



IPUG

NUMBER 4
VOLUME 3
JULY 1981

INDEPENDENT PET
USERS
GROUP

COMMITTEE

Chairman: Mick Ryan

Vice Chairman: Pete Dowson

General Secretary: Eli Pamphlett,
7 Lower Green,
Tewin,
Welwyn Garden City,
Herts, AL6 0JX.
Tel: Tewin (043 871) 7325

Treasurer: Luke Gardiner

Software Librarian: Bob Wood

Membership Enquiries: IPUG, Mill Hayes House,
68 Mill Hayes Road,
Knypersley,
Stoke-on-Trent,
Staffs.

Assistant Editors: Mike Todd
Ray Hunt

Editor: Ron Geere,
109 York Road,
Farnborough,
Hants, GU14 6NQ.



INDEPENDENT

PET USERS GROUP

Vol. 3, No.4

Newsletter

JULY 1981

Europe's first independent magazine for PET users

Page	Contents
78	Editor's Notebook
78	Relative Files
80	Shop Window
82	Matters Arising
83	Time for Machine Code
85	Commodore Column
86	User's View of DOS 2.1
90	Compendium Erratum
91	Mikro Revisited
92	Words of Warning

The opinions expressed herein are those of the author and not necessarily those of IPUG or the editor. Items mentioned in "Shop Window" are culled from advertisers' material and IPUG do not necessarily endorse or recommend such items - *caveat emptor*

EDITOR'S NOTEBOOK

Coincidence is a strange phenomenon and one never knows when it will strike. In the same post came not one, but two articles on the same subject, namely DOS 2.1. I make no apologies for presenting both of these useful contributions.

As I write the pre-occasion flurry of activity associated with the PET Show is at its height. Who would have thought when three years ago PET was introduced to the UK and IPUG was formed, that it would generate so much business and spin-offs and would occasion an exhibition where virtually every single stand had something of interest. The VIC looks like being a very important addition to Commodore's armoury and when you have finished playing games with it, one can always use it as an intelligent IEEE-488/RS232C interface.

Now since IPUG is three years old, it reminds me that our Annual General Meeting is approaching and although I have some reservations, I do not intend to seek re-election for the coming year.

R.D.G.

