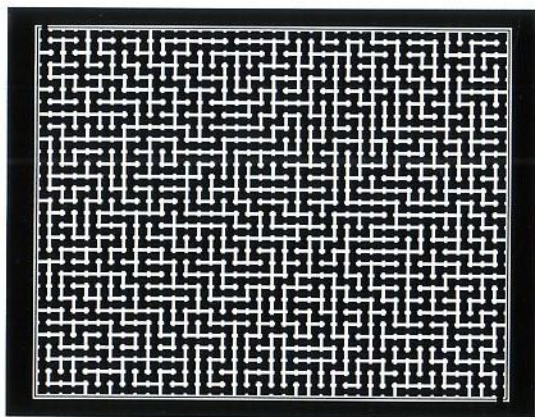
```

Continued on page 50

Minotaur's Maze

Miroslaw Bobrowski with a lesson in getting lost.

This contribution from Poland is an excellent example of the speed gained by using machine code. Setting up mazes can be a fairly straight-forward job in Basic, so long as you don't mind the maze being unsolvable. Creating a route through the maze can take an awful lot of checking and this is what takes the time.



A typical maze

This program does this very quickly and could form the basis for lots of different games. Type the program in and save it to disc. On running it you will be asked how large you want the maze to be, anything up to 38 by 30 squares, which makes a fairly impressive maze of 1140 cells.

The maze is drawn in mode 4 and appears very quickly indeed with two entrances - don't worry about the complexity, it is solvable! Once the maze is on the screen the program checks to see if your printer is ready, prompts you if it isn't, then prints the screen out.

CHANGES AND IMPROVEMENTS

The printer dump is for a nine pin Epson compatible printer and the dump code starts at line 2640. It is assembled from &900 and could easily be replaced with your own driver, but keep in mind that this is in mode 4.

A simple improvement would be to offer the option to print rather than going straight to it. This would let you look at several mazes and print the ones you like best. *FNready* checks for the printer at the start of the program, you could move this down the sequence and put a simple 'YES/NO' prompt before *PROCprinter* is called.

Beyond that, a screen dump could be useful for putting mazes into DTP packages, and you might want to try and make it work in mode 0 - this would give you the option to increase the maze size. Using other modes could give the addition of colour.

```
10 REM Program Minotaur's Maze
20 REM Version B 1.0
30 REM Author Miroslaw Bobrowski
40 REM BEEBUG Jan/Feb 1994
50 REM Program subject to copyright
60 :
100 MODE 7:PROCchar:PROCsetsize
110 DIM code 2000:PROCassemble
120 *FX 5,1
130 REPEAT UNTIL FNready
140 MODE 4:VDU 23,1,0;0;0;0;
150 CALL buildmaze
160 PROCborder
170 PROCprinter
```

Minotaur's Maze

```
180 MODE 7:END
190 :
1000 DEF PROCsetsize
1010 FOR Y%=1 TO 2:PRINTTAB(8,Y%)CHR$14
1CHR$129CHR$157CHR$131"MINOTAUR'S MAZE
"CHR$156:NEXT
1020 PRINT"CHR$134"Enter number of col
umns"CHR$129"(4-38) : "CHR$131:REPEAT:INP
UTTAB(34,5)SPC5TAB(34,5)""H%:IF H%<4 OR
H%>38 SOUND1,-12,1,10:UNTIL FALSE ELSE U
NTIL TRUE
1030 PRINT"CHR$134"Enter number of rows
"CHR$129"(4-30) : "CHR$131:REPEAT:INPU
TTAB(34,7)SPC5TAB(34,7)""V%:IF V%<4 OR V
%>30 SOUND1,-12,1,10:UNTIL FALSE ELSE UN
TIL TRUE
1040 M%=H%*V%-1:R%=20-H%DIV2:Q%=1036-32
*(V%+2)
1050 ENDPROC
1060 :
1070 DEF PROCchar
1080 VDU 23,224,0;0;0;0;23,225,195,129,
129,129,129,129,129,255,23,226,255,129,1
28,128,128,128,129,255,23,227,195,129,12
8,128,128,128,129,255,23,228,255,129,129
,129,129,129,195,23,229,195,129,129,
129,129,129,129,195
1090 VDU 23,230,255,129,128,128,128,128
,129,195,23,231,195,129,128,128,128,128,
129,195,23,232,255,129,1,1,1,1,129,255,2
3,233,195,129,1,1,1,1,129,255,23,234,255
,129,0,0,0,0,129,255,23,235,195,129,0,0,
0,0,129,255
1100 VDU 23,236,255,129,1,1,1,1,129,195
,23,237,195,129,1,1,1,1,129,195,23,238,2
55,129,0,0,0,0,129,195,23,239,195,129,0,
0,0,0,129,195
1110 ENDPROC
1120 :
1130 DEF FNready
1140 *FX 21,3
1150 Z%=ADVAL(-4):VDU2,1,0,1,0,3
1160 IF ADVAL(-4)<=Z%VDU7:PRINT"TAB(7
)CHR$133"Get your printer ready""TAB(9)C
HR$133"then press any key":REPEAT UNTIL
GET:PRINT:=FALSE
1170 =TRUE
```

```
1180 :
1190 DEF PROCborder
1200 VDU 7,30:MOVE R%*32+24,999:DRAW (R
%+H%)*32+4,999:DRAW (R%+H%)*32+4,987-32*
V%:DRAW (R%+H%)*32-8,987-32*V%
1210 MOVE (R%+H%)*32-28,987-32*V%:DRAW
R%*32-8,987-32*V%:DRAW R%*32-8,999:DRAW
R%*32+4,999
1220 ENDPROC
1230 :
1240 DEF PROCprinter
1250 CALL &900
1260 VDU 2,1,27,1,87,1,1
1270 PRINT"TAB(7)"Minotaur's Maze No.";
!&7D AND &FFFFFFF
1280 VDU 1,27,1,87,1,0,3
1290 ENDPROC
1300 :
1310 DEF PROCassemble
1320 cell=&70:tcell=&72:row=&74:col=&75
1330 index=&76:dir=&77:counter=&78
1340 lim1=&7A:lim2=&7B:tempa=&7C
1350 rnd=&7D:oswrch=&FFEE
1360 :
1370 FOR pass=0 TO 2 STEP 2
1380 P%=code:[OPT pass
1390 .buildmaze
1400 LDA &FE68:BEQ buildmaze:STA rnd
1410 LDA #0:STA lim1:LDA #H%:STA lim2
1420 JSR random:STA col
1430 LDA #0:STA lim1:LDA #V%:STA lim2
1440 JSR random:STA row
1450 JSR calccellno
1460 LDA#0:STA counter:STA counter+1
1470 .main
1480 JSR check
1490 LDA index:CMP #1:BEQ choosedir
1500 STA lim2:INC lim2:LDA #1:STA lim1
1510 JSR random:STA index
1520 .choosedir
1530 LDX index:LDA tq,X:STA dir:TAX
1540 LDY #0:CLC:LDA (cell),Y:ADC power,
X:STA (cell),Y
1550 JSR display
1560 LDA kk,X:BEQ findrow
1570 BMI minus:INC col:BNE findrow
1580 .minus:DEC col
```

```

1590 .findrow
1600 LDA wk,X:BEQ newcell
1610 BMI minus2:INC row:BNE newcell
1620 .minus2:DEC row
1630 .newcell
1640 JSR calccellno
1650 DEC dir:DEC dir:LDA dir:BPL newdir
1660 ADC #4
1670 .newdir
1680 STA dir:TAX
1690 LDY #0:CLC:LDA (cell),Y:ADC power,
X:STA (cell),Y
1700 JSR display
1710 INC counter:LDA counter:BNE checke
nd
1720 INC counter+1
1730 .checkend
1740 CMP #(M%MOD256):BNE notend
1750 LDA counter+1:CMP #(M%DIV256):BNE
notend
1760 LDA #0:STA col:STA row:JSR calccel
lno
1770 LDY#0:CLC:LDA (cell),Y:ADC #1:STA
(cell),Y
1780 JSR display
1790 LDA #(H%-1):STA col:LDA #(V%-1):ST
A row:JSR calccellno
1800 LDY#0:CLC:LDA (cell),Y:ADC #4:STA
(cell),Y
1810 JMP display
1820 .notend
1830 JMP main
1840 :
1850 .random
1860 LDA rnd:AND #648:ADC #638
1870 ASL A:ASL A
1880 ROL rnd+2:ROL rnd+1:ROL rnd
1890 LDA rnd
1900 CMP lim1:BCC random:CMP lim2:BCS r
andom
1910 RTS
1920 :
1930 .check
1940 LDY #0:LDX #0:STX index
1950 LDA row:CMP #1:BCC notsol
1960 SEC:LDA cell:SBC #H%:STA tcell
1970 LDA cell+1:SBC #0:STA tcell+1

```

```

1980 LDA (tcell),Y:BNE notsol
1990 INX:LDA #0:STA tq,X
2000 .notsol
2010 LDA col:CMP #(H%-1):BCS notso2
2020 CLC:LDA cell:ADC #1:STA tcell
2030 LDA cell+1:ADC #0:STA tcell+1
2040 LDA (tcell),Y:BNE notso2
2050 INX:LDA #1:STA tq,X
2060 .notso2
2070 LDA row:CMP #(V%-1):BCS notso3
2080 CLC:LDA cell:ADC #H%:STA tcell
2090 LDA cell+1:ADC #0:STA tcell+1
2100 LDA (tcell),Y:BNE notso3
2110 INX:LDA #2:STA tq,X
2120 .notso3
2130 LDA col:CMP #1:BCC notso4
2140 SEC:LDA cell:SBC #1:STA tcell
2150 LDA cell+1:SBC #0:STA tcell+1
2160 LDA (tcell),Y:BNE notso4
2170 INX:LDA #3:STA tq,X
2180 .notso4
2190 STX index:TXA:BEQ nextcell
2200 RTS
2210 :
2220 .nextcell
2230 INC col:LDA col:CMP #H%:BNE notedg
e
2240 LDA #0:STA col
2250 INC row:LDA row:CMP #V%:BNE notedg
e
2260 LDA #0:STA row
2270 .notedge
2280 JSR calccellno
2290 LDY #0:LDA (cell),Y:BEQ nextcell
2300 JMP check
2310 :
2320 .calccellno
2330 LDA row:STA tempa
2340 LDA #0:STA cell
2350 LDX #8
2360 .multbyH
2370 LSR tempa:BCC skip
2380 CLC:ADC #H%
2390 .skip
2400 ROR A:ROR cell
2410 DEX:BNE multbyH
2420 STA cell+1

```

Minotaur's Maze

```
2430 CLC:LDA cell:ADC col:STA cell
2440 BCC addmazeaddr:INC cell+1
2450 .addmazeaddr
2460 CLC:LDA #maze MOD 256:ADC cell:STA
cell
2470 LDA #maze DIV 256:ADC cell+1:STA c
ell+1
2480 RTS
2490 :
2500 .display
2510 PHP:LDA #31:JSR oswrch
2520 LDA col:ADC #R%:JSR oswrch
2530 LDA row:ADC #1:JSR oswrch
2540 LDY #0:CLC:LDA (cell),Y:ADC #224:J
SR oswrch
2550 PLP:RTS
2560 :
2570 .tq EQU B0:EQU D 0
2580 .kk EQU B 0:EQU B 1:EQU B 0:EQU B &FF
2590 .wk EQU B &FF:EQU B 0:EQU B 1:EQU B 0
2600 .power EQU B 1:EQU B 2:EQU B 4:EQU B 8
2610 .maze EQU STRING$(255,CHR$0):EQU
STRING$(255,CHR$0):EQU STRING$(255,CHR
$0):EQU STRING$(255,CHR$0):EQU STRING$(
120,CHR$0)
2620 |:NEXT
2630 :
2640 osword=&FFF1:osbyte=&FFF4
2650 x=&70:z=&72:y=&80
2660 tint=&74:count=&75:octet=&76
2670 :
2680 FOR pass=0 TO 2 STEP 2
2690 P%=&900
2700 [OPT pass
2710 LDA #26:JSR oswrch
2720 LDA #3:LDX #10:JSR osbyte
2730 .clear JSR next
2740 .line JSR esc
2750 LDA #65:JSR oswrch
2760 LDA #8:JSR oswrch
2770 .load LDA #252:STA y
2780 LDA #3:STA y+1
2790 .newline LDA #0:STA x:STA x+1
2800 .bit JSR esc
2810 LDA #76:JSR oswrch
2820 LDA #192:JSR oswrch
2830 LDA #3:JSR oswrch
2840 .start LDA #4:STA count
2850 LDA y:STA z
2860 LDA y+1:STA z+1
2870 .readpal LDX #&70:LDY #0:LDA #9
2880 JSR osword
2890 LDA tint:AND #7:TAY
2900 .byte LDX #0
2910 LDA pattern,Y
2920 .loop1 LSR A:ROL octet,X
2930 LSR A:ROL octet,X
2940 INX:CPX #3
2950 BNE loop1
2960 .loop2 DEC count
2970 BEQ print
2980 SEC:LDA z:SBC #4:STA z
2990 LDA z+1:SBC #0:STA z+1
3000 JMP readpal
3010 .print LDX #0
3020 .loop3
3030 LDA octet,X:JSR oswrch
3040 INX:CPX #3
3050 BNE loop3
3060 .xloop
3070 CLC:LDA x:ADC #4:STA x
3080 LDA x+1:ADC #0:STA x+1
3090 CMP #5:BNE start
3100 JSR next
3110 .yloop
3120 SEC:LDA y:SBC #16:STA y
3130 LDA y+1:SBC #0:STA y+1
3140 LDA y:CMP #(Q%MOD256):BNE cont
3150 LDA y+1:CMP #(Q%DIV256):BEQ end
3160 .cont
3170 JMP newline
3180 .end
3190 JSR esc
3200 LDA #64:JSR oswrch
3210 LDA #3:LDX #0:JMP osbyte
3220 .esc
3230 LDA #27:JMP oswrch
3240 .next
3250 LDA #10:JMP oswrch
3260 .pattern
3270 EQU D &FFFFFF00
3280 EQU D &FF000000
3290 |:NEXT
3300 ENDPROC
```

Putting Directories to Work (Part 2)

Michael Shepherd goes in search of the lost file.

Last month's program, *Contents*, presented the user with a display of the current directory which was then scanned in order that a further selection could be made. *Contents* assumes the user knows the full pathname of the file required. If the pathname isn't known, *Contents* is of less use, as the user then has to scan a number of directories to find the required file.

Routines like *EXALL display all the directories and files stored on an ADFS disc. It is thus possible to read through the list and find any file or directory. However, this takes time and is the type of task best left to the computer to carry out, given an appropriate routine.

The accompanying program, *Search*, will find any named ADFS file or directory (if the name is duplicated it will find one of the occurrences) and give the full pathname. If the target is a file it will then ask for the appropriate language and load it. If the target is a directory, it will make it the current directory. If the name of the target is duplicated and the file found is not the one required, it should be renamed and the search repeated.

The search starts in the root directory and traces the directory structure branch by branch. If the target is a directory, then each directory is checked against the target as it is located, otherwise the next sub-directory is sought. The outermost directory is found on each branch, then, if the target is a file, the

directories are searched for the target as the path is retraced to the root directory. This may sound the wrong way round, but the logic is simpler and 'lost' files are more likely to be in 'outlying' directories than those nearer home.

Searches are always carried out on the current directory, so that it is only a matter of copying it into user RAM, using the fourth byte of the entry to distinguish files from directories. The program will accommodate up to 23 levels of directories, and screen messages indicate the progress of the search.

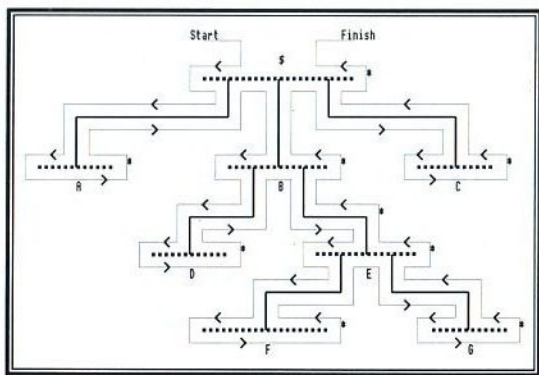


Figure 1. Hypothetical directory structure

The search technique is illustrated in Fig. 1, which shows a hypothetical directory structure containing directories \$ (the root directory) and A - G. In order to provide a full description, it is assumed that the search is for a directory which does not exist, thus ensuring a search of the complete disc. Table 1 lists the screen messages which would appear as the search proceeds. For clarity, the search has been restricted to that for a directory; if the target had been a file there would

Putting Directories to Work

have been a search for the file every time the search returned to a parent directory.

Searching directory \$ for Sub-directory
Searching directory \$.A for Sub-directory
Searching directory \$ for Sub-directory
Searching directory \$.B for Sub-directory
Searching directory \$.B.D for Sub-directory
Searching directory \$.B for Sub-directory
Searching directory \$.B.E for Sub-directory
Searching directory \$.B.E.G for Sub-directory
Searching directory \$.B.E for Sub-directory
Searching directory \$.B for Sub-directory
Searching directory \$ for Sub-directory
Searching directory \$.C for Sub-directory
Searching directory \$ for Sub-directory

Table 1. Scanning the directories of Fig. 1

Starting with the first entry in the root directory (\$), each entry is examined in turn until sub-directory A is found. At this point a number of things take place -

The sub-directory's name is read and added to the pathname (=\$.A).

If the target is a directory, the target name is checked against the new directory name. If it matches the search is ended.

The position of the next entry in the current directory is recorded.

The length of the new pathname is recorded (=3).

The sub-directory A is made the current directory and copied into user workspace.

The search for a sub-directory resumes at the first entry of the new current directory A.

This cycle is repeated until the search reaches the end of the directory without finding a further sub-directory. If the target had been a file, this current directory A would be searched again to see if it contains the target file (this is indicated in Fig. 1 by *). If not, (and this must always happen if the target is a

directory), the program returns to the previous directory \$ and resumes the interrupted search from where it was broken off, at the same time shortening the pathname appropriately. In this way it next finds directory B, which it enters and scans in exactly the same way to find directory D. This contains no sub-directory but is searched for the target before resuming the scan of B.

The further scan of B leads to directory E which contains directories F and G. In the scan of E, when F is encountered it is entered, scanned for further directories and then searched for the target before returning to E to continue the scan there. The same process is repeated when G is encountered before returning to E and thence to \$. Resumption of the scan of \$ finds directory C before the search eventually finishes when the end of directory \$ is reached.

In reality, it is to be expected that the target entry would be found, assuming the user knew on which disc the entry had been made. However, if the search was unsuccessful the disc can be changed and the search continued from the prompt.

If the search is successful then the target, if a file, can be loaded into the appropriate language. If a directory, it is already the current directory and immediately accessible.

THE PATHNAME

In the first article, and again here, reference is made to alterations to the pathname. As pathnames are a vital feature of the ADFS, it is worthwhile explaining how the pathname is recorded in the programs.

Apart from the root directory, which is always named '\$', an ADFS entry name

may be up to 10 characters long. As directory names in a pathname must be separated by '.', a directory can occupy up to 11 characters in a pathname. Thus a pathname length of 255 characters (the maximum length of a BASIC string) can hold at least 24 levels of directory. This is composed of -

$$\begin{aligned} & \$ \cdot + 23 \text{ units of } (' \text{directory name}' + \cdot) = \\ & 2 + 23 \times 11 \end{aligned}$$

Using the variable names assigned in the programs, a directory system can be visualised as shown in Table 2.

level%	pathname\$	pathlength%(level%,0)
0	\$	1
1	\$.FIRST	7
2	\$.FIRST.SECOND	14
3	\$.FIRST.SECOND.THIRD	20

Table 2. Recording pathnames

It can be seen that, as the level of directory increases (i.e. the further the current directory is away from the root directory), the pathname has to be incremented by adding '.' and the new directory name to it. More concisely:

$$\text{pathname at (level\%+1) = pathname at (level\%) + '\cdot' + (directory name)}$$

Moving closer to the root directory shortens the pathname as a result of removing the old directory name from the pathname:

$$\text{pathname at (level\%-1) = pathname at level\% - '\cdot' - (directory name)}$$

The simplest way of accomplishing this curtailment is by use of the statement:

$$\text{pathname at (level\%) = LEFT$(pathname$,pathlength%(level\%,0))}$$

A little thought will show that this statement is always true and will, if

required, accommodate decrements of level% which are greater than 1.

The array used in the program Contents, pathlength%(level%), is in fact one-dimensional. In the program Search a two-dimensional array is used; the elements pathlength%(level%,1) contain the points at which the scan of the directory is to be resumed on return from the higher level.

These programs should be of great help when using ADFS discs with complex structures, I hope it will also throw some light on the way ADFS works.

```

10 REM Program Search
20 REM Version B1.0
30 REM Author J M Shepherd
40 REM BEEBUG Jan/Feb 1994
50 REM Subject to copyright
60 :
100 ON ERROR GOTO 1960
110 MODE 0:OSCLI"DIR $":
120 pathname$=STRING$(255, " "):pathnam
e$=""
130 DIM pathlength%(22,1):pathlength%(
0,0)=1
140 FOR level%=0 TO 22:pathlength%(lev
el%,1)=%&2000:NEXT:level%=0
150 :
160 PROCtarget:found%=FALSE:dirend%=FA
LSE
170 H%=2:PROChugo(H%)
180 REPEAT
190 PROCnext_dir:IF type%=0 PROCsearch
_dir
200 IF found% PROCsuccess ELSE IF leve
l%>0 PROCbackwards ELSE dirend%=TRUE
210 UNTIL found% OR dirend%
220 VDU3:IF found% PROClanguage ELSE P
ROCfailure
230 IF accepted% RUN ELSE CLS:END
240 :
1000 DEF PROCyesno
1010 PRINTTAB(20,28);"Please confirm (Y

```

Putting Directories to Work

```
(N) ";
1020 REPEAT:G=((GET MOD 32)-14)/11:UNTI
L G=INT(G):accepted%=-G
1030 PRINTTAB(20,28);SPC(22)
1040 ENDPROC
1050 :
1060 DEF PROCtarget
1070 PRINTTAB(20,18);"Give file/directo
ry name required ";
1080 REPEAT
1090 REPEAT
1100 INPUTTAB(56,18)input$
1110 UNTIL LENinput$>0 AND LENinput$<11
1120 target$="":FOR I=1 TO LENinput$:A=
ASC MID$(input$,I,1):IF A>95 A=A-32
1130 target$=target$+CHR$(A):NEXT
1140 PRINTTAB(20,20);"Is this a directo
ry or a file (D/F) ? ";
1150 REPEAT:type%=1-((GET-4) MOD 32)/2:
UNTIL type%=1 OR type%=0
1160 PRINTTAB(20,26)"Target is ";MID$("
directoryfile",10-type%*9,9);" ";target$
1170 PROCyesno:UNTIL accepted%:
1180 PRINTTAB(20,26)"Is a print-out nee
ded?";SPC(15);:PROCyesno:CLS:IF accepte
d%VDU2
1190 ENDPROC
1200 :
1210 DEF PROCshow_dir
1220 PRINTTAB(10);"Searching directory
";pathname$;" for ";
1230 IF search% PRINT "target ";target$
ELSE PRINT "sub-directory"
1240 ENDPROC
1250 :
1260 DEF PROCsearch_dir
1270 search%=TRUE:E%=&2000:PROCshow_dir
:found%=FALSE
1280 REPEAT
1290 IF E%?3 DIV &80=type% PROCstandard
ise(E%):IF modified$=target$ found%=TRUE
1300 IF found%=FALSE E%=E%+&1A
1310 UNTIL found% OR ?E%=0
1320 ENDPROC
1330 :
1340 DEF PROCnext_dir
```

```
1350 REPEAT
1360 E%=pathlength%(level%,1):search%=F
ALSE:PROCshow_dir
1370 REPEAT
1380 modified$="":IF ?E%>0 AND (E%?3 DI
V &80)=1 PROCstandardise(E%) ELSE E%=E%+
&1A
1390 UNTIL LENmodified$ OR ?E%=0
1400 IF modified$=target$ found%=TRUE E
LSE IF ?E% PROCforwards
1410 UNTIL (?E%=0 AND modified$="") OR
found%
1420 ENDPROC
1430 :
1440 DEF PROCforwards
1450 pathname$=pathname$+"."+modified$:
pathlength%(level%,1)=E%+&1A
1460 level%=level%+1:pathlength%(level%
,0)=LENpathname$
1470 H%=(E%&16 AND &FFFFFF):PROCchugo(H%
):OSCLI"DIR "+pathname$
1480 ENDPROC
1490 :
1500 DEF PROCbackwards
1510 level%=level%-1:E%=pathlength%(lev
el%,1)+&1A
1520 pathname$=LEFT$(pathname$,pathleng
th%(level%,0)):OSCLI"DIR"+pathname$
1530 H%=!&24D1 AND &FFFFFF:PROCchugo(H%)
:OSCLI"DIR "+pathname$
1540 ENDPROC
1550 :
1560 DEF PROCchugo(H%)
1570 A%=&72:X%=&70:Y%=0:?X%=0:X%!1=&1FF
B:X%?5=8
1580 X%?6=(H% AND &FF0000)/&10000
1590 X%?7=(H% AND &FF00)/&100
1600 X%?8=(H% AND &FF)
1610 X%?9=5:X%?10=0:X%!11=0
1620 CALL &FFF1
1630 ENDPROC
1640 :
1650 DEF PROCstandardise(X)
1660 modified$="":char%=0
1670 REPEAT
1680 A=? (X+char%) MOD &80:IF A>95 A=A-3
2
```

Continued on page 20



File Handling (Part 2)

Marshal Anderson delves further into data files.

Last month we looked at a simple way of saving data to a disc using PRINT# and INPUT#. The example dealt with a whole file at once, which limits the size of a file to the amount of memory available in the machine. So what do you do if you want to just read or write one part of a file?

Basic provides PTR#<file handle> to help with this. It represents the position of the *pointer* in the file, that is the place Basic will read or write from. It follows, then, that if you know where the information you want is stored within a file you can read or write just that part. This can be done by making the length of each chunk the same, a method that's used in this month's example.

To get full control of your file you need to deal with it a byte at a time. This will allow you to read and write data of variable lengths once you have understood the basic principle. So, although the task in the example could be done with PRINT# and INPUT#, it is used to demonstrate the fundamentals of byte-at-a-time file handling.

Basic's BPUT# and BGET# allow you to deal with individual bytes on the disc and EXT# tells you how long a file is. Because you can tell the pointer where to point to as well as find out where it is currently pointing we have some very powerful tools for file manipulation.

There is, of course, a down side to this flexibility. With PRINT# and INPUT# Basic takes care of the starts, length, and ends of the items within a file. Using a byte at a time requires *you* to manage

this which means the extensive use of loops, counters and markers and some fairly clever string handling.

The best way to illustrate these tools is to take the task we set last month and use these new methods to give us a more flexible system. The program *Betterfiles* does this and I will take you through it step by step to show you what's going on. A little warning note here; we dealt last month with the problems caused by files being left open; this will often happen if there are errors in your program or you use the Escape key to leave it. Before you begin, type *KEY 0 CLOSE#X|M. Then, any time the program stops with an error you can press function key 0 to make sure the file is closed.

There are two ways into the program; start a new file or use an existing file. If you start a new file *PROCcreatefile* creates a blank file on disc. Otherwise an existing file is opened.

Either way, the program moves on to *PROCread_add_edit* which covers the main operations. This starts by opening the file in question with OPENUP. When a file is opened the pointer is at the beginning, pointing at byte 0. Line 1440 checks to see if the pointer and the length of the file are the same - if this is true then the pointer is pointing at the end of the file and it must be empty so the program calls *PROCaddrrecords*.

This procedure first calls *PROCgetinfo* which asks you to type in the information for *just one* employee. The data goes into the variables *NAME\$, AGE, WAGE* and *ID\$*. Now we call

First Course

PROCwriterecord and this is where you will need to start concentrating.

Look at line 1690. The four variables are assembled into a long string ready for writing, we'll call this a record. The numeric values are converted to text; this is a good idea as it allows you to deal with any number, not just the 255 limit in a byte. Basic provides easy ways of converting between textual and numeric values. A '#' is placed between each bit of text so that we can separate them again when we read them in. This marker is purely arbitrary; you just need a character that won't be used in the record.

To keep things simple I have decided to use 100 byte blocks for each record, this makes moving the pointer easy as we know that records start at byte 0, 100, 200 etc. Every time we put a byte to, or get a byte from the file, it is written or read at the pointer, which then moves to the next byte. To make sure the pointer moves on to the next whole hundred we must make our record 100 characters long. Line 1700 does this, padding the string out with hashes. I've done it this way, rather than just move the pointer on, to make sure that no random garbage can confuse things. It's a bit of overkill here but it's a good principle.

The loop starting at 1710 uses *BPUT#* to write the bytes to the disc. The form for *BPUT#* is:

```
BPUT#<filehandle>,<byte to be saved>
```

The byte in this case is the ASCII value of each character in the string. Each time a byte is written the pointer moves on so, at the end of the loop, it will have moved on exactly 100. Once the record is written the program passes back to *PROCwriterecord* where the option is offered to write another.

Now go back to *PROCread_add_edit* and assume we have used an old file. The program enters a loop which first calls for a record to be read from the file, it assumes that the pointer is at the start of a record - which it should be!

PROCreadfromfile uses *FNonelump* to fill up the variables from the disc. *FNonelump* reads along the file, adding the characters represented by the bytes to *A\$*, until it finds a '#'. The data is read by *BGET#<file handle>* which you read as if it were a numeric variable. Once *A\$* is finished, the hash is stripped off the end and the result passed back to *PROCreadfromfile*. With the variables *AGE* and *WAGE* the string is converted into a number using *VAL*.

You can see it is vital that data is written and read in the same order, otherwise you will lose track of everything. Once the variables are full they are printed out by *PROCprintinfo*.

An option is given to modify the record or continue, let's look at continuation first. After reading the last record the pointer will have been left somewhere *before* the start of the next record, the next whole hundred - we need to get it to the start of the next record. Line 1510 uses a little maths to do this; divide by 100, use *INT* to knock off the decimal points, add 1, multiply by 100. You can see that *PTR#<file handle>* is dealt with just like a variable.

PROCmodify allows part of the file to be altered. First it uses *PROCgetinfo* to get new values for *NAME\$, AGE, WAGE, and ID\$*. We need to write over the part of our file that contained the original record. To do this we must move the pointer back to the start of the record. Line 1790 uses the same maths as before but doesn't add 1 before multiplying back up. Having done

this, *PROCwriterecord* puts the new information on the disc, the extra hashes making sure that the old information is completely overwritten.

POSSIBLE ERRORS

The errors described last month may still come up for the same reasons. A new one will be the 'Can't extend' error. You will get this if you create and alter a lot of files in DFS. DFS fits files into any space it can find, or starts them immediately after the last one. This can mean that your file can come up against the start of the next one on the disc if you try to add records. While experimenting with this program use a fresh disc and don't delete files.

You can see the flexibility of using this byte-at-a-time approach. Take some time getting used to the pointer then, instead of using fixed lengths for records, try using markers for the starts and ends. Using EXT# the way I have is a bit of a short cut and stops you being able to delete records as the program needs to find the end of the file. Try using *DUMP to see how others have done it, then see if you can do it better.

```

10 REM Program BetterFiles:
20 REM Version B1.0
30 REM Author  Marshal Anderson
40 REM BEEBUG  Jan/Feb 1994
50 REM Program subject to copyright
60 :
100 MODE 3
110 REPEAT
120 A=FNmenu
130 IF A=1 PROCcreatefile:PROCread_add
_edit
140 IF A=2 PROCgetfilename:PROCread_ad
d_edit
150 UNTIL A>2
160 END
170 :
1000 DEF FNmenu

```

```

1010 PRINT "1) Create new file""2) Use
existing file""3) Finish"
1020 INPUT Y
1030 =Y
1040 :
1050 DEF PROCcreatefile
1060 INPUT"Name of file to create";FILE
$
1070 X=OPENOUT FILE$
1080 CLOSE#X
1090 PRINT FILE$;" created"
1100 ENDPROC
1110 :
1120 DEF PROCgetfilename
1130 INPUT"What is the name of your fil
e";FILE$
1140 ENDPROC
1150 :
1160 DEF PROCreadfromfile
1170 NAME$=FNonelump
1180 AGE=VAL(FNonelump)
1190 WAGE=VAL(FNonelump)
1200 ID$=FNonelump
1210 ENDPROC
1220 :
1230 DEF FNonelump
1240 A$=""
1250 REPEAT
1260 A$=A$+CHR$(BGET#X)
1270 UNTIL RIGHT$(A$,1)="#"
1280 =LEFT$(A$,LEN(A$)-1)
1290 :
1300 DEF PROCgetinfo
1310 PRINT""You are going to enter deta
ils for 1 employee.""You need name, age
, wage and ID code."
1320 INPUT "Name ";NAME$
1330 INPUT "Age ";AGE
1340 INPUT "Wage ";WAGE
1350 INPUT "ID code";ID$
1360 ENDPROC
1370 :
1380 DEF PROCprintinfo
1390 PRINT NAME$,AGE," ";WAGE,ID$
1400 ENDPROC
1410 :
1420 DEF PROCread_add_edit
1430 X=OPENUP FILE$

```

First Course

```
1440 IF PTR#X=EXT#X PROCaddrecords:CLOS
E#X:ENDPROC
1450 REPEAT
1460 PROCreadfromfile
1470 PROCprintinfo
1480 PRINT"'C'ontinue OR 'M'odify"
1490 A$=GET$
1500 IF A$="M" PROCmodify
1510 IF A$="C" THEN PTR#X=((INT(PTR#X/1
00))+1)*100
1520 UNTIL PTR#X=EXT#X
1530 PRINT"Add more records? (Y/N) "
1540 A$=GET$
1550 IF A$="Y" PROCaddrecords
1560 CLOSE#X
1570 ENDPROC
1580 :
1590 DEF PROCaddrecords
1600 REPEAT
1610 PROCgetinfo
1620 PROCwriterecord
```

```
1630 PRINT"Another? (Y/N) "
1640 A$=GET$
1650 UNTIL A$="N"
1660 ENDPROC
1670 :
1680 DEF PROCwriterecord
1690 record$=NAME$+"#"+STR$(AGE)+"#"+S
TR$(WAGE)+"#"+ID$
1700 record$=record$+STRING$(100-LEN(re
cord$),"#")
1710 FOR Y%=1 TO 100
1720 BPUT#X,ASC(MID$(record$,Y%,1))
1730 NEXT
1740 ENDPROC
1750 :
1760 DEF PROCmodify
1770 PRINT"Type in new info"
1780 PROCgetinfo
1790 PTR#X=(INT(PTR#X/100))*100
1800 PROCwriterecord
1810 ENDPROC
```

Putting Directories to Work (continued from page 16)

```
1690 IF A=13 ELSE modified$=modified$+C
HR$(A):char%=char%+1
1700 UNTIL A=13 OR char%=10
1710 ENDPROC
1720 :
1730 DEF PROCsuccess
1740 PRINT'TAB(10);"Entry ";CHR$34;targ
et$;CHR$34;" has been found."
1750 PRINT'TAB(10);"The pathname is: ";
pathname$+"."+target$:found%=TRUE
1760 REPEAT:PRINT'TAB(10);"RETURN to co
ntinue";:UNTIL GET=13
1770 ENDPROC
1780 :
1790 DEF PROCfailure
1800 PRINT'TAB(10);"Entry ";CHR$34;targ
et$;CHR$34;" not found."
1810 PRINTTAB(8,28);"Try again ? ";:PRO
Cyesno
1820 ENDPROC
1830 :
1840 DEF PROClanguage
1850 READ data$:CLS:PRINTTAB(10,16);:OS
```