--o0o--

RELATIVE FILES

By W.J.Northcott

Tired of waiting for BASIC 4.0 ROMs I have resorted to the following program which shows how to use relative files from BASIC 2.0. I also discovered that when using PRINT# to the command channel on the 4040, one must use ';' as a terminator, otherwise the spurious <CR/LF> generated by the PET will be interpreted as significant by the 4040 disk unit.

```

10 REM RELATIVE FILE TEST PROGRAM
11 REM RL=RECORD LENGTH SA=SEC. ADDRESS FOR RELATIVE FILE
12 REM NR=NO. OF RECORDS ON RELATIVE FILE
15 RL=50:SA=2:NR=200:NH=INT(NR/256):NL=NR-NH:RD=200
20 F$="0:RELFIL,L,"+CHR$(RL):REM FILE NAME
25 PRINT"<clr>SETTING UP FILE":T=TI
30 OPEN15,8,15:OPEN2,8,SA,F$:REM EQUIVALENT TO
    'DOPEN,FN,L(RL)'
```

```

31 INPUT#15,A,B$,C$,D$:IFA<>0THENL=31:GOTO200:REM ERROR
35 R$="P"+CHR$(SA+96)+CHR$(NL)+CHR$(NH):PRINT#15,R$;:
   PRINT#2,CHR$(255):REM EQUIV TO 'RECORD' STATEMENT
36 INPUT#15,A,B$,C$,D$:IFA<>50THENL=36:GOTO200:REM ERROR
37 R$="P"+CHR$(SA+96)+CHR$(1)+CHR$(0):PRINT#15,R$;
38 INPUT#15,A,B$,C$,D$:IFA<>0THENL=38:GOTO200:REM ERROR
39 SU=(TI-T)/60:PRINT"<clr>WRITING RECORDS":T=TI
40 FORI=1TO200:PRINT#2,"RECORD #"STR$(I)CHR$(13);
41 INPUT#15,A,B$,C$,D$:IFA<>0THENL=41:GOTO200:REM ERROR
42 NEXT
43 CLOSE2:WR=(TI-T)/60:T=TI:PRINT"<clr>READING RECORDS":
   OPEN2,8,SA,F$
44 INPUT#15,A,B$,C$,D$:IFA<>0THENL=44:GOTO200:REM ERROR
45 FORI=1TORD
50 RN=1+INT(NR*RND(1)):RH=INT(RN/256):R$="P"+CHR$(SA+96)+
   CHR$(RN-RH)+CHR$(RH)
60 PRINT#15,R$;
61 INPUT#15,A,B$,C$,D$:IFA<>0THENL=61:GOTO200:REM ERROR
70 INPUT#2,A$:PRINTRN,A$
71 INPUT#15,A,B$,C$,D$:IFA<>0THENL=71:GOTO200:REM ERROR
80 NEXT
90 CLOSE2:RT=(TI-T)/60:CLOSE15
92 PRINT"<dn>TIME TO SET UP"NR" RECORDS OF"RL" BYTES"SU"
   SECS
94 PRINT"<dn>TIME TO WRITE"NR" RECORDS"WR" SECS
96 PRINT"<dn>TIME TO READ"RD" RECORDS"RT" SECS
100 GETA$:IFA$=""THEN100
120 OPEN1,8,15,"SO:RELFIL":CLOSE1
122 END
200 PRINT"<clr>DISK ERROR
210 PRINT"<dn>NO."A" "B$" TRACK "C$" SECTOR "D$
220 PRINT"<dn>IN LINE"L:CLOSE15:CLOSE2

```

time to set up 200 records of 50 bytes 12.1666667 secs
time to write 200 records 18.1166667 secs
time to read 200 records 67.9 secs

SHOP WINDOW

Books do not normally feature in this column, but this one is by Hal Chamberlin of PET music and high-resolution graphics board fame. 'Musical Applications of Microprocessors' is published by Haydon Book Company, distributed by John Wiley. The work can be described as comprehensive and justifies the cost of £15.20.

The latest compiler to appear in the UK is the Professional Software's PC-BASIC Compiler supporting all BASIC 2 commands and all CBM peripherals. The compiler is stored on CBM 3040, 4040 or 8050 disk and is used in conjunction with a run-time package in EPROM. Further details are available and the compiler for £300+VAT from Intex Datalog Ltd, Eaglescliffe Industrial Estate, Eaglescliffe, Stockton-on-Tees, Cleveland, TS16 0PN. Tel: (0642) 781193.

Now BASIC 4 can be enhanced with the BIZCHIP which adds 21 new commands. The chip is Toolkit-compatible and fits 8032, 4032 or 3032 with BASIC 4 upgrade. Commands include advanced string handling, 'unBREAKable' INPUT which allows commas etc., numeric validation, fast disk access, stop key enable/disable. BIZCHIP and its manual cost £35.00 (state \$9000 or \$A000 slot) from Fine Curtis & Gross, Room 3, Hamilton House, Mabledon Place, London, WC1H 9BD. Tel: 01-388 7345/6.

A sound-effects generator board programmed with PEEK/POKE/SYS/USR commands can be obtained for the PET (also Nascom, UK101 and others). The sound chip is the ubiquitous AY-3-8910 and the board is available for both old- and new-ROM style machines. The unit contains a small amplifier and speaker and costs only £43.00+VAT. Contact Easicomp Ltd., 57, Parana Court, Sprowston, Norwich, NR7 8BH. They also market an ultra-fast 16-bit A-D converter.

Described as the 'ultimate software development tool'

the screen-based machine-code writer/disassembler from Davidson-Richards Ltd. comes with optional EPROM loader. The software requires 16/32K with version 3 or 4 ROM. Software on disk or tape is £75.00 including p & p and VAT. The software with EPROM programmer is £125.00 inclusive. Davidson-Richards are at 14, Duffield Road, Derby, DE1 3BB. State machine details when ordering.

INSTANT ROM is 2K or 4K of CMOS memory with battery back-up. Just treat it as ordinary addressable RAM. Locations are easily modified and data is retained for years. For details contact Greenwich Instruments Ltd., 22, Bardsley Lane, Greenwich, London, SE10 9RF. Tel: 01-853 0868.

The Ventrol Petalker is a vocal synthesiser for the PET at £138.00 including speaker, power supply and VAT. Complete with sample program and instructions the unit's vocabulary covers alphanumeric (it says here !). Details from Ventrol Electronics Ltd., 472, Halifax Road, Rochdale, Lancs, RL1 2RD. Tel: Rochdale 41970.

TECPACs are a collection of professional software for designers, engineers and scientists and are a Commodore Approved product. Contact the Technical Software Centre, BHRA Fluid Engineering, Cranfield, Bedford, MK43 0AJ. Tel: (0234 750 102).

Small Systems Engineering have now come up with a 'stunt box' which will enable PET to run under the CPM operating system opening up the way for Cobol, Fortran, Algol, Coral-66, Lisp and APL languages. The hardware costs £550 and connects to the IEEE port. SME also are introducing a 22M-byte hard disk unit at about £4,500. The mind boggles....

R.D.G.

MATTERS ARISING

Owing to a printing error part of the RH-side of p71 was masked off and did not reproduce. Owing to demand for the missing information I have decided to reproduce the DATA values again:

```

700 REM MACHINE CODE DATA
705 DATA634,865
710 DATA162,1,32,207,2,162,2,32,207,2,169,1,133,58,32,229
715 DATA2,169,4,133,59,32,229,2,162,1,32,240,2,162,2,32
720 DATA240,2,162,0,134,186,32,225,255,162,2,173,3,34,205
,3
725 DATA33,144,23,208,19,201,255,240,40,173,2,34,205,2,33
,144
730 DATA9,208,5,230,186,32,240,2,162,1,32,42,3,166,189,32
735 DATA240,2,76,160,2,32,112,247,32,225,241,32,225,241,
76,204
740 DATA255,169,0,32,229,2,169,0,76,229,2,162,3,32,188,
247
745 DATA32,50,242,76,204,255,138,133,187,9,32,133,63,169,
0,133
750 DATA62,32,112,247,160,0,132,188,32,225,241,164,188,
145,62,200
755 DATA166,150,208,10,192,5,144,238,170,240,9,152,16,232
,169,255
760 DATA160,3,145,62,32,204,255,166,187,164,188,200,152,
149,59,96
765 DATA138,133,189,9,32,133,63,169,0,133,62,181,59,133,
190,24
770 DATA101,58,133,58,160,0,145,62,144,2,230,59,165,59,
200,145
775 DATA62,162,3,32,188,247,136,132,191,177,62,32,50,242,
164,191
780 DATA200,196,190,144,242,76,204,255

```

I should add that the routine is for BASIC 2.0 only. A BASIC 4.0 version is on its way, plus a routine to merge from disk to memory, leaving the result in memory.

TIME FOR SOME MACHINE CODE

The following routines give some pointers in using the jiffy clock in machine code. If you simply want HMMSS then do JSR READCLK then JSR CLKPRINT.

```

1000 ;*****
1020 ;* ROUTINE TO RESET THE CLOCK TO *
1040 ;* ZERO. ROUTINES ARE BASIC 2.0. *
1060 ;*****
1080 ;
1100 TIMEPTR=$1F
1120 TIMESET=$C8EB
1140 ;
1160 ZEROCLK LDA #<ZERO
1180 LDY #>ZERO ;POINT TO
1200 STA TIMEPTR
1220 STY TIMEPTR+1 ;TIME REQUIRED
1240 LDA #6
1260 JSR TIMESET ;SET THE TIME
1280 RTS
1300 ;
1320 ZERO TXT "000000"
1340 ;
1360 ;*****
1380 ;* ROUTINE TO PRINT THE TIME AS MM:SS *
1400 ;*****
1420 ;
1440 PRINT=$E3D8
1460 STROUT=$CA1C
1480 XCOORD=$C6
1500 YCOORD=$D8
1520 SETCSR=$E25D
1540 CLKDESCR=$61
1560 CLKPRINT=$CA1F
1580 READCLK=$CE2E
1600 ;
1620 LDA XCOORD ;SAVE CSR POSN
1640 PHA
1660 LDA YCOORD
1680 PHA
1700 ;