```
CLI"INFO "+target$
1860 PRINTTAB(20,18);"Facilities availa
ble"+STRING$(30," ")
1870 FOR line%=1 TO 6
1880 PRINTTAB(25,20+line%);STR$(line%);
". "+MID$(data$,line%*5-4,5):NEXT
1890 PRINTTAB(30,28);"Enter choice (1 -
6)";
1900 REPEAT:lang%=GET-49:UNTIL lang%>-1
AND lang%<6
1910 lang%=MID$(data$,lang%*5+1,5)
1920 IF type%=1 OSCLI"KEY5 OSCLI"DIR "+
target$ "|M*CAT|M"
1930 IF type%=0 OSCLI"KEY5 LOAD"+CHR$34
+target$+CHR$34+"|M*CAT|M"
1940 OSCLI"FX138,0,133":OSCLI lang$
1950 :
1960 REM error handling
1970 REPORT:PRINT" at line ";ERL:STOP
1980 :
1990 DATA"BASICSHEETSTOREWORD EDIT TERM
"
```

Converting Newline Codes in Text

Colin Robertson sorts out some text transfer problems.

Text (ASCII) files are useful for transferring data between different applications and different machines. These days, BBC owners may well find themselves needing to transfer data to and from PCs. It is assumed here that they have the means to read and write discs in the appropriate formats, or to transfer data via the serial ports.

BBC micros and the PCs differ in the coding and the method they use to generate new lines in text files. BBC machines frequently work with files containing carriage returns (CR) but no linefeeds (LF). When such a file is sent to screen, printer or serial port through the BBC's output routine OSASCI, each CR is detected and sent to OSNEWL, which causes a LF to precede the CR. Thus the normal newline coding in a BBC micro is LF/CR.

PCs expect files to contain both carriage returns and linefeeds. Moreover, the normal coding for a newline is a carriage return followed by a linefeed (CR/LF). This is a reason why text files exported from a BBC micro to a PC can fail to be properly recognised by the PC, although this does not necessarily apply to all PC software. Likewise, a DOS text file may not be fully acceptable to BBC software. For example, Wordwise Plus treats all linefeeds as 'illegal' characters and replaces them with the 'double bar'.

The *Convert* program listed below assembles a machine code utility enabling the coding of newlines in text files to be output in a choice of styles when the file is typed (i.e. using *TYPE)

through the output system. It works by redirecting the OSWRCH vector at &20E, so that output can be processed by external routines before being sent on its way to the original address. Type in the assembly code carefully, making sure to save it before attempting to run it. When you run it, the object code will be assembled to &900 and information will be displayed about three available CALL addresses (*nolfs*, *doslfs* and *normal*). Make a note of these addresses. Also, you should accept the prompted option to save the object code under the filename *Feeder*.

OPERATION

In *Convert*, there is a 'toggle' feature that governs the redirection and restoration of the vector. As assembled, a NOP code is placed at *.start* and an RTS code at *.normal*. Initially the code has not been executed, so the vector has not been redirected and *.oldvec* does not contain a valid address (it contains zero). If *.normal* is called in this condition, nothing happens because the first instruction encountered is RTS. A call to either *.nolfs* or *.doslfs* executes a jump to *.start*, and if the instruction *.start* is NOP, the vector will be redirected, the old address being stored at *oldvec*. However, the redirection routine also writes an RTS at *.start* and an NOP at *.normal*. This prevents a repeated attempt to redirect the vector while it is already redirected (which would cause the original address to be 'lost'), but it enables the vector to be restored by a call to *.normal*. The restoration routine interchanges the two instructions again. The reason for including this mechanism is to allow

Converting Newline Codes in Text

freedom in the sequence of using the calls while permitting the code to be completely disengaged when not in use.

It will be seen that a call to *.doslfs* causes the linefeed code &0A to be stored at *.char*. Each call to OSWRCH is redirected to *.newvec*. If the output character is a LF it is suppressed. If it is a CR (&0D) it is sent on, but is then followed by the contents of *.char*, which is a LF. Thus the output text contains a LF after each CR, but no other LFs, whatever the input might have contained.

The other option is to call *.nolfs*. This works in a similar way to *.doslfs*, except that the character written at *.char* is not a LF (it is actually zero). In this case, any LFs in the stream are removed as before, but none are added. The result is text with all LFs removed.

It should be noted that the Feeder routine determines the output style of newlines absolutely, i.e. it does not depend on the style used in the input stream. So, no harm is done if you attempt to process a text file that already has the desired style. This should make it useful for both importing and exporting text files even if their initial newline style is not known.

Although most users will find a load address of &900 to be satisfactory, the code can be assembled to any other suitable address in RAM, by assigning a different value to *load%* at the start of the program. However, the object code is not relocatable and must be re-assembled if the loading address is to be changed.

The object code can be loaded with *RUN Feeder or, from the current directory or the current library directory, with the abbreviation *Feeder. It starts with an RTS to prevent automatic execution at the time of loading.

A DEMONSTRATION

Demo is a simple Basic demonstration program that types an input text file and spools it to an output file three times; the first with DOS-type newlines (CR/LF), the second with no LFs at all and the third in the native BBC newlines (LF/CR). You can use any text file by renaming it INFILE or altering line 100 to carry the filename. Afterwards, you can verify the contents of any of the files with *DUMP.

A point to mention about the Demo program is the inclusion of lines 200 and 240, which turn the screen off temporarily while using *TYPE in the 'nolfs' mode. This is because the screen needs LFs to generate its own newlines and when they are absent the display simply over-writes line after line without scrolling. In some circumstances, serious disruption can arise from allowing output to reach the screen while the input file is being processed. The device of using *FX3,6 to disable the screen can be used wherever required.

On the Master series, depending on the actual content of the input file, it may be better to use *PRINT rather than *TYPE. *TYPE converts non-printing characters into GSREAD format and, as a consequence of this, prints the double-bar 'control' character | as ||. *PRINT, on the other hand, sends all the codes in the file as they are. In the 'normal' condition, no LFs will be added if *PRINT is used, but the *doslfs* and *nolfs* options of Feeder will work as described.

The principles of the Demo program can easily be employed in programs written in other languages, especially as the statements consist predominantly of star commands. The author has used this utility as part of a Wordwise Plus segment program that can modify the

Converting Newline Codes in Text

formatting features selectively in text files intended for transfer to other systems.

```
10 REM Program Convert
20 REM Version B1.0
30 REM Author C W Robertson
40 REM BEEBUG Jan/Feb 1994
50 REM Program subject to copyright
60 :
100 MODE 7
110 load%=%900
120 wrchv=%20E
130 FOR pass=0 TO 3 STEP3
140 P%=load%
150 [OPT pass
160 RTS
170 .nolfs
180 LDA #0:STA char:JMP start
190 .doslfs
200 LDA #&0A:STA char
210 .start
220 NOP:LDA wrchv
230 STA oldvec:LDA wrchv+1
240 STA oldvec+1:LDA #newvec MOD 256
250 STA wrchv:LDA #newvec DIV 256
260 STA wrchv+1:LDA #&60
270 STA start:LDA #&EA
280 STA normal:RTS
290 .normal
300 RTS:LDA oldvec
310 STA wrchv:LDA oldvec+1
320 STA wrchv+1:LDA #&60
330 STA normal:LDA #&EA
340 STA start:RTS
350 .newvec
360 CMP #&0A:BEQ suppress
370 CMP #&0D:BEQ addlf
380 JMP send:RTS
390 .addlf
400 JSR send:LDA char
410 CMP #&0A:BNE suppress
420 JMP send
430 .suppress
440 RTS
450 .send
460 JMP (oldvec)
470 .oldvec
480 EQUW 0:RTS
490 .char
```

```
500 EQUW &0A:RTS
510 ]
520 end%=P%
530 NEXT
540 PRINT"Load Address: &;~-load%
550 PRINT"End address: &;~-end%
560 PRINT"For no LFs CALL &;~nolfs
570 PRINT"For DOS LFs CALL &;~doslfs
580 PRINT"To reset to normal CALL &;
~normal
590 PRINT"Save as 'Feeder' (Y/N)?"
600 REPEAT:A$=GET$
610 pos=INSTR("YyNn",A$):UNTIL pos>0
620 cmd$="SAVE Feeder "+STR$~load%+"
"+STR$~end%
630 msg$="Not saved!"
640 IF pos <3 THEN OSCLI cmd$:msg$="S
aved!"
650 PRINT msg$
```

```
10 REM Program Demo
20 REM Version B1.0
30 REM Author Colin Robertson
40 REM BEEBUG Jan/Feb 1994
50 REM Program subject to copyright
60 :
100 fname$="INFILE"
110 MODE 0:d$="TYPE "
120 *FEEDER
130 CALL &909
140 *SPOOL outCRLF
150 OSCLI d$+fname$
160 *SPOOL
170 CALL &901
180 CLS
190 PRINT"Screen disabled - please wa
it"
200 *FX3,6
210 *SPOOL outnoLF
220 OSCLI d$+fname$
230 *SPOOL
240 *FX3
250 CLS
260 CALL &930
270 *SPOOL outBEEB
280 OSCLI d$+fname$
290 *SPOOL
300 PRINT"Program finished":END B
```

Machine Code Corner

This month Toady gets interrupted (some hope! - ed.)

Happy New Year, toad fans, hope you've recovered from all the jollities. All humbug, say I - I spent the whole time at my Beeb, as ever, toiling away night and day to bring you even more dazzling displays of (That'll do, frog, Ed.)

I've had some more interesting questions from Arthur Adams. Arthur has written a screen-dump ROM, but he's not sure how to code the bit which starts the thing in response to a certain combination of key-presses - he'd like to use Ctrl - Shift - '@', as does one of the commercial screen-dump utilities.

The point is that the screen dump software has to sit there, not getting in the way, while the screen is being set up by whatever program; then when the dump is needed it must be ready to go, preferably in response to an unlikely combination of key-presses. Ctrl - Shift - '@' (all at once) seems to me to fit the bill. The '@' mustn't appear on the screen, else it would show on the printout.

There are several ways you can do this from any machine-code routine, and it's especially easy from a sideways ROM. For example, you could divert the interrupt vector into your ROM, and then read the keyboard at each interrupt (each key-press causes an interrupt), using OSBYTE &79 or some such. Or you could use the events system - event 2 is 'character entering buffer'.

Again, sideways ROMs on the Master benefit from service call &15, 'polling interrupt'. Although my final recommendation to Arthur is not going to be this option, I'd like to digress for a moment (Cloches de l'enfer! Not again! Ed.) and have a look at the polling

interrupt system, because it's very clever and very handy. Anybody who managed to read my thingy on sideways ROMs without falling asleep will know that the MOS issues 'service calls' to the ROMs for all kinds of reasons: call 4 means 'your turn to check out an as yet unrecognised star-command to be found at (&F2),Y'; call 9 means 'user has issued a *HELP call - please do your bit'. There are lots of others, warning ROMs of things like 'reset has occurred' (call &27), 'close all files' (&26). Call &15, 'polling interrupt', is just an alarm-call - when the polling system is activated, it just means 'hello, ROM, another 10 milliseconds have gone by'. Your code then does its thing, whatever it may be, and goes dormant again.

Now, the MOS needs to save unnecessary slowing-down of the computer by issuing these calls only when wanted. Such slowing may only be slight, but every little helps (or hinders? ed.). So, any ROM can turn polling on using OSBYTE 22, 'increment ROM polling semaphore', and it can issue OSBYTE 23 to signal that it no longer needs the polling. Notice that I carefully did not say that it can turn the polling off. There might well be one or more other ROMs active which still do want the polling; therefore, what OSBYTE 22 really does is to increase by one the contents of a system variable (it's at &0243, if you ever need it), which the MOS reads before issuing a polling interrupt; the interrupt is not issued if the variable holds zero, which is its default state. So, any ROM, by incrementing this number, can turn the polling on, but it won't be turned off until every ROM which has at any time incremented it has later decremented it

again. Clever stuff, eh? In fact, you can read or write this flag, either directly at $\&0243$ or by OSBYTE $\&B3$, but the 'legal' convention should normally be observed.

This would do Arthur's job, as 100 checks a second are easily enough to keep a reliable eye on the keyboard, and I hope you'll agree that it's a useful item in one's repertoire, but in this case there is a neater way. We'll divert the vector INSV at $\&022A/\&022B$, then every time a character-key is pressed the MOS will call our code with the ASC of the character held in A. If it's not '@' we ignore it by jumping to the address that was in the vector before we diverted it. If it is '@', we'll check Control and Shift; if they aren't both down we'll still ignore it, otherwise we'll turn off the cursor with VDU 23,1,0,0,0,0,0,0,0; do the printout and exit.

This method is demonstrated in this month's listing, although we use it to call a demo routine rather than type in a whole screen-dump. You can also save typing by tacking the real code onto one of the ROM headers published last year.

First let's look at diverting the vector INSV - lines 580 to 670. We can't just divert it straight to the address of the routine *.checkKeys* in our ROM; some other ROM will be paged in at the time you call for the screen dump, and the MOS would call that address in the other ROM, not ours: all you'd get would be a crash. What we do is divert INSV into one of a series of specially calculated addresses in page $\&FF$, and leave a note of our ROM number and the address of *.checkKeys* in an 'agreed' place in page $\&0D$. Then, when something passes through INSV into the page $\&FF$ address, the MOS looks up the details in page $\&0D$, pages in our ROM and calls *.checkKeys*. When our routine exits, the

MOS picks up where it had left off before the keypress.

How do we calculate the page $\&FF$ and page $\&0D$ addresses? First find the 'vector number' by subtracting $\&200$ from its address and dividing the answer by two. Multiply this number by three and add $\&FF00$ to get the one address, and add $\&D9F$ for the other. All this rigmarole really means is that the vectors are two bytes 'wide', whereas the extended vector addresses (both lots) are three bytes each, so that from a given start-address you find your vector by counting up in twos, then from the other two starting-points you count up in threes.

Our vector is at $\&022A$, so its 'vector number' is $\&15$: $\&022A - \&0200 = \&2A$; $\&2A/2 = \&15$ and $3 * \&15 = \&3F$. Adding $\&3F$ to $\&0D9F$ gives $\&0DDE$, which is where we place the low byte of the address of our routine. In the next two locations we place the high byte of the address and then the number of the slot our ROM is currently in (line 630), having read the latter from $\&F4$ (where the current ROM number always lives) in line 600.

Since $3 * \&15$ is $\&3F$, the page $\&FF$ address to which we divert INSV is $\&FF3F$: lines 660 to 670 write $\&FF$ and $\&3F$ into the two bytes of INSV, after lines 580 and 590 have safely tucked away its previous contents into two bytes at the end of our ROM. Note also that we disable interrupts immediately before changing the vector, in case an interrupt should occur and use the vector with one byte changed and the other not. Unlikely, unless you believe in Sod's law - but show me a programmer who doesn't. To minimise the time spent with interrupts off (as you always should), we load up X and Y with the necessary numbers before the SEI. As long as the actual changing of

Machine Code Corner

the vector is done last, there is no need to disable interrupts while setting up the page &0D addresses - they will be meaningless until the vector points into page &FF, drawing the MOS's attention to them.

All the above is to 'arm' our ROM, as it were, to switch it on in response to the star-command *DUM (for 'dump on'; lines 510 to 560); after this we balance the stack, claim the call (LDA #0) and exit (line 740). I have included a little confirmatory message as well, otherwise it might seem that nothing has actually happened after the star-command. In fact, once *DUM has been typed and the ROM activated, all further key-presses will be examined by the routine *.checkKeys*, looking for the '@' key. Now ASC '@' is &40, but we're looking for '@' with Shift and Ctrl pressed as well. Holding down Ctrl subtracts &40 from the ASCII codes of the alphabetic keys, that's why Ctrl - B sends a 2 to the VDU drivers, rather than &42, and so turns the printer on.

So, we're actually looking for a zero (&40 - &40) in the accumulator at the beginning of our routine, line 820. If it's not zero we just jump to the address originally held in INSV and now stored at the end of the ROM. If it is zero, we test a system variable at &025A (line 840). If Ctrl is down, bit 6 of &025A will be set (= &40), and if Shift is down, bit 3 will be set (=8). So, mask off all but those two bits - AND #&48 - and check that both are down - CMP #&48. If this test is passed we kill the cursor, send our demo message, and - as if this were the real printer dump - restore INSV to 'disarm' the ROM. So, when testing, don't forget to do *DUM each time, or it won't work.

If it is a printer dump which you add on at line 900, you'll want to leave lines 860-

890 in place to send the *killCursorBytes* (lines 1070-1090) to the VDU drivers, omitting just the silly message. If not, omit the whole brouhaha. There must be other uses for a routine which can be called from places where a star-command can't be sent, using key-presses which are not echoed to the screen - from the text screen of View, for example. And if you don't want the thing deactivated after a single call, leave out all of lines 920-960, substituting merely: 920 PHA:JMP (oldVec).

This month's competition - we can cut down by a few bytes on the testing of &025A. How? Next month: how a sooty flue on a BBC B could put British Rail out of action for a week (the wrong sort of coal?).

```
10 REM Program Ctrl-Shift-@
20 REM Version B1.0
30 REM Author Mr Toad
40 REM BEEBUG Jan/Feb 1994
50 REM Program subject to copyright
60 :
100 index=&BE:indexHi=&BF
110 insVlo=&022A:insVhi=&022B
120 osnewl=&PF7:osasci=&PF3
130 osbyte=&FFF4
140 :
150 FOR N%=4 TO 6 STEP 2
160 Z%=?2+&100*?3-&200*(N%=4)
170 P%=&8000:O%=Z%
180 [ OPT N%
190 :
200 BRK:BRK:BRK
210 JMP start
220 EQU &82:EQU offset MOD &100
230 EQU &94
240 .title
250 EQU "Arthur's ROM":BRK:EQU &1B
260 .offset
270 BRK:EQU "(C) Beebug Jan 1994"
280 :
290 .print
300 STX index:STY indexHi:LDY #&FF
```

```

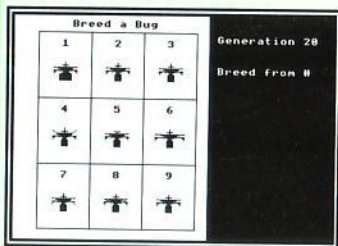
310 .loop
320 INY:LDA (index),Y
330 JSR osasci
340 CMP #&1B:BNE loop
350 RTS
360 :
370 .help
380 JSR osnewl
390 LDX #title MOD &100
400 LDY #title DIV &100
410 JSR print
420 JSR osnewl
430 :
440 .out
450 PLY:PLX:PLA:RTS
460 :
470 .start
480 PHA:PHX:PHY
490 CMP #9:BEQ help
500 CMP #4:BNE out
510 LDA (&F2),Y:AND #&DF
520 CMP #ASC"D":BNE out
530 INY:LDA (&F2),Y:AND #&DF
540 CMP #ASC"U":BNE out
550 INY:LDA (&F2),Y:AND #&DF
560 CMP #ASC"M":BNE out
570 :
580 LDA insVhi:STA oldVecHi
590 LDA insVlo:STA oldVec
600 LDA &F4
610 LDY #checkKeys DIV &100
620 LDX #checkKeys MOD &100
630 STA &0DE0:STY &0DDF:STX &0DDE
640 LDX #&3F:LDY #&FF
650 SEI
660 STY insVhi
670 STX insVlo:CLI
680 :
690 LDX #on MOD &100
700 LDY #on DIV &100
710 JSR print
720 :
730 .claimOut
740 PLY:PLX:PLA:LDA #0:RTS
750 :::::::::::::::
760 .exit
770 PLA
780 .notAt

```

```

790 JMP (oldVec)
800 :
810 .checkKeys
820 CMP #0:BNE notAt
830 PHA
840 LDA &025A:AND #&48
850 CMP #&48:BNE exit
860 LDX #message MOD &100
870 LDY #message DIV &100
880 JSR print
890 :
900 .yourCodeGoesInHere
910 :
920 LDX oldVec:LDY oldVecHi
930 STZ &0DE0:STZ &0DDF:STZ &0DDE
940 SEI
950 STX insVlo:STY insVhi
960 CLI:PLA:JMP (insVlo)
970 :
980 .on
990 EQUW &0D0D
1000 EQUW "ROM is on"
1010 EQUW &1B0D
1020 :
1030 .message
1040 EQUW &0D0D
1050 EQUW "'Allo, sailor! 'Oos a cheeky
boy, then?"
1060 .killCursorBytes
1070 EQUW &0117
1080 EQUW 0:EQUW 0
1090 EQUW &1B
1100 :
1110 .oldVec
1120 BRK
1130 .oldVecHi
1140 :
1150 ] NEXT
1160 FOR N%=7 TO 4 STEP-1
1170 IF N%?&2A1 NEXT:PRINT"Sorry - no
free SRAM slot.":END
1180 OSCLI "SRW. "+STR$-Z%+" "+STR$-(O%
+1)+ " 8000 "+STR$ N%
1190 N%?&2A1=&82
1200 PRINT"Running in slot ";N%
1210 N%=4:NEXT:END
1220 Floreat programmer,
1230 sit Beebug semper in flore.

```



PERSONALISED ADDRESS BOOK - on-screen address and phone book
PAGE DESIGNER - a page-making package for Epson compatible printers
WORLD BY NIGHT AND DAY - a display of the world showing night and day for any time and date of the year

Applications I Disc

BUSINESS GRAPHICS - for producing graphs, charts and diagrams
VIDEO CATALOGUER - catalogue and print labels for your video cassettes
PHONE BOOK - an on-screen telephone book which can be easily edited and updated
PERSONALISED LETTER-HEADINGS - design a stylish logo for your letter heads
APPOINTMENTS DIARY - a computerised appointments diary
MAPPING THE BRITISH ISLES - draw a map of the British Isles at any size
SELECTIVE BREEDING - a superb graphical display of selective breeding of insects
THE EARTH FROM SPACE - draw a picture of the Earth as seen from any point in space

File Handling for All

on the BBC Micro and Acorn Archimedes

by David Spencer and Mike Williams

Computers are often used for file handling applications yet this is a subject which computer users find difficult when it comes to developing their own programs. *File Handling for All* aims to change that by providing an extensive and comprehensive introduction to the writing of file handling programs with particular reference to Basic.

File Handling for All, written by highly experienced authors and programmers David Spencer and Mike Williams, offers 144 pages of text supported by many useful program listings. It is aimed at Basic programmers, beginners and advanced users, and anybody interested in File Handling and Databases on the Beeb and the Arc. However, all the file handling concepts discussed are relevant to most computer systems, making this a suitable introduction to file handling for all.

The book starts with an introduction to the basic principles of file handling, and in the following chapters develops an in-depth look at the handling of different types of files e.g. serial files, indexed files, direct access files, and searching and sorting. A separate chapter is devoted to hierarchical and relational database design, and the book concludes with a chapter of practical advice on how best to develop file handling programs.

The topics covered by the book include:

Card Index Files, Serial Files, File Headers, Disc and Record Buffering, Using Pointers, Indexing Files, Searching Techniques, Hashing Functions, Sorting Methods, Testing and Debugging, Networking Conflicts, File System Calls

The associated disc contains complete working programs based on the routines described in the book and a copy of Filer, a full-feature Database program originally published in BEEBUG magazine.



ASTAAD

Enhanced ASTAAD CAD program for the Master, offering the following features:

- * full mouse and joystick control
- * built-in printer dump
- * speed improvement
- * STEAMS image manipulator
- * Keystrips for ASTAAD and STEAMS
- * Comprehensive user guide
- * Sample picture files

	Stock Code	Price
ASTAAD (80 track DFS)	1407a	£ 5.95
Applications II (80 track DFS) *	1411a	£ 4.00
Applications I Disc (40/80T DFS)	1404a	£ 4.00
General Utilities Disc (40/80T DFS)	1405a	£ 4.00
Arcade Games (40/80 track DFS)	PAG1a	£ 5.95
Board Games (40/80 track DFS)	PBG1a	£ 5.95

	Stock Code	Price
ASTAAD (3.5" ADFS)	1408a	£ 5.95
Applications II (3.5" ADFS)	1412a	£ 4.00
Applications I Disc (3.5" ADFS)	1409a	£ 4.00
General Utilities Disc (3.5" ADFS)	1413a	£ 4.00
Arcade Games (3.5" ADFS)	PAG2a	£ 5.95
Board Games (3.5" ADFS)	PBG2a	£ 5.95

All prices include VAT where appropriate. For p&sp see Membership page.

Board Games

SOLITAIRE - an elegant implementation of this ancient and fascinating one-player game, and a complete solution for those who are unable to find it for themselves.

ROLL OF HONOUR - Score as many points as possible by throwing the five dice in this on-screen version of 'Yahtzee'.

PATIENCE - a very addictive version of one of the oldest and most popular games of Patience.

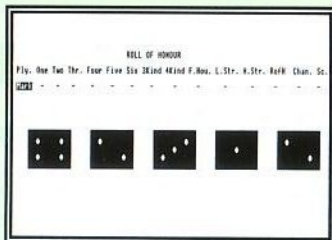
ELEVENENSE - another popular version of Patience - lay down cards on the table in three by three grid and start turning them over until they add up to eleven.

CRIBBAGE - an authentic implementation of this very traditional card game for two, where the object is to score points for various combinations and sequences of cards.

TWIDDLE - a close relative of Sam Lloyd's sliding block puzzle and Rubik's cube, where you have to move numbers round a grid to match a pattern.

CHINESE CHEQUERS - a traditional board game for two players, where the object is to move your counters, following a pattern, and occupy the opponent's field.

ACES HIGH - another addictive game of Patience, where the object is to remove the cards from the table and finish with the aces at the head of each column.



Applications II Disc

App011 (009)	Owner	14 39 34	
PS	Option 00 (011)		
Dir. App011	Lib. Librari1		
ISLAND	UR/	F	UR/
FOREIGN	UR/	SHARES	UR/
TEMP	UR/		UR/
7X8			
App011 (009)	Owner		
PS	Option 00 (011)		
Dir. App011	Lib. Librari1		
ISLAND	FFFF3000	FFFF3000	005000
UR/	08.06.92	000E48	
F	00003747	00003747	0005C8
UR/	17.06.92	005144	
FOREIGN	FFFF6000	FFFF6000	002000
UR/	17.06.92	004690	
SHARES	FFFF7C00	FFFF7C00	004000
UR/	17.06.92	003824	
TEMP	FFFF7C00	FFFF7C00	004000
UR/	17.06.92	000592	
SHARE CLOCK	7C00 8000	FFFF7C00	FFFF7C00

CROSSWORD EDITOR - for designing, editing and solving crosswords

MONTHLY DESK DIARY - a month-to-view calendar which can also be printed

3D LANDSCAPES - generates three dimensional landscapes

REAL TIME CLOCK - a real time digital alarm clock displayed on the screen

RUNNING FOUR TEMPERATURES - calibrates and plots up to four temperatures

JULIA SETS - fascinating extensions of the Mandelbrot set

FOREIGN LANGUAGE TESTER - foreign character definer and language tester

SHARE INVESTOR - assists decision making when buying and selling shares

LABEL PROCESSOR - for designing and printing labels on Epson compatible printers

Arcade Games

GEORGE AND THE DRAGON - Rescue 'Hideous Hilda' from the flames of the dragon, but beware the flying arrows and the moving holes on the floor.

EBONY CASTLE - You, the leader of a secret band, have been captured and thrown in the dungeons of the infamous Ebony Castle. Can you escape back to the countryside, fighting off the deadly spiders on the way and collecting the keys necessary to unlock the coloured doors?

KNIGHT QUEST - You are a Knight on a quest to find the lost crown, hidden deep in the ruins of a weird castle inhabited by dangerous monsters and protected by a greedy guardian.

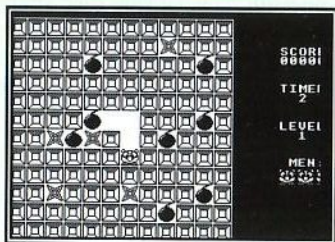
PITFALL PETE - Collect all the diamonds on the screen, but try not to trap yourself when you dislodge the many boulders on your way.

BUILDER BOB - Bob is trapped on the bottom of a building that's being demolished. Can you help him build his way out?

MINEFIELD - Find your way through this grid and try to defuse the mines before they explode, but beware the monsters which increasingly hinder your progress.

MANIC MECHANIC - Try to collect all the spanners and reach the broken-down generator, before the factory freezes up.

QUAD - You will have hours of entertainment trying to get all these different shapes to fit.



	Stock Code	Price
File Handling for All Book	BK02b	£ 9.95
File Handling for All Disc (40/80T DFS)	BK05a	£ 4.75
Joint Offer book and disc (40/80T DFS)	BK04b	£ 11.95
Magscan (40 DFS)	0005a	£ 9.95
Magscan (80T DFS)	0006a	£ 9.95
Magscan (3.5" ADFS)	1457a	£ 9.95

	Stock Code	Price
File Handling for All Disc (3.5" ADFS)	BK07a	£ 4.75
Joint Offer book and disc (3.5" ADFS)	BK06b	£ 11.95
Magscan Upgrade (40 DFS)	0011a	£ 4.75
Magscan Upgrade (80T DFS)	0010a	£ 4.75
Magscan Upgrade (3.5" ADFS)	1458a	£ 4.75

All prices include VAT where appropriate. For p&p see Membership page.

Tel. (0727) 840303

Fax. (0727) 860263



512 Forum

by Robin Burton

This month the Forum is a bit of a mixed bag, but I hope there's a bit to interest everyone and perhaps a timely reminder to anyone who hasn't given sufficient thought to what happens after April. We'll start with a point raised by a reader in response to my articles on PKZIP.

FINAL PKZIP POINTS

Thanks to Philip Draper (why is it always the same people?) for dropping me a line saying how much he'd enjoyed the PKZIP articles and that he'd learned a bit more about the program as well. Thanks too for reminding me about one quite important point about PKZIP which, frankly, I'd forgotten to mention. My excuse is that I use a hard disc and, naturally enough, lack of free disc space is rarely an issue. However, if you're restricted to floppies there's a point which you must bear in mind and which I completely failed to warn you about.

When PKZIP is compressing data it creates a temporary file which it uses as virtual memory, unless the total amount of data is very small so both the input and output fit into RAM. Obviously the quantity of output data is usually much less than the input, or there wouldn't be much point in the exercise, but equally the two together will often exceed the amount of free RAM in the machine, even in an expanded 512.

For example, if an 800K disc contains, say, 700K which compresses to 250K, added to the space for PKZIP plus sundry workspace the total's about a

megabyte. This clearly won't fit into RAM and the source floppy doesn't have enough free space either, so unless you tell PKZIP how to handle the problem the job can't succeed and you'll get a 'disc full' message. Mind you, at least PKZIP does tidy up after such an event, but that sometimes adds to the confusion because you can't actually see what happened after the evidence has been cleared away.

By default PKZIP creates its temporary file on the current drive, so if in the above case the current drive contains too much source data there's no chance. However, if the current drive is the (empty) destination drive, or you tell PKZIP to put the temporary file on the destination disc everything will be fine. It's quite easy and it can be done in two ways. The most obvious way is another command line directive, although in this case the letter used, "-b", isn't as obvious as other PKZIP options. Just add the '-b' option plus the temporary file's drive ID to the command like this:

```
PKZIP -a -bb: <archive-name>
```

which would tell PKZIP to use drive B: for its temporary file, assuming you're zipping from A:. Alternatively, you can permanently set the option as an environment variable, say in your autoexec.bat or in the security job itself, by the command: SET PKTMP=C: although this option is likely to be more useful with a hard disc.

MAKING PLANS

To many of you the rest of the Forum will be well known, but in my experience a fair number are always caught napping when anything changes. Also, to be fair,

there may be a few readers who've purchased second-hand machinery recently and who therefore haven't had the benefit of year's of BEEBUG magazine to learn about everything useful. It should by now be no secret that BEEBUG will cease publication after a couple more issues, in April.

Although it's sad for those of us who still enjoy our old machines, whether 512 or 6502, we must be realistic and face the fact that technology moves ever onward and in truth the old machine is now something of a dinosaur, no matter how useful you still find it and how much dinosaurs are in fashion at the moment. All credit to BEEBUG for continuing support so long after all others lost interest I say, but inevitably all good things come to an end.

No doubt a good many of you intend to continue to use your old faithful machine for some time yet. Why not? I for one certainly do. In fact I'll probably use mine until it breaks down and I can't get the bits to fix it, although the writing's already on the wall because many chips which are now out of production are already difficult to get hold of and some just aren't available at all. A bit of advice! DO NOT blow up your Tube ULA!!!

I don't know if you've considered all the implications of being 'alone' with your machine, but now's the time if you haven't done so before. Even forgetting potential hardware problems, there will no longer be any automatic regular source of information, hints, help, software or news. If you don't already belong to a computer club or user group and don't do something about it soon you'll be entirely cut off from all other BBC micro users after April. Likewise, those few people who can still supply

software for the BBC or 512 will be cut off from you too, so don't leave it any longer to take steps to minimise this.

One excellent source of hundreds if not thousands of programs is Alan Blundell's BBC PD. I've mentioned BBC PD before and make no apology for doing it again. I have no connection with it and while modesty or good taste might prevent Alan from advertising his services too blatantly I suffer no such misgivings. For the committed BBC or 512 user there is a range of software in BBC PD's catalogue that would take you months to wade through and evaluate. An extra benefit is that costs are minimal. There's no excuse for being bored or dissatisfied with old software just because BEEBUG magazine is no longer published. Contact BBC PD now if you haven't done so before and help Alan to continue his sterling work. Of course you could be complacent and leave it a year or so before you do it, then you might find he's been forced to close down too owing to lack of support. If you're guilty don't say you weren't warned!

OTHER SOFTWARE

On a similar note Essential Software, run by yours truly, has been operating for several years now, since 1989 in fact, but still not everyone knows about it. I've not previously shouted about it in this column too much, but in fact even when regular reviews of 512 products or software were being published in BEEBUG some readers managed to miss them, even when they were on the page adjacent to 512 Forum! With only three issues to go, in addition to BBC PD I'm going to make sure everyone knows about Essential Software before it's too late. If Mike Williams disapproves perhaps I'll get sacked from writing 512 Forum, but I'll just have to take that risk.

There follows a brief outline of ES's 512 software range. You will see that prices have been reduced by about 50% as a final gesture to Forum readers from me (original price in brackets). If there's anything you want, now is the time to get it. This isn't (quite) a closing-down offer so there's no time limit on the offers as such and I'll supply everything, as well as still answering letters, as long as I can, but in purely practical terms I can't keep adequate stocks of EPROMs and floppy discs for ever and neither is it realistic to buy them in ones and twos 'ad hoc'. Inevitably at some point I'll have to call a halt, so this time next year could be too late. Memory use is shown only for programs that use 512 RAM, otherwise it's zero. All cheques payable to me please, and overseas orders add £1.00 to EPROM prices. If you need more info. please send an S.A.E. I also have the latest (and definitely final) update to PCCE and TXTMOUSE from David Harper, making both programs probably as near perfect as is possible in the 512 - yours for £1 with any ES order, or £2 on their own as before.

THE LIST

RAMDISC (£16.95) £9.95

Allows a 512 ramdisc to be configured as any size from 10K upwards and as any drive. Includes AMNESIA, allowing deletion of the ramdisc without re-boot, and DISCID which allows switching of drive IDs plus suspension of any drive so it can't be accessed (and reinstatement!).

INTERCOM (£12.95) £6.95.

A command line editor with history recall. Commands can be recalled manually or by automated matching against a (wildcarded) search pattern. Word processor like editing during command entry. Configurable, memory resident, with 1k buffer uses 2.7K of RAM.

PFKEYS (£7.95) £4.95

Allows up to 30 user defined commands to be assigned to Shift/Ctrl/Alt function keys, so commands can be executed by a single key-press. Command can be saved to disc, re-loaded or amended as required. Configurable, memory resident, uses 3K of RAM.

HDSETUP (£10.95) £5.95

Partitions a winchester for any DOS partition from 1 to 32Mb. in 1Mb. steps, plus optimised directory sizing for smaller partitions.

SUPRSTAR (£14.95) £7.95

Gives an independent, true mode 7 screen for star commands, activated by hot-keys. Star commands issued previously are retained and the DOS screen plus the application or activity are preserved too. In SUPRSTAR normal BBC micro facilities include full BBC cursor editing, proper Escape processing, programmable function keys and correct MOS error handling.

GOBBC/512 (£14.95) £7.95

With SUPRSTAR £14.00 the pair.

Allows you to 'drop into' a completely normal BBC environment from the 512 and run any BBC task, then return to DOS where you left it. (NB. You must have SUPRSTAR for this.)

PRNTSCRN (£14.95) £7.95

Prints any DOS display (graphics text or both). Two programs, GRDUMP and PRNTSCRN, both actioned by hot-keys hence can be used at any time. Configurable for 9 or 24 pin printers, using Epson commands, full IBM graphics support. Can also print SUPRSTAR screens.

CLMOUSE (£12.95) £6.95

Provides mouse cursor control for programs which don't normally use a

mouse. Mouse buttons can be configured to generate any key press (defaults Return and Escape) and sensitivity can be adjusted as required. Settings can be saved to or loaded from disc. Trackerball compatible, application independent.

SCRNSAVE (£9.95) £5.95

Allows the DOS display to be saved directly to either BBC filing system. Filing system, drive and filename are all user definable. Triggered by hot-keys, so any display can be saved including those from within applications.

TRANSLATE (£14.95) £7.95

Allows any (PC or BBC) keypress to be generated by any key on any model of BBC micro. Includes numeric keypad for model B/B+, plus otherwise unavailable PC keys on all models, eg. right-shift, scroll-lock. Definitions can be saved to/loaded from disc and can include Shift and/or Ctrl modifiers.

MEMOPAD (£9.95) £5.95

A 'pop-up' notepad providing 10K bytes (4 A4 pages of text) of instant, independent note-taking storage. Contents of the pad can be saved to or loaded from disc, cleared or printed. Access by hot-keys, uses one BBC sideways RAM slot. Configurable for all sideways RAM types.

MISCELLANEOUS I (£11.95) £6.50

Includes:

Select - a batch file menu driver providing the missing link between the user and batch files, allowing on-line selection of options in batch files.

Colordef - allows screen colours to be changed on hot-keys instantly. Change foreground or background colours whenever you like without access to the command line.

Suspend - pauses the 512. Suspend and resume are on hot-keys, stop the machine any time.

Lock - locks the machine from the command line with a user-entered password. The password is needed again to release the system, when LOCK reports if illegal access was attempted. Proof against all but Break.

Sound - allows full access to the BBC micro's sound facilities from DOS, parameters exactly as BBC Basic.

Envelope - allows sound envelopes to be defined in DOS, exactly as BBC Basic.

MISCELLANEOUS II (£14.95) £7.95

Includes:

Keyclick - adds 'key clicks', like a PC. Clicks can be loud, quiet and switched on or off as required.

Oscli - * commands direct from DOS without clearing the screen. Much smaller and faster than STAR, plus no screen disruption, especially useful in batch files.

Cursor - allows changes to the size/flash-rate of the DOS cursor on hot-keys. Scrnpres - blanks the display if the machine isn't used for a (user-defined) period preventing screen 'burn-in'. The display is reinstated by any key press.

List - is a cursor controlled, bi-directional scrolling text lister. Includes line-up/down, page-up/down, start/end of file, go-to line, ASCII filter, tab expansion, auto browse etc. A much more flexible replacement for 'TYPE'.

Lockword - like SUSPEND, but needs a (user-defined) password to release 512.

Public Domain

Alan Blundell looks at two professional-quality new releases into the world of PD.

I have a couple of new items of PD software to report again this issue, both of which are of very high quality. I keep thinking that there must be an end to the new software which keeps appearing, especially as the circle of Beeb programmers shrinks, but these gems just seem to keep appearing.

ADDOS

ADDOS is the name given to a modified version of Acorn's own 1770 Disc Filing System (DFS) which has been produced (by modifying the original Acorn product) by Andrew J Donald. The DOS part of the name (I suspect) comes from the Disc Operating System (DOS) used on PC type computers, but the AD part is more obvious.

The original Acorn DFS was designed around a disc controller chip called the 8271, which worked in single density disc formats only. When Acorn introduced the Master series, it took the opportunity to change to using a 1770 disc controller chip, which is capable of double density operations as well as single density. Single density and double density, for those who are unfamiliar with these terms, are basically two different ways of recording the magnetic data on a disc surface. Fairly obviously from the names, double density recording fits more data onto a disc than single density, but the name is nevertheless a little misleading - you don't get twice as much data on a double density disc, more like 1.6 times the amount available in single density mode. This meant that the DFS had to be rewritten to work with the new chip, resulting in the Acorn 1770 DFS. This DFS worked just like the original (8271) DFS, in that it worked in single density only. Double density disc recording was left for the ADFS ROM, but other

software suppliers took the opportunity to introduce double density DFS systems.

These never achieved very widespread use because there was no guarantee that someone you might send a disc to could use double density DFS discs. Andrew J Donald saw the potential benefits of having a double density DFS and, rather than simply buying a non-Acorn alternative, set to work on Acorn's own 1770 DFS. He very cleverly succeeded in modifying it to work in double density rather than single density, and has managed to persuade Acorn to allow him to distribute the resulting ROM image via the public domain. He kindly sent me a copy of the ROM image, which I have tried out on a Master 128 and have found it to work fine.

Andrew has included a short text file with the ROM image which gives a few pointers on the use of the new double-density DFS, but has also written an article giving more details, including a description of how he made the conversion work. The article will be published in a magazine for Beeb enthusiasts called *ByteBack*.

ByteBack is a newish A5 format magazine for BBC users and enthusiasts, run by Paul Harvey and produced roughly once a month. It costs £1.00 per copy and is available from 33 King Henry's Mews, Enfield Lock, Middlesex EN3 6JS, should you wish to get in touch to find out more about ADDOS. ByteBack is in some ways similar to early issues of BEEBUG in that it includes hints and tips for beginners as well as more advanced material, and general discussion articles (in relation to the subject matter of this column it also includes a PD software reviews section, so it isn't entirely out of context here).

SIDEWAYS PRINTING

The second new program which I have to report on this issue is the latest from David Harper, whose name should be familiar to regular readers. David has this time come up with a program for the 6502 side of a Master (or Model 'B'), rather than the Master 512 co-processor for which he has been very prolific in the past. This program, a sideways printing utility, is up to his usual high standard and is very professionally presented.

The Sideways Printer is a program which allows text to be output "sideways" on a dot-matrix printer. In other words, text is printed in "landscape" format (rather than the usual "portrait" format), and is thus rotated by 90 degrees from the way it normally appears.

To use the program you need a 24-pin (or 48-pin) Epson-compatible dot matrix printer and a BBC Model B with OS Version 1.20, Model B+ or Master having at least one free bank of sideways RAM.

Apart from a couple of star commands to turn it on and off, the Sideways Printer is totally transparent to the user and to the system. This means it can be used with any word processor, or indeed with any other program (such as a spreadsheet) which outputs text to the printer. It is fully Tube compatible, and even works well on the Master 512.

The Sideways Printer can only be used for text output. It will not cope with pictures, diagrams, etc. which are output in dot graphics mode. To use it you must first load the ROM image file into a bank of sideways RAM. On the Master or B+128 you do this with a command such as:

```
*SRLOAD SIDEPR 8000 <n>
```

where <n> is the number of the sideways RAM bank you wish to use. You then need to initialize the program. Some versions of the Operating System (on the Compact and later versions of the Master 128) allow you to do this as you load by

adding an "I" to the end of the *SRLOAD command. Otherwise, the approved way is to press Ctrl-Break (in fact, you can also initialize the program by writing the value &82 to memory location &2A1+<n>, where <n> is again the number of the sideways RAM bank).

Note that this program must be run in sideways RAM. It will not work if you blow it into an EPROM. Some versions of sideways RAM sold for the Model B are designed to "look like" ROM. The Sideways Printer will not work with one of these.

For the sake of compatibility, the Sideways Printer has been designed to respond to the same sequences as an Epson printer normally does. Thus, for example, sending to the printer the two ASCII characters 27,52 (ESC,"4") starts printing in italics, whether the Sideways Printer is on or off. Most word processors have their own method of issuing control sequences to the printer, of course.

There are limitations, and the most severe one is that all sideways output occurs at 12 characters per inch (apart from doubling the character width). A number of printer effects have been implemented, however. The ones that work include bold, italics, underline, double-width and double-height characters, as well as various line spacings.

SOFTWARE SOURCES

Next issue, I promise that I will give some information about sources of PD software. This has been promised for a couple of issues now, but I am still checking which libraries and user groups are still active. If you run one or other such source of PD, please let me know as soon as possible so that I can make the information as complete as possible. As usual, my address is 18 Carlton Close, Blackrod, Bolton BL6 5DL.

B

Inter-Mail

Tim Parsons assesses a recent product for BBC micro users, Inter-Mail, a letter/correspondence manager for use with Computer Concepts' Inter-Base and Inter-Word.

Product	Inter-Mail
Supplier	Synectics 10 Bollin Close, Sandbach, Cheshire CW11 9TZ.
Price	£14.95 inc. (Cheques payable to M.T.Pickering)

At the outset I must stress that to use Inter-Mail you must have ADFS, shadow RAM, Inter-Word and Inter-Base (IB), together with a spare ROM socket or other sideways RAM for the Inter-Mail ROM image.

I purchased Computer Concepts Inter-Word a long time ago. When I upgraded to ADFS I used a directory for each correspondent, which worked quite well. I then purchased Inter-Base. Looking around for ready made Inter-Base programs I discovered Inter-Mail written by Martin Pickering, who has also written a very good book on the Inter-Base Programming Language. Among other things this explains very clearly how Inter-Base programs can communicate with Inter-Word, and for any serious users of Inter-Base it is a must.

Inter-Mail brings Inter-Word and Inter-Base together quite conclusively; all you have to do when initialising is to enter your name, address and phone number, and then, when required, enter the names and addresses of people you may wish to write to (there is also space for phone numbers, birthdays, reminders and any other relevant information). This will automatically result in an "IB. Card Index" Database called *Names* which is

indexed on the surname (and an index number). More names and addresses can be added at anytime.

From the Inter-Mail menu (Inter-Word lookalike, of course), you have nine options from which to choose. To start a letter you simply *GET* the address from the *Names* Card Index and enter Inter-word (all automatically). The reference number is placed in the top left-hand corner, your name and address is placed in the top right-hand corner with today's date below it. The recipients address is placed just below that and to the left, in bold (in the correct place for window envelopes). Inter-Mail is quite intelligent in its choice of Dear Sir/Madam or Christian name, and you are then ready to start your prose, assisted I trust by SpellMaster.

When finished, or even before you finish your current letter, it is then a very simple matter to return to the Inter-Mail menu by pressing F0 (this is most important) to save the letter with its reference number as the filename. This will then automatically create another "IB. Card Index" database called *LETLIST* which is indexed on the reference number. An optional comment can also be included. From the Inter-Mail menu you can if you wish *GET* any other address, which can be put into your current letter at the cursor position, or print out the current letter or even start a new letter with a different name and address.

While using Inter-Mail you also have full access to either the *LETLIST* or *NAMES*

```

INTER-MAIL
(C) Synectics 1991

1) Save letter on disc
2) Find old letter on disc
3) Find address
4) Enter new address
5) Change Options
6) Print letter
7) Start new letter 4.245
8) Start Mailshot
9) Utilities

14th November 1993
ESC INTER-WORD menu
<f0> To return here

Please enter choice

Current addressee is.
Mike Williams

```

The Inter-Mail menu

card index allowing you to make any alterations or additions you wish as you go. This is also possible with Inter-Base alone, but don't alter the index numbers.

It is also possible, and indeed recommended, to alter the options to suit contained in the !BOOT file. This contains a lot of the start-up settings such as Pre and Post Print, which are all automatically sent to Inter-word when starting a new letter.

THE MENU OPTIONS AVAILABLE

1. Save letter on disc

This is the same as the Inter-word menu option 1.

2. Find old letter on disc

This enables you to interrogate the *LETLIST* card index by entering the reference number (if known). It is possible to scroll through the whole list, but this is terribly slow. It would be better/quicker, to be able search for a surname, but this is not implemented.

3. Find address

This enables you to interrogate the *NAMES* card index, either by surname or by an index number. If not found correctly the first time, it is an easy matter to look at the neighbouring "cards" with the Ctrl and cursor up and

down keys. When the address is found there are many options available: (G)et addr, (L)etter, (A)dd, (E)dit, (P)rint, (F)ind, (D)elete, (ESC) menu.

4. Enter new address

With this option you can enter new addresses with as much additional information as you wish, such as birthdays etc., or just general reminders.

5. Change options

This enables the !Boot file to be altered, which will not (of course) come into effect until the disc is next booted.

6. Print letter

This performs the same as the Inter-word menu option 6.

7. Start new letter

This starts a new letter with the *current addressee* automatically placed in position for a window envelope.


8. Start Mailshot

This is the Mail Shot facility, which will print out letters to all the people with certain characters entered into the *CATEGORY* field of the *NAMES* card index.

9. Utilities

This is a built in "feature" (not a bug) which provides "hooks" into IB programs written by yourself! There is an excellent example supplied, a ROM image called *IUrom*, which in itself shows what can be done!

You may have noticed that I have found it very difficult to find fault with Inter-Mail; even the manual is beyond reproach. If you have the right equipment, Inter-Mail will make full use of it, although if you are using a floppy disc (I have been using a hard disc) you will probably find you eventually get the disc full message.

For prospective IB programmers there is plenty of scope for further expansion, which is where Martin's excellent book would come in handy. 

M-Base (Part 3)

Ian Palmer rounds off his database.

This is final section of the M-Base program. Other than the search described before in the viewing records section, there are three more searches, each initiated via option 8 of the main menu. This option leads to a further menu with four options, the fourth being return to main menu.

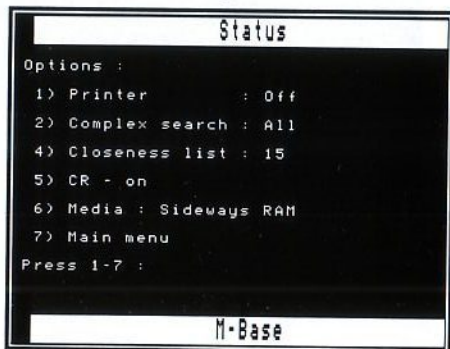
The first option is the global search. This is similar to the search described before, only it searches all fields, not just the first one. Press f0) to move through records found.

Complex search is a case sensitive search which allows you to specify which field to search, and you can even have more than one search string. When selecting this search you are presented with a blank record entry. Place the search strings in the fields required. When you have entered them all the search will commence. By default only records containing all the search strings will be displayed. By changing the state of the 'Complex Search' flag (option 9) to 'any', records which contain any of the search strings will be displayed. Records can be altered whilst they are displayed.

The final search is the 'Closeness search' and this requires a more detailed description as it is a very powerful search with unusual qualities. First of all it is important to point out that this search does not search for specific strings, in fact it can sometimes find records which apparently have nothing to do with what you are looking for, *apparently* being the operative word.

Like the complex search it allows more than one search string to be typed in. In fact the strings (or targets) are entered in exactly the same way as in the complex search. I will use the word 'target', instead of 'string', from now on because of the way the search works.

There are two types of data the search can handle. The first is numerical data, ideally percentages, e.g. 100, 54 etc. Only whole numbers will work properly, and the best results are produced if all fields carry the same weight.



The M-Base status display

An example of a database which might use this type of data is a computer games database, where ratings are given for features like adventure, action, speed, thought required, where each game has a value for each category related to how much of that feature is present in that game.

Thus, with this search, you could give your ideal ratings for a game you wish to play, and M-Base would then produce a list of the closest games to your ideal. The list starts with the closest and as it progresses the games get further from your ideal. The length of the list depends upon the value of the 'Closeness list' flag, default 15, but can be anything from 3 to 50.

It is possible to be vague about any targets given. If, for example, you don't mind about the exact value, but you want it to be above, say, 50 then placing '50>' as the target will allow this consideration. Similarly '50<' allows the search to find those of less than 50.

Company
Name: Risc User Magazine
Address 1: 117 Hatfield Rd
Address 2: St Albans
Tel: 0727 840303
M-Base Item : New

M-Base - entering a new record

Because each target carries the same weight, when you have more than one target you may notice individual fields fluctuating quite a lot. For example if you had one of your targets as '50<' you may start off with values below 50 for that field, then some above 50, then some below again. This is due to the fact that overall the records with values over 50 in that field were closer than some that had values lower than 50.

The second type of comparison which is possible in the closeness search is word analysis. With this comparison the target is broken down into single words and those words are searched for, so word order is not important, i.e. 'Three two one' is the same as 'One two three'. Also, if only some of the words are found, that record will be higher in the final list than those that have none of the words in it, but lower than any with all the words in it. But this is not all that is done. The target words are further broken down into word structures, and these too are used in the comparisons, so words with the same 'build' are searched for. i.e. 'Three' is the same as 'Free', 'Touched' is the same as 'Touchwood', etc.

You can mix the two types of comparison, i.e. numerical and word build. Although, as stated before, the best results are obtained when the numerical data is in the form of percentages.

M-BASE ON A BEEB

This program, with minor alterations, can be used on computers other than the Master range. First of all lines 1010 to 1040 need to be deleted, as do lines 1160 to 1190.

Then the value of DC% in line 1240 needs to be changed to TRUE, to make M-Base start up in disc mode. Also the 'ON O% PROCview ,' in line 2130 needs to be changed to a sequence of lines of the form : IF O%=1 PROCview : ELSE IF O%=2 PROCadd : ELSE etc.

To reduce the size of the program, lines which are specific to sideways RAM use can be removed, for example : lines 1790, 1800, 3330, etc. The only problem with running M-Base on other computers is memory. For this reason you need to lower page to a level which will allow access to disc, whilst giving enough memory for the program to operate.

```

10 REM Program M-BASE-3
40 REM BEEBUG Jan/Feb 1994
4740 REM ***** Search *****
4750 :
4760 DEF PROCsearch
4770 PROCclear:PROCTitle("Search",1):PROCTitle("M-Base : "+N$,22):PROCcentre("Select search type :",4)
4780 PROCcentre("1) Global ",6):PROCcentre("2) Complex",8):PROCcentre("3) Closeness",10):PROCcentre("4) Exit ",12):VDU23;11,0;0;0;0;0;
4790 REPEAT:ST%=VAL(GET$):UNTIL ST%>0 AND ST%<5
4800 IF ST%=4 PROCclear:ENDPROC
4810 IF ST%=1 PROCglobal:PROCclear:ENDPROC
4820 FOR v%=1 TO F$:S$(v%)="" :NEXT
4830 PROCsee(0):v%=1:REPEAT
4840 S$(v%)=FNinput(F$(v%),"",X$(v%),Y$(v%),L$(v%),32,126)
4850 IF INKEY=58 AND v%>1 v%=v%-1:ELSE v%=v%+1
4860 UNTIL v%>F% OR (key%>0 AND key%<12)
4870 IF ST%=3 PROCclose:PROCclear:ENDPROC

```

```

4880 PRINTAB(0,22);SPC(80);:PROctitle(
"Searching",22):FOR s%=1 TO N%:PROCread(
s%)
4890 f%=NOT(CS%)
4900 FOR A%=1 TO F%:IF INSTR(T$(A%),S$(
A%))=0 AND NOT CS% f%=FALSE:A%=F%
4910 IF INSTR(T$(A%),S$(A%)) AND CS% AN
D S$(A%)<>" " f%=TRUE:A%=F%
4920 NEXT:IF f% PROCalter(s%):PRINTTAB(
0,22);SPC(80);:PROctitle("Searching",22)
4930 NEXT
4940 PROCclear:ENDPROC
4950 :
4960 DEF PROCglobal
4970 PROCclear:PROctitle("Global Search
",1):PROctitle("M-Base",22)
4980 S$=FNinput("Search string : ","",0
,4,20,32,126):PRINTAB(0,22);SPC(80);:PR
Octitle("Searching",22)
4990 S$=FNlow(S$):FOR a%=1 TO N%:PROCre
ad(a%)
5000 FOR B%=1 TO F%:T$=FNlow(T$(B%))
5010 IF INSTR(T$,S$) PROCalter(a%):B%=F
%:PRINTTAB(0,22);SPC(80);:PROctitle("Sea
rching",22)
5020 NEXT:NEXT:ENDPROC
5030 :
5040 DEF PROCclose
5050 LOCAL G%,T%,s%,f%:PRINTAB(0,22);S
PC(80);:PROctitle("Searching",22)
5060 FOR G%=1 TO F%:IF S$(G%)<>" " AND V
AL(S$(G%))=0 AND S$(G%)<>"0" L$(G%)=FNle
x(S$(G%)):S$(G%)=FNlow(S$(G%)):ELSE L$(G
%)=""
5070 NEXT
5080 G%=0:FOR s%=1 TO N%:PROCread(s%)
5090 T%=0:FOR f%=1 TO F%
5100 IF(VAL(S$(f%))<>0 OR S$(f%)="0")AN
D INSTR(S$(f%),"<")=0 AND INSTR(S$(f%),"
">")=0 T%=T%+ABS(VAL(S$(f%))-VAL(T$(f%)))
5110 IF INSTR(S$(f%),"<") AND VAL(S$(f%
))<VAL(T$(f%)) T%=T%+ABS(VAL(S$(f%))-VAL
(T$(f%)))
5120 IF INSTR(S$(f%),">") AND VAL(S$(f%
))>VAL(T$(f%)) T%=T%+ABS(VAL(S$(f%))-VAL
(T$(f%)))
5130 IF L$(f%)<>" " T%=T%+FNfuzzy(S$(f%
),T$(f%),L$(f%))
5140 NEXT:IF G%=0 G%=1:C%(1)=s%:T%(1)=T
%:ELSE PROCins
5150 NEXT:FOR s%=1 TO G%:PROCalter(C%(s

```

```

%)):NEXT:ENDPROC
5160 :
5170 DEF PROCins
5180 IF T%>T%(G%) AND G%=MG% ENDPROC
5190 IF T%>T%(G%) G%=G%+1:T%(G%)=T%:C%(
G%)=s%:ENDPROC
5200 LOCAL Q%,W%,X%
5210 Q%=0:REPEAT:Q%=Q%+1:UNTIL T%<T%(Q%
) OR Q%>=G%:IF Q%=MG% T%(Q%)=T%:C%(Q%)=s
%:ENDPROC
5220 FOR W%=MG%-1 TO Q% STEP-1:C%(W%+1)
=C%(W%):T%(W%+1)=T%(W%):NEXT:IF G%<MG% G
%=G%+1
5230 T%(Q%)=T%:C%(Q%)=s%:ENDPROC
5240 :
5250 DEF FNlex(I$):LOCAL O$,A$,B$,A%,B%
,U%
5260 I$=FNlow(I$):i$=I$:O$="" :FORA%=1 T
O LEN(I$):U%=TRUE
5270 A$=MID$(I$,A%,1):B$=MID$(I$,A%+1,1
):IF INSTR("aeiouy",A$) U%=FALSE:IF RIGH
T$(O$,1)<>"*" O$=O$+ "*"
5280 IF (B$="h" AND (A$="g" OR A$="p" O
R A$="t")) OR A$="f" U%=FALSE:O$=O$+"e":
IF B$="h" A%=A%+1
5290 IF A$="x" AND RIGHT$(O$,1)<>"*" O$
=O$+"*" :U%=FALSE
5300 IF A$="z" OR A$="s" O$=O$+"s":U%=F
ALSE
5310 IF A$=RIGHT$(O$,1) U%=FALSE
5320 IF U% AND INSTR("djkgctfphw",A$) U
%=FALSE:IF RIGHT$(O$,1)<>"e" O$=O$+"e"
5330 IF U% AND INSTR("mn",A$) O$=O$+"^"
:U%=FALSE
5340 IF U% O$=O$+A$
5350 NEXT:=O$
5360 :
5370 DEF FNfuzzy(A$,T$,L$):LOCAL p%,pp%
,p%,B$,A%,B%
5380 B$=FNlex(T$):p%=0:pp%=0
5390 A%=0:B%=0:REPEAT:B%=B%+2:P%=p%:IF
INSTR(B$,FNgetword(L$)) A%=A%+1
5400 p%=P%:P%=pp%:IF INSTR(i$,FNgetword
(A$)) A%=A%+1
5410 pp%=P%:UNTIL P%>=LEN(A$):=75-(A%/B
%*75)
5420 :
5430 DEF FNgetword(I$):LOCAL O$,A$
5440 REPEAT:P%=P%+1:A$=MID$(I$,P%,1):IF
A$<>" " O$=O$+A$
5450 UNTIL A$="" OR P%>=LEN(I$):=O$ B

```


Wordwise User's Notebook: Multiplying the Mail

by Colin Robertson

One of the commonest chores that word processors get called upon to perform is the production of mailshots or multiple copies of a standard letter to be sent to different addresses; the letters differing only in the intended recipients name, address, and salutation (e.g. Dear Aunt Mabel).

This process is usually referred to as mail-merging; the facility to do it being a standard feature of many proprietary text processing packages. Unfortunately, only a crude suggestion of a segment program, for such a process, comes as standard with Wordwise Plus (WW+).

The WW+ utility described here aims to correct that deficiency, and is one I've used a lot; I find it takes quite a bit of the strain out of writing begging letters to multi-millionaires and football pools winners!

But no doubt you'll have your own ideas about using it: pleas for help with the local Scout, Guide or Brownie pack for instance, subscription reminders for a club or society, or maybe letters to be sent to members of the local Beeb user group. Whatever your particular interest, it'll certainly come in very handy when you need to send out several copies of a standard, but personalised letter.

BEFORE USE

Before using the multiple letter utility, a basic letter must be prepared, without

the name and address of the recipient, but with the keyword #Address (you can of course change this to anything that takes your fancy) at the position where a name/address is intended to go. This letter should then be saved in a file for later use when running the utility. The

Phone : 0011-112233	The Very Small Mammal Society, 23 Smogton Vale Crescent, Hemelworth New Town, Hertingfordshire, HH1 4HH. 20th April 1993.
#Address	1993 VOLE APPEAL
Dear	
	Many thanks for your contribution to our 1993 special appeal. Although the amount raised, has yet to be worked out, it does appear that we will reach, if not pass, our target figure.
	As you are aware, the Society is engaged in an extensive small mammal educational support programme. The money raised this year will enable us to provide better vocational training for voles, an area which has suffered severely from lack of funds following last year's cutbacks in Government spending. Funds permitting, we hope to get our VTS (Vole Training Scheme) back on course again by the early part of next year.
	<u>Dates for your diary</u>
Jun 5-	Rat cheese supper & disco. (All rats must be accompanied admission by ticket only)
Jun 10 -	Mouse club
Jun 17 -	Friends of the Gerbil (At Kevin's house)
	I am in the process of preparing a schedule of our activities over the Christmas period, and will send you a copy as soon as I can find it. Hopefully, the mice haven't been at it. In the meantime, I look forward to seeing you at our next meeting.
	Yours sincerely,
	Albert L. Brush (President)

Figure 1. Example standard letter

names/addresses to which the letter is to be sent, are also prepared separately

Mr D. Smith,
87 Sentinel Avenue,
Hemelworth New Town,
Hertingfordshire,
HH1 2HH.

Miss G. Romero,
82 Dawn Rise,
Pittscombe Terrace,
The Pitts,
Little Hemel,
Hertingfordshire,
HH2 2HH.

Mr A.C. Currant,
73 Resistor Road,
Ohmington,
Buckshire.

Peter S. Piper,
57 Variety Crescent,
Hemelworth New Town,
Hertingfordshire,
HH1 4HH.

Mrs G. Batten,
The Sticks,
34 Deal Way,
Woodwell,
Oakshire.

Figure 2.
Example address list

and held in their own file, with addresses separated by a blank line (carriage return). The first address should not be preceded by a blank line.

An example letter VOLELET, and address list VOLEADD, are given to illustrate what's required. Using these two files, follow the steps given below, to practice using the utility.

```
*|File - MULTEXE
*|EXEC file to load the multiple letter
*|mailmerge utility
*WORDWISE
:SELECT SEGMENT 0
:LOAD TEXT "MULTLET"
:SELECT SEGMENT 1
:LOAD TEXT "MULTPRN"
:SELECT SEGMENT 2
:LOAD TEXT "MULTRES"
:P%=FALSE
:SELECT TEXT
<RETURN>
<RETURN>
```

Lines beginning *| are comments ignored during execution, and <RETURN> denotes a carriage return.

The segment programs loaded by the above EXEC file are:

MULTLET - Multiple letter facility
(main program)

MULTPRN - Print letter

MULTRES - Reset utility operation

Having prepared the EXEC file as above, the utility can be set up by typing *EXEC MULTLET (or simply *MULTLET) from Basic or the Wordwise Plus menu mode. As described, MULTLET will load segments 0, 1, and 2. Other text areas are not affected. Segment usage can of course be altered to meet your own specific requirements. In fact just about anything can be changed with very little trouble.

USE OF SEGMENT/TEXT AREAS

Main text - Automatically loaded by utility with user specified letter

SEGMENT 0 - Multiple letter program

SEGMENT 1 - Print letter

SEGMENT 2 - Reset

UTILITY FUNCTIONS

The following functions are provided:

- Full prompts and error messages for ease of use
- Automatic insertion of addresses
- Automatic positioning of cursor for name insertion
- No restrictions on letter length or embedded commands
- Print letter function
- Reset function

SEGMENT PROGRAMS AND EXEC FILE

The utility is best set up by creating an EXEC file to load the appropriate segments with their programs, as in the following, where the file MULTEXE contains:

FUNCTION KEYS

Pressing Shift and one of the red function keys f0, f1, or f2 invokes the associated segment program to implement the following:

- f0 - Initial load of letter/insert address
- f1 - Print letter i.e. current contents of main text area
- f2 - Reset utility and clear main text area

METHOD OF USE

1. Load the MULTIPLE LETTER utility e.g. by typing *EXEC MULTLET or *MULTLET from either Basic or THE Wordwise Plus menu mode.
2. Enter Edit mode by pressing Escape.
3. Press Shift-f0 to execute segment 0. First time around, prompts will be given for the name of the standard letter file, and that of the address list. If these are answered correctly, the letter will be loaded into the main text area, and the first address will be inserted into the letter. The cursor will then be positioned immediately after "Dear " to allow the required name and/or form of address to be inserted (note that as set up, "Dear " is also mandatory, but once again you can change it to suit yourself). Subsequent operation of Shift-f0 will insert a fresh address in the letter already loaded.
4. If an address is to be omitted, simply press Shift-f0, and the utility will go on to the next one.
5. Press Shift-f1 to send the letter to the printer. The utility does not itself make use of formatting or printer control codes e.g. for paper positioning or paging. These should be embedded as

required in the basic letter. For Epson and compatible printers a likely command might be OC12 at the end of a letter to give a form throw. If you're using single sheets of paper, and the letter extends over two or more sheets, don't forget to use EM as well as EP so that WW+ will pause at page boundaries.

6. Press Shift-f0 to insert the next address. Repeat 3, 4 and 5 as required.
7. Should you run out of addresses, in other words, come to the end of the address file, the message:
"No more addresses in file - <address filename>"
will be displayed, where, not surprisingly, <address filename> stands for the name of the address file that you specified. After a brief wait while the message is displayed, the original (unaddressed) contents of the letter file will be displayed in the main text area, and operation of the utility suspended.
8. To reset the utility, press SHIFT+f2. This will reset everything and clear the main text area. The utility can be started again simply by pressing SHIFT+f0, whereupon fresh prompts for letter and address list files will be given.

The Break key has also been programmed to provide a reset function by simulating the action of pressing Shift-f2. So if you press this by accident, the utility won't be left in an 'interesting condition'. Should you do this, simply repeat utility operation from step 2 onwards, skipping (as described above) those addresses that you've already dealt with.

Wordwise User's Notebook

9. When you have finished using the utility, press Shift and f2. This resets everything, closes all files, and clears the main text area. The utility is left in place. To use it again, just repeat the above steps.

USING THE EXAMPLE LETTER AND ADDRESS LIST

Having loaded and/or reset the utility, follow the above steps from 2 onwards.

Enter VOLELET when prompted for the name of the letter file.

Enter VOLEADD when prompted for the name of the address file.

FINAL THOUGHTS

Although the utility is both useful and fairly robust, like every other program that's ever been written, there's still scope for amendments, improvements, additional features, and all manner of goodies, add-ons and bells and whistles.

For instance, I've deliberately kept the user input mechanisms as simple as possible, but they could be extended to provide file name validation, Escape trapping rather than disabling etc. Keyword use could also be made user/run-time selectable; formatting and special printing features could be added (e.g. pitch selection, style, size etc), and so on. So if you'd like to add your own features, please go ahead.

SEGMENT PROGRAM MULTLET

```
REM MULTLET(Multipl@ Letters)
```

```
REM Program to produce multiple copies  
REM of a standard letter. Each one can  
REM be uniquely named and addressed
```

```
REM using addresses taken automatically  
REM from an address file. The contents  
REM of the letter are available in the  
REM main text area for users to modify  
REM for each recipient.
```

```
REM Disable ESCAPE  
*FX200,1
```

```
REM Check to see if letter has already  
REM been loaded.
```

```
IFF%=FALSE THEN GOTO start-prog
```

```
REM Program BREAK key to execute the  
REM segment 2 reset function  
*KEY10":DOLINE""||GSEG2""|M|M"
```

```
SELECT TEXT  
CURSOR TOP
```

```
REM Check to see if all the addresses  
REM have been used up.
```

```
IFN%=TRUE THEN GOTO enter-name
```

```
REM Delete previous address  
FIND MARKERS
```

```
IF EOT THEN PROC no-markers  
FKEY 7
```

```
GOTO insert-name-and-address
```

```
.start-prog
```

```
CLOSE#0  
CLS
```

```
PRINT  
REPEAT  
VDU131,141
```

```
PRINT " Multiple Letter Utility  
"
```

```
TIMES 2  
PRINT
```

```
REM Set yellow background
```

```
REPEAT  
VDU131,157  
PRINT
```

```

TIMES 11

REM Reposition cursor +display
REM instructions in bluetext

VDU31,2,5,132
PRINT "Producesmultiple copies of a"
VDU31,2,6,132
PRINT"standard letter in asemi-
automatic"
VDU31,2,7,132
PRINT"fashion. Addresses areinserted"
VDU31,2,8,132
PRINT"automatically from thespecified"
VDU31,2,9,132
PRINT "addressfile. Letters may beof"
VDU31,2,10,132
PRINT "anylength."
VDU31,2,12,132
PRINT "Press SHIFTplus function keyfor"
VDU31,2,14,132
PRINT "f0=nextaddress, f1=print,
f2=reset"

REMLetter/program setup flag
F%=TRUE
REMAddress file exhaustedflag
N%=FALSE

PRINT
VDU31,0,15
REPEAT
PRINT
PRINT "Enter letter file name"
PRINT "or press RETURN to cancel :";
A$=GLK$
IF A$="" THEN PROCend
A%=OPENIN A$
IF A%=0 THENPROCfile-not-found
UNTIL A%<>0

REPEAT
PRINT
PRINT "Enter address file name"
PRINT "or press RETURN to cancel :";
B$=GLK$
IF B$="" THEN PROCend
B%=OPENIN B$

```

```

IF B%=0 THENPROCfile-not-found
UNTIL B%<>0

SELECTTEXT

REM Get basic letter into textsegment

LOAD TEXT A$
CLOSE #A$

REM FindAddress position in text

FIND"#Address"
IF EOT THENPROCaddress-key-missing
DELETEWORD

.insert-name-and-address

FKEY3

REM Get address from file andinsert
REM in letter.

IF EOF#B% THENPROCno-more-addresses
REPEAT
C$=GLF$#B%
IF C$<>"" THEN TYPE C$+"|R"
UNTIL C$="" OREOF#B%

FKEY 3

.enter-name

REM Positioncursor for entering name

FIND"Dear"
CURSOR RIGHT 5

REM EnableESCAPE
*FX200,0
DISPLAY
END

.address-key-missing
PRINT
PRINT "<#Address> keyword missingfrom
letter"
PROCerror-output

```

Wordwise User's Notebook

```
PROCend
ENDPROC

.error-output
  VDU 7
  TIME=0
  REPEAT
  UNTIL TIME=150
  CLS
ENDPROC

.file-not-found
  PRINT
  PRINT "File not found"
  PROCerror-output
ENDPROC

.no-more-addresses
  CLS
  PRINT
  PRINT "No more addresses in file - "+B$
  PROCerror-output
  CLOSE#B%
  FKEY 3
  FKEY 7
  TYPE "#Address"
  N%=TRUE

REM Enable ESCAPE
*FX200,0

  DISPLAY
  END
ENDPROC

.no-markers
  CLS
  PRINT
  PRINT "Markers missing from letter"
  PROCerror-output
  CURSOR TOP
  DISPLAY
  PROCend
ENDPROC

.end
  CLOSE#0
  F%=FALSE
```

```
REM Enable ESCAPE
*FX200,0
SELECT TEXT
*WORDWISE
END
ENDPROC
```

SEGMENT PROGRAM MULTPRN

```
REMP.MULTPRN
```

```
REM This program forms part of the
REM MULTIPLE LETTER utility. It runs
REM in SEG 1 and provides a print
REM facility which avoids the use of
REM menu mode.
```

```
SELECT TEXT
PRINTTEXT
DISPLAY
END
```

SEGMENT PROGRAM MULTRES

```
REM P.MULTRES
```

```
REM This program forms part of the
REM MULTIPLE LETTER utility. It runs
REM in SEG 2 and provides a reset
REM facility. From a good housekeeping
REM point of view, it's a good idea to
REM use this (by pressing SHIFT+f2)
REM once all addresses have been used
REM or you've finished with the
REM utility, since it ensures that all
REM files are closed and things are
REM tidied up.
```

```
CLOSE #0
F%=FALSE

REM Enable ESCAPE
*FX200,0

SELECT TEXT
DELETETEXT
DISPLAY
END
```

B



SIDEWAYS MOVE

I have a BBC model B fitted with a Watford 32K shadow RAM card and ATPL sideways ROM card. The latter is fitted with 16K of RAM. I now wish to make copies or backups of my numerous ROMs by copying the contents of a selected ROM to disc so that it could be subsequently be loaded to RAM at &8000 and operated from there.

The problem is that I do not know how to go about this. Is there any method by which I can move, say, the contents of ROM number 7 into sideways RAM without activating the ROM itself, and then follow this with a *SAVE "ROMNAME" 8000 C000 to save an image of the ROM to disc for subsequent reloading at &8000.

Also, have you come across any Basic program to convert Irish Grid References to latitude and longitude and vice versa? I already have programs which cater for National Grid References. I am also interested in software which will predict the position of the stars and planets at any given time.

Jeffrey Clarke

*The task of copying the contents of a ROM to disc for subsequent loading into sideways RAM is not as simple as it sounds. If you attempt to use Basic, then the Basic ROM is paged in, not the one you want. The Master 128 supports commands like *SRLOAD and *SRWRITE for accessing sideways ROM and RAM areas, and similar commands may be implemented by proprietary products too. The following issues of BEEBUG contain articles relating to this: Vol.8 No.1, Vol.9 No.2 and Vol.11 No.9. maybe other readers can help.*

We are unaware of any program to work with Irish Grid References - again other readers may be able to help. As far as stars and planets are concerned, articles and programs have appeared in past issues of BEEBUG: Vol.7 No.6, Vol.8 No.7 (magazine disc) and Vol.11 No.8.

CONGRATULATIONS

I would like to congratulate all concerned in writing this marvellous little magazine. As secretary of a national organisation, I have found the *JobLog* program (Vol.11 No.7) of enormous assistance. Being also in the middle of a nationwide survey, the *Census* suite of programs (Vol.12 No.3) have come in the nick of time.

When you eventually cease publication there will be a really big gap in this far-flung outpost of the Commonwealth.

Peter Holt
New Zealand

VIEWING SPACE

I cannot get the View Printer Driver from BEEBUG Vol.12 No.4 to work satisfactorily. I have repeatedly checked the accuracy of the program I copied from the magazine but cannot find an error. My computer is a Master 128 with a Panasonic KX-P1081 printer. The computer works satisfactorily as far as I have tried to use it except for line spacing. When printing a View document the printout is double-spaced. I have tried entering the stored command LS 0 but this has no effect, and LS 1 results in only one line of a document being printed per page.

M.A.Broughton

*One question to ask is whether the same results occur both on screen and to the printer. One answer may lie simply with the DIP switch settings on the printer. Normally one such switch selects either CR alone, or CR/LF at the end of each line. If this is in the CR/LF position then try the alternative setting. Otherwise both the computer and the printer are generating linefeeds giving the double line spacing experienced. Alternatively, using *FX6,10 will inhibit the computer from sending its own linefeed character. You may need to experiment to get it right.*

ENHANCED HEARING

Having typed in and used the 'Hearing Test' program (BEEBUG Vol.12 No.4), it occurs to me that it would be easier to let the computer record the levels as they are decided, and then print them out so that they could be entered onto graph paper at ones leisure. In this way one can concentrate on the test without having to bother with recording the results. I have found this can be achieved by inserting the following lines into the program:

```
60 DIM store(100)
65 DIM freq(100)
70 L=1
80 tone=131
1382 store(L)=yournoise%
1384 freq(L)=tone
1386 L=L+1
1388 tone=tone*2^(1/12)
1455 LOCAL P
1465 CLS: FOR P=1 TO L-1
1467 VDU 2
1470 PRINT " "; P " Level ";store(P);
" at a frequency of ";INT(freq(P))
```

```
1475 NEXT P: VDU 3
1477 ENDPROC
1480 :
```


B.W.Fursman

WORDWISE-MAIL - THE LAST WORD?

I wrote WW-Mail many years ago for my own use. It was never intended to be sold. WW-Mail lacks many features because, for writing letters to my friends, I did not need them. Corplan is undoubtedly a more comprehensive package intended for the professional, and Colin Robertson deserves much praise for its design. If you want something simple, which you can easily modify then WW-Mail might do.

If you are planning to review Inter-Mail (my similar program) then may I get in first and say that the same comments apply. However, Inter-Mail supports DFS and ADFS, hard discs, and recognises the date on the Master. It has more features including label printing and Mail-Shot provision.

Martin Pickering

Interested in Inter-mail - see this month's review. 

512 Forum (continued from page 33)

When locked the screen display is blanked, perfect security if little fingers or big noses might interfere while you're away.

CPFS - 16K EPROM plus disc (£24.95) £14.95

The Co-Pro Filing System uses the (internal or external) 512 memory as a ramdisc for the BBC micro. CPFS is a full filing system (with extras) and can be used with any application. Transfer rate up to 125K bytes/sec. Up to 127 files in any number of directories.

CPFS+ - 16K EPROM plus disc (£29.95) £19.95


Provides the facilities of CPFS plus a configurable printer buffer of from 4K to

1000K in 4K steps. Memory not used by the buffer remains available for CPFS files.

FASTBOOT (Sideways RAM) (£10.95) £5.95 EPROM (£14.95) £8.95

Boots the 512 in half the time (13-15 secs.) of a 640K disc and there's no need for a disc change from 800K to re-boot either.


IN CLOSING

That's all for this month except to remind you that if you want to write to me about any Forum topic, past or present, before BEEBUG ceases publication you'd better do that quickly too, and of course to wish you all a happy new year. 

Points Arising . . .

MAGSCAN INDEX FOR Vol.12 No.6

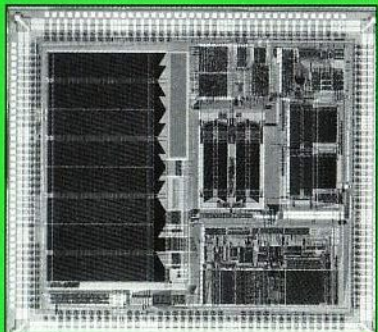
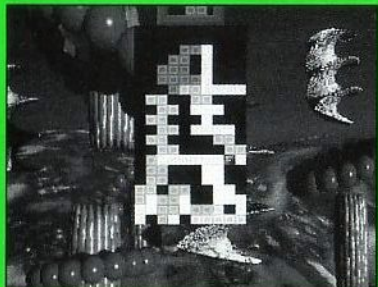
The index supplied on the magazine disc for this issue has a small omission. The second entry in the file MUST be

preceded by a line containing: ^D as with the first entry in the file. Use Edit (on the Master 128) or a word processor to make this change. 

RISC

user

**The number one
subscription
magazine for the
Archimedes**



SUBSCRIPTION DETAILS

As a member of BEEBUG you may subscribe to RISC User for the reduced rate of £18.40 (a saving of £1.50 on the normal subscription rate).

Overseas subscription rates are as follows:

£27.50 Europe and Eire, £33.50 Middle East, £36.50 Americas & Africa, £39.50 Elsewhere

RISC User, probably the most popular subscription magazine for the Archimedes, offers all the information you need as an Archimedes user. In every issue of RISC User you will find a wealth of articles and programs with professionally written reviews, lively news, help and advice for beginners and experienced users, and items of home entertainment.

The B5 size of RISC User allows a sophisticated design, big colour illustrations and pages full of information, and yet is still a convenient size to assemble into an easy-to-use reference library. Altogether, in its six years of existence, RISC User has established a reputation for a professional magazine with accurate, objective and informed articles of real practical use to all users of Acorn's range of RISC computers.

Contents of the latest Vol.7 Issue 3 of RISC User:

INTRODUCING THE ARM7 FAMILY

An introduction to the new ARM processors expected to provide the power for Acorn's next generation of computers.

SELECT AND COLLECT: PUBLIC DOMAIN GAMES

A description of the very best games titles available as public domain software.

IMPRESSION STYLE

A review of Computer Concepts' latest development to its Impression DTP software.

TWO CD-ROM SYSTEMS

A comparative review of two low cost CD-ROM drives from Morley and Cumana.

FIREWORDZ: INTEGRATED SOFTWARE FROM COLTON

An examination of the merits of integrated software in the form of *Firewordz* from Colton Software.

WIMP TOPICS

A major series aimed at readers interested in Wimp programs and Wimp programming. Each article looks at aspects of a particular topic.

WRITE-BACK

The readers' section of RISC User for comment, help, information - a magazine version of a bulletin board.

INTO THE ARC

A regular series for beginners.

TECHNICAL QUERIES

A column which answers your technical queries.

THE DOS SURVIVAL GUIDE

A series of articles on how to use the PC Emulator.

Graffer (continued from page 8)

```
2070 FOR B=1 TO N:MTH$(B)=MT$(B+START-1
):NEXT B
2080 DATA JAN,FEB,MAR,APR,MAY,JUNE,JULY
,AUG,SEPT,OCT,NOV,DEC
2090 DATA JAN,FEB,MAR,APR,MAY,JUNE,JULY
,AUG,SEPT,OCT,NOV,DEC
2100 ENDPROC
2110 :
2120 DEF PROCopt:LOCAL X
2130 *FX113,1
2140 *FX112,1
2150 VDU28,0,26,79,3:REM Sets text wind
ow to middle 22 lines.
2160 CLS
2170 REPEAT:PRINT""1 - Run Programme""
"2 - Plot Another Graph on Same Axes""3
- Save Graph to File""4 - Print Graph""
"5 - Quit":X=GET
2180 UNTIL X>48 AND X<54
2190 IF X=50 AND FILE=1 THEN PROCfile E
LSE IF X=50 THEN PROCin2
2200 CLS
2210 *FX112,0
2220 *FX113,0
2230 IF X=49 THEN RUN
2240 IF X=50 THEN PROCplot:PRINT"?":;0=
GET:PRINT CHR$(127):PROCopt
2250 IF X=51 THEN CHAIN "SAVESCREEN"
2260 IF X=52 THEN CHAIN"DUMP2"
2270 IF X=53 THEN VDU26:CLS:END
2280 ENDPROC
2290 :
2300 DEF PROCaxes
2310 A=1:XS%=144:YS%=10
2320 CLS
2330 VDU5:PLOT 4,XMA-550-6*YTLEN,YMI-50
:PRINT TI$
2340 IF SUPP=0 THEN PLOT 4,XMA+25,10 EL
SE IF YMIN>0 OR YMAX<0 THEN PLOT 4,XMA+2
5,YMI+10 ELSE PLOT 4,XMA+25,10
2350 PRINT XT$
2360 IF SUPP=0 THEN PLOT 4,-5-6*YTLEN,Y
MA+40 ELSE IF XMIN>0 OR XMAX<0 THEN PLOT
4,XMI-5-6*YTLEN,YMA+40 ELSE PLOT 4,-5-6
*YTLEN,YMA+40
2370 PRINT YT$
2380 PLOT 4,-25,10:PRINT"0":IF TIM=0 TH
EN PLOT 4,-6,-15:PRINT"0"
```

```
2390 IF YMIN>0 OR YMAX<0 THEN PLOT 4,XM
I,YMI:PLOT 6,XMA,YMI ELSE PLOT 4,XMI,0:P
LOT 6,XMA,0
2400 IF XMIN>0 OR XMAX<0 THEN PLOT 4,XM
I,YMI:PLOT 6,XMI,YMA ELSE PLOT 4,0,YMI:P
LOT 6,0,YMA
2410 A=1
2420 REPEAT
2430 PROCp1
2440 IF TIM=0 THEN PRINT A*XINT ELSE IF
R=1 THEN PLOT 0,20,-XI/2,0:PRINT Y(A) EL
SE IF S=1 THEN PLOT 0,120,-XI/2,0:PRINT M
TH$(A)
2450 PROCp2
2460 A=A+1
2470 UNTIL A*XINT>XMAX
2480 IF TIM=0 THEN PROCq2
2490 PROCq3:PROCq4
2500 VDU4
2510 ENDPROC
2520 :
2530 DEF PROCp1
2540 IF SUPP=0 THEN PLOT 4,A*XI-XS%, -15
ELSE IF YMIN>0 OR YMAX<0 THEN PLOT 4,A*
XI-XS%,YMI-15 ELSE PLOT 4,A*XI-XS%, -15
2550 ENDPROC
2560 :
2570 DEF PROCp2
2580 Q=0 : PLOT 4,A*XI,YMI-20
2590 IF STUB=0 THEN REPEAT:PLOT 1,0,10:
PLOT 0,0,10:Q=Q+20:UNTIL Q>YMA-YMI+40
2600 IF STUB=1 THEN IF YMIN>0 OR YMAX<0
THEN PLOT 1,0,10 ELSE PLOT 0,0,10-YMI:P
LOT 1,0,-10
2610 ENDPROC
2620 :
2630 DEF PROCp3
2640 IF SUPP=0 THEN PLOT 4,-172,YS%+(A*
YI) ELSE IF XMIN>0 OR XMAX<0 THEN PLOT 4
,XMI-172,YS%+(A*YI) ELSE PLOT 4,-172,YS%
+(A*YI)
2650 ENDPROC
2660 :
2670 DEF PROCp4
2680 Q=0:PLOT 4,XMI,A*YI
2690 IF STUB=0 THEN REPEAT:PLOT 1,10,0:
PLOT 0,10,0:Q=Q+20:UNTIL Q>XMA-XMI
2700 IF STUB=1 THEN IF XMIN>0 OR XMAX<0
```

```

THEN PLOT 1,10,0 ELSE PLOT 0,10-XMI,0:P
LOT 1,-10,0
2710 ENDPROC
2720 :
2730 DEF PROCq2
2740 A=-1
2750 REPEAT
2760 PROCp1:PRINT A*XINT
2770 PROCp2
2780 A=A-1
2790 UNTIL A*XINT<XMIN
2800 ENDPROC
2810 :
2820 DEF PROCq3
2830 A=1
2840 REPEAT
2850 PROCp3:PRINT A*YINT
2860 PROCp4
2870 A=A+1
2880 UNTIL A*YINT>YMAX+YRG/100
2890 ENDPROC
2900 :
2910 DEF PROCq4
2920 A=-1
2930 REPEAT
2940 PROCp3:PRINT A*YINT
2950 PROCp4
2960 A=A-1
2970 UNTIL A*YINT<YMIN-YRG/100
2980 ENDPROC
2990 :
3000 DEF PROC hist
3010 PLOT 4,0,0
3020 FOR A=1 TO N
3030 PLOT 102,T(A,0)*UNITX,T(A,1)*UNITY
:PLOT 4,T(A,0)*UNITX,0
3040 NEXT A
3050 ENDPROC
3060 :
3070 DEF PROCintim2
3080 REPEAT
3090 FOR A=0 TO N:PRINT"INPUT Y";A;:INP
UT " T(A,1):NEXT A
3100 PRINT"Accept these? (Y or N)":P=GE
T
3110 UNTIL P=89 OR P=121
3120 ENDPROC
3130 :

```

```

3140 DEF PROCscal:LOCAL G
3150 CLS:PRINT TAB(20,3)"XMIN=";XMIN,"
XMAX=";XMAX
3160 PRINT TAB(20,5)"YMIN=";YMIN,"
YMAX=";YMAX
3170 PRINT TAB(10,8)"Enter your selecte
d graph boundaries:"
3180 REPEAT
3190 IF TIM=0 THEN INPUT TAB(20,10)"XMI
N="XMIN,TAB(35,10)"XMAX="XMAX
3200 INPUT TAB(20,12)"YMIN="YMIN,TAB(35
,12)"YMAX="YMAX
3210 PRINT TAB(20,20)"XMIN=";XMIN,"
XMAX=";XMAX
3220 PRINT TAB(20,22)"YMIN=";YMIN,"
YMAX=";YMAX
3230 PRINT TAB(10,24)"Accept these? (Y
or N)":G=GET
3240 UNTIL G=89 OR G=121
3250 ENDPROC
3260 :
3270 DEF PROCin2
3280 PRINT""XMIN="XMIN,"XMAX="XMAX
3290 PRINT""YMIN="YMIN,"YMAX="YMAX""
3300 PRINT"N="N""
3310 IF TIM=1 THEN PROCintim2 ELSE PROC
innot2
3320 ENDPROC
3330 :
3340 DEF PROCinnot2
3350 REPEAT
3360 FOR A=1 TO N
3370 PRINT"INPUT X";A," Y"A;:INPUT" T(
A,0),T(A,1)
3380 NEXT A
3390 PRINT"Accept these? (Y or N)":P=GE
T:UNTIL P=89 OR P=121
3400 ENDPROC
3410 :
3420 DEF PROCerror
3430 IF ERR<>204 AND ERR<>222 THEN REPO
RT:PRINT" at line ";ERL:END
3440 CLS:PRINT TAB(27,10)"File not foun
d: try again"
3450 TIME=0:REPEAT UNTIL TIME=200
3460 *DIR^
3470 RUN
3480 ENDPROC

```

Personal Ads

BEEBUG members may advertise unwanted computer hardware and software through personal ads (including 'wants') in BEEBUG. These are completely free of charge but please keep your ad as short as possible. Although we will try to include all ads received, we reserve the right to edit or reject any if necessary. Any ads which cannot be accommodated in one issue will be held over to the next, so please advise us if you do not wish us to do this. We will accept adverts for software, but prospective purchasers should ensure that they always receive original copies including documentation to avoid any abuse of this facility.

We also accept members' Business Ads at the rate of 40p per word (inclusive of VAT) and these will be featured separately. Please send all ads (personal and business) to MEMBERS' ADS, BEEBUG, 117 Hatfield Road, St. Albans, Herts AL1 4JS.

Urgently Wanted: BBC Educational software for Childrens voluntary work, also add-ons, peripherals, mice, digitiser, art and DTP packages, Replay ROM, ROMs etc. even BBC machines - dead or alive. Tel. Shipley 0274 596751.

Monitor wanted for M128, colour. Tel. Newcastle-upon-Tyne 0661 871680.

Master 512 with expansion to 1Mb with B&W monitor, 40/80T 5.25" and 3.5" disc drives, Panasonic 1081 printer with spare ribbons, manuals, software, discs including Elite, all the Dabs Shareware with lots of other for the PC side of things, discs with lots of DFS and ADFS discs brimming with utilities, ROMs etc. All BEEBUG magazines to date, Dumpout III, Viewstore, FastBoot ROM, Compact and Mouse Driver Utils by Cliff Bowman, 10Mb hardDrive, lots of other stuff - to much to enumerate. Tel. Pontypridd 0443 206771.

WANTED: Genlock for BBC B. Tel. Coventry 0203 610445.

Acorn Electron with Plus 1, AP3 3.5" disc drive, View and Viewsheet, manuals and software £90. Tel. Edinburgh 031-445 1802.

BBC B issue 7, twin 40/80T Cumana disc drive with own PSU, 20Mb Winchester (Solidisk), Zif socket, WW+, Z80 2nd Processor complete with full manuals and original discs £200 or may split if offers reasonable. Tel. Oxford 0865 62688.

Master 128, boxed in perfect condition, Cumana disc drive CS400 40/80T + PSU boxed, Voltmace joystick, mouse, a good assortment of software, Super Art package and Educational Games, manuals and tape included £250 o.n.o. Tel. St Albans 0727 834060 after 5pm.

M128 fitted with BEEBUG Master ROM 05, Wordwise Plus OD, Forth 0.1, DAF 01, Comstar 2.3, Philips monitor CM11342/05G, Cumana double disc drive, mains powered, Epson printer FX80, complete set £350. Hybrid Music 5000 synthesiser, Hybrid Music 4000 keyboard, with footswitch, Hybrid Music Toolbox disc, all with User Guides and discs, "Play" tutor and disc (cost over £400) £95 o.n.o. Wordwise Plus II with segment discs and books £15, BEEBUG Masterfile II (ADFS) £10, BEEBUG Printwise disc and guide £8, BEEBUG Studio 8 disc and guide £5, Creative Graphics book with DFS disc £5, Fast Access Volume 1 (12 DFS discs in binder) £6, Advanced User Guide for the BBC £8, collect from West Surrey or transport by arrangement. Tel. Farnham 0252 710219.

WANTED: Embroidery machine for linking to a computer such as the "Poem 500 CNC", also wanted monitor such as AKF30. Tel. Leeds 0532 755276.

Quantity of new 2764, 2732 and 6265 LP devices, please ring for details with sensible offers. Tel. Lancashire 0254 703175.

Watford RAM/ROM board with 128 dynamic RAM, 16k static RAM and battery backup £35, toolkit plus £7, dustcover for BBC £2, BEEBUG magazines Vol.10 No.6 to Vol.11 No.9 free. Tel. Hampshire 0705 479082 after 5.30pm (ask for David).

WANTED: Disc copy of BEEBUGs 'Dumpmaster'. Tel. Surrey 0276 27408.

50 x 5.25" discs, some with progs, complete in box £10, 30hr Basic £4, 100 Progs for BBC, book and tape (Gordon) £4, Sureshot joystick £3, Mini Office II £3, all including postage, Triumph-Adler wide carriage daisywheel printer inc.

cable & spare daisywheel £50, buyer collects. Tel. Barnsley 0226 762450.

Ideal BBC/Master to Archimedes entry upgrade system, A310M, Med Res Philips monitor, indispensable 4Mb RAM IFEL upgrade, interfaced 5.25", twin 40/80T disc drives in monitor plinth, serial link, PC Emulator V1.7, Dr DOS 5.0, considerable amount of software, utilities and games, offers around £500 invited. Tel. Rutland 0572 821313.

BBC Master with single 40/80T Cumana disc drive, and tape recorder, Interword, Viewspell, View printer driver generator all manuals included, reference manuals. £200 o.n.o. Tel. Bath 0225 422937.

WANTED: Wapping Editor DTP ROM, with disc and manual. Tel. Bucks 0753 884360.

WANTED: Original Repton game for BBC B on 5.25" disc. Tel. Derbyshire 0773 822477.

WANTED: 1770 DFS, Spellmaster, Wapping Editor DTP. Tel. Luton 0582 413990.

WANTED: Any Epson printer, any reasonable price paid. Tel. Suffolk 0502 725016.

BBC Master 128+ Turbo co-processor with ROM software including Viewstore and Spellmaster, complete Hybrid Music system including Music 5000 synthesiser, 4000 keyboard, 3000 expander, 1000 Amplifier + pair of Panasonic speakers, Panasonic KXP-1081 printer, Microvitec Cub 452 colour monitor, Cumana 40/80T switchable 5.25" disc drive, large number of books/manuals, lots of tape/disc software thrown in, would prefer to sell complete but will split if necessary. All in excellent condition, Ideal for school music department. Originally cost £2000 - Offers? Tel. West Yorkshire 0484 654186.

Archimedes 310, 4Mb, ARM3, IDE FS 170Mb + 40Mb, Replay DIY, RISC OS 3.1 £850 o.n.o. Tel. 0727 840303 (ask for Paul extn 236).

SJ MDFS fileserver for Econet with 20Mb hard disc and matching tapestreamer and 35Mb tapes £500. Continued manufacturer's support still available. Tel. 0225 464313.

MAILER

A mailing list database that can hold a maximum of 300 to 2000 addresses, depending on disc space. Prints up to four labels per row. Allows a total of eight lines per address, and addresses can be aligned to the left or right of labels, or centralised. Can also be set up for different label sizes. Other features include sort, print as a list etc...

£8.00

Please state BBC B/Master, ADFS/DFS, 40/80T, 5.25"/3.5" disc.

1 Park Avenue,
Markfield, Leicester
LE67 9WA

**TRITON
SOFTWARE**

HINTS HINTS HINTS HINTS HINTS

and tips and tips and tips and tips and tips

FASTER BASIC

P.J. Vincent

When performing certain operations in Basic, some programming techniques are more efficient than others (timewise). Generally speaking, REPEAT-UNTIL is faster than IF-THEN-GOTO, but FOR-NEXT is faster than REPEAT-UNTIL.

A GOSUB is slightly faster than a procedure, contrary to what the User Guide suggests, and it is quicker to use global variables, again different to what the User Guide says. Turning off the ADC conversions though has very little effect upon timings.

WHICH DAY IS IT?

The short function/program below allows the day of the week to be calculated from a date entered. Note that the format required by the code below is fairly exacting; the format being DD:MM:YYYY, where DD is the date (with a zero preceding if necessary), MM likewise the month and YYYY the year. The ":" may be replaced by most characters, but it is best to standardize on just one separator.

```

100 REM DateDemo
110 DIMdays$(7):FORA%=0TO7:READday$(A%):
NEXT
120 CLS:PRINT
130 REPEAT
140 INPUT LINE"date:"A$:D%=VALA$
150 M%=VALRIGHT$(A$,7):Y%=VALRIGHT$(A$,4)
)-1600
160 PRINTdays$(FNday)"DAY"
170 UNTIL FALSE
180 END
190 DATA NO,SUN,MON,TUES,WEDNES,THURS,FR
I,SATUR
200 :
1000 DEF FNday:L%=(Y%MOD4ORY%MOV100=0ANDY%

```

```

MOD400)>0:Y%=Y%-1:M%=M%-1:=((Y%+Y%DIV4-Y%D
IV100+Y%DIV400+31*M%-(M%DIV2)-(M%=8)-(M%=1
0)+(M%>1)+(L%ANDM%>1)+D%+6)MOD7)+1AND(D%>0
ANDD%<(32-(M%MOD7MOD2)+(M%=1)+(L%ANDM%=1))
ANDM%>0ANDM%<13ANDY%>0)

```

TESTING OSBYTE ROUTINES

Jonathan Temple

Although many OSBYTE routines are not implemented as *FX calls, they can be easily tested from Basic, to see their effect, with the following function key definition. When prompted, enter the value for A (the OSBYTE call number) and the value of X and Y registers. The routine is then called and the results printed out.

```

*KEY0
I.'"A-"A%"X-"X%"Y-"Y%:R=USR&FFF4:P.'"A
=";R
A.&FF:P."X=";(R A.&FF00)/265:P."Y=";(R
A.&FF0000)/16^4|M

```

INVISIBLE BOOTS

Paul Baron

If you don't like your !BOOT file to write PAGE=&1900 and CHAIN"MENU" all over the nice clean screen, here is one method of preventing it. When *BUILDing the file, create the first line as XXXXX. Use a disc sector editor such as Disc Doctor, to alter the Xs to the hex values:

```
1C 01 07 01 07
```

These are the VDU codes to create a zero sided text window. As it is the codes themselves that are used, nothing appears on the screen. The first thing that the menu program (or whatever) should do is to change the mode or text window, otherwise all screen output will remain invisible.

B

BEEBUG MEMBERSHIP

Send applications for membership renewals, membership queries and orders for back issues to the address below. All membership fees, including overseas, should be in pounds sterling drawn (for cheques) on a UK bank. Members may also subscribe to RISC User at a special reduced rate.

RENEWAL RATES FOR BEEBUG MAGAZINE AND MAGAZINE DISC SUBSCRIPTIONS

See May 1993 Editorial for further explanation

The table below shows the renewal rate applying after the June issue 1993 according to the first issue of the renewal period. For joint BEEBUG/RISC User subscriptions add half the appropriate BEEBUG renewal rate to the full RISC User renewal rate; (UK £18.40, Europe & Eire £27.50, Middle East £33.50, Americas & Africa £36.50, Elsewhere £39.50).

Renewal Issue	Issues to go	Mag UK	Mag Europe	Mag Mid-E	Mag Am+Af	Mag Else	Disc UK	Disc O'Seas
Jun	9	16.56	24.75	30.15	32.85	35.55	45.00	50.40
Jul	8	14.72	22.00	26.80	29.20	31.60	40.00	44.80
Aug	7	12.88	19.25	23.45	25.55	27.65	35.00	39.20
Sept	6	11.04	16.50	20.10	21.90	23.70	30.00	33.60
Oct	5	9.20	13.75	16.75	18.25	19.75	25.00	28.00
Nov	4	7.36	11.00	13.40	14.60	15.80	20.00	22.40
Dec	3	5.52	8.25	10.05	10.95	11.85	15.00	16.80
J/F '94	2	3.68	5.50	6.70	7.30	7.90	10.00	11.20
Mar '94	1	1.84	2.75	3.35	3.65	3.95	5.00	5.60

BACK ISSUE PRICES (per issue)

All overseas items are sent airmail. We will accept official UK orders for subscriptions and back issues, but please note that there will be a £1 handling charge for orders under £10 which require an invoice. There is no VAT on magazines.

Volume	Magazine	5" Disc	3.5" Disc
6	£1.00	£3.00	£3.00
7	£1.10	£3.50	£4.00
8	£1.30	£4.00	£4.00
9	£1.60	£4.00	£4.75
10	£1.60	£4.75	£4.75
11	£1.90	£4.75	

POST AND PACKING

Magazines and discs are postcode a. Please add the cost of p&p when ordering. When ordering several items use the highest price code, plus half the price of each subsequent code. UK maximum £8.

Post Code	UK, BFPO Ch.1	Europe, Eire	Americas, Africa, Mid East	Elsewhere
a	£ 1.00	£ 1.60	£ 2.40	£ 2.60
b	£ 2.00	£ 3.00	£ 5.00	£ 5.50

BEEBUG
117 Hatfield Road, St. Albans, Herts AL1 4JS
Tel. St. Albans (0727) 840303, FAX: (0727) 860263

Office hours: 9am-5pm Mon-Fri Showroom hours: 9am-5pm Monday to Saturday
(24hr Answerphone for Connect/Access/Visa orders and subscriptions)

BEEBUG MAGAZINE is produced by
RISC Developments Ltd.

Editor: Mike Williams
Assistant Editor: Kristina Lucas
Editorial Assistance: Marshal Anderson
Production Assistant: Sheila Stoneman
Advertising: Sarah Shrive
Subscriptions: Helen O'Sullivan
Managing Editor: Sheridan Williams

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, RISC Developments Limited.

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £50 per page, but please give us warning of anything substantial that you intend to write. A leaflet 'Notes to Contributors' is available on receipt of an A5 (or larger) SAE.

Please submit your contributions on disc in machine readable form using plain text format if possible for text, but please ensure an adequate written description is also included of your submission and the contents/format of your disc.

In all communication, please quote your membership number.

RISC Developments Ltd (c) 1994

Printed by Arlon Printers (0923) 268328

ISSN - 0263 - 7561

Magazine Disc

January/February 1994

GRAFFER - This program provides a comprehensive graphing facility for the BBC micro covering bar charts, histograms, function plots and others. More than one type of graph can be plotted at the same time.

MINOTAUR'S MAZE - This program not only generates mazes of a size determined by the user, but ensures that the maze is actually solvable.

PUTTING ADFS DIRECTORIES TO WORK (Part 2) -

The program Search on this disc will search through any ADFS directory structure for any named file or directory giving the full pathname.

FIRST COURSE - This month's demonstration program shows how to use the BPUT, BGET and EXT commands in file handling.

TEXT CONVERSION - Two programs are provided to assist with the problems of text conversion as described in the magazine.

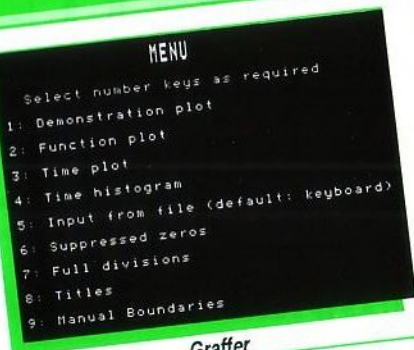
MR TOAD'S MACHINE CODE CORNER - This month's machine code demonstration shows how a printer dump can be initiated with Ctrl-Shift-@.

M-BASE (Part 3) - The magazine disc contains the complete M-Base program including part three which adds three more search options to the previous version of the program.

WORDWISE USERS NOTEBOOK - The disc contains the complete collection of Wordwise Plus programs and example files as described in the Wordwise User's Notebook in this issue.

MAGSCAN DATA - Bibliography for this issue of BEEBUG (Vol.12 No.8).

AMPLE MUSIC ITEMS - As an extra this disc contains six musical items in the form of Ample files for playing via Hybrid Technology's Music 5000. A Jukebox player program is also included. This month's pieces are by Chopin, Grieg, Heller and Mendelssohn (full details are included on disc).



M-Base

Searching directory \$ for Sub-directory
Searching directory \$.A for Sub-directory
Searching directory \$.B for Sub-directory
Searching directory \$.B.D for Sub-directory
Searching directory \$.B for Sub-directory
Searching directory \$.B.E for Sub-directory
Searching directory \$.B.E.G for Sub-directory
Searching directory \$.B.E for Sub-directory
Searching directory \$.B for Sub-directory
Searching directory \$ for Sub-directory
Searching directory \$.C for Sub-directory
Searching directory \$ for Sub-directory

Putting Directories to Work

ALL THIS FOR £4.75 (5.25" & 3.5" DISC) + £1 P&P (50p FOR EACH ADDITIONAL ITEM)
Back issues (5.25" and 3.5" discs from Vol.6 No.1) available at the same prices.

FOR DISC (5.25" or 3.5") RENEWAL SUBSCRIPTION RATES (UK AND OVERSEAS) PLEASE SEE
TABLE ON FACING PAGE

Prices are inclusive of VAT and postage as applicable. Sterling only please.

RISC Developments, 117 Hatfield Road, St.Albans, Herts AL1 4JS

STOP PRESS ... STOP PRESS ... STOP PRESS ... STOP PRESS ... STOP PRESS ...

BEEBUG

Don't miss out -
ORDER NOW!

A5000 for only £850 2Mb/80Mb HD LEARNING CURVE

Once again we have managed to obtain a number of A5000 Learning Curve systems at a ridiculously low price. Only a few are left in stock, so hurry if you are not to miss out.

The A5000 on offer is the top of the range computer from Acorn, and features the 25MHz ARM3 processor (the latest models are equipped with a 33MHz ARM3 making them marginally faster).

We are offering a range of specially priced upgrades provided they are all supplied or fitted at the same time of your order for the A5000. All are fully guaranteed for 12 months along with the computer. So whether you can only afford the computer now at £850, or a full blown system we can supply exactly what you need.

Monitor options:

Standard (AKF40) £149
MultiScan (AKF18) £179
High Res. (AKF50) £250

RAM Upgrades:

2-4Mb £94
2-8Mb £379

A5000 Specification

RISC OS 3.1
2Mb RAM
80Mb Hard drive
25MHz ARM3

Learning Curve pack

PC Emulator v1.7
DOS 5
First Word Plus
Acorn DTP
Audio Training Tape
Pacmania
Genesis Plus

Additional Hard Drives:

160Mb	£199
260Mb	£279
450Mb	£449

These drives are fitted in addition to the standard 80Mb and do not replace them.

BEEBUG Ltd

117 Hatfield Road, St Albans, Hertfordshire AL1 4JS.
Tel: 0727 840303 Telesales Direct: 0727 840305 Fax: 0727 860263
Prices shown are exclusive of £8.00 for carriage and VAT.