```

```

1720 ;SET NEW CURSOR COORDINATES
1740 ;
1760 LDA #10
1780 LDY #3
1800 STA XCOORD
1820 STY YCOORD
1840 JSR SETCSR
1860 ;
1880 ;READ & PRINT THE TIME
1900 ;
1920 LDA #<CLKHEADER
1940 LDY #>CLKHEADER
1960 JSR STROUT ;PRINT HEADER
1980 JSR READCLK ;FIND THE TIME
2000 JSR DISPTWO ;PRINT TWO DIGITS
2020 LDA #45
2040 JSR PRINT ;PRINT SEPARATOR
2060 JSR DISPTWO ;PRINT TWO DIGITS
2080 ;
2100 PLA ;RESTORE CSR POSN
2120 STA YCOORD
2140 PLA
2160 STA XCOORD
2180 JSR SETCSR
2200 RTS
2220 ;
2240 ;*****
2260 * SUBROUTINE TO MAKE THE CLOCK .*
2280 ;* STRING DESCRIPTOR POINT TWO- .*
2300 ;* BYTES ONWARD THEN APPEAR AS A *
2320 ;* LENGTH OF TWO BYTES AND PRINT *
2340 ;* THE SUBSTRING SO DEFINED . . .*
2360 ;*****
2380 ;
2400 DISPTWO LDY #0
2420 LDA #2
2440 STA (CLKDESCR),Y ;SET STRING LENGTH
2460 INY
2480 CLC
2500 ADC (CLKDESCR),Y ;ADVANCE TWO
2520 STA (CLKDESCR),Y ;CHARACTERS

```

```

2540          INY
2560          LDA (CLKDESCR),Y ;CARRY OVER
2580          ADC #0 ;IF REQUIRED
2600          STA (CLKDESCR),Y
2620          JSR CLKPRINT ;DISPLAY THE TIME
2640          RTS
2660          ;
2680  CLKHEADER  TXT "EVENT TIME :<rvs>"
2700          BYT 0
2720          ;

```

R.D.G.

---o0o---

COMMODORE COLUMN

A recent visit to Commodore at Slough provided some useful information. Some of you may have been awaiting the -07 printer ROM. The news is that it has bugs in it (surprise !) and is not going to be released. BASIC 4.0 with its big 'fast garbage collect' selling point has a bug associated with the process such 'bottom of strings' pointer moves down to eventually cause '?OUT OF MEMORY ERROR' instead of doing a 'collect'. The circumstances under which this bug manifests itself are somewhat obscure. If concatenating three strings results in insufficient space, then garbage collect fails.

The Commodore office at Euston Road, has finally closed and is now used by Sumlock Bondain Ltd.

The new Micromainframe, the MMF9000, with its 96K virtual memory and 36K ROM, will use the Motorola 6809 processor chip in addition to a 6502. An intelligent operating system will give full compatibility with Commodore software and peripherals. Free languages are included in the below - £2000 price.

Out of the blue comes yet another PET variation - sort of 40-column 8032. The lower half is an 8032 model, but the upper is 4032, as an approximate description.

---o0o---

A USERS VIEW OF DOS2.1

By B. D. Grainger

I have for some weeks been using DOS2.1 in my 2040 disk unit. The purpose of this article is to give my impressions of the new disk ROMs as a guide to those considering purchase. I should explain that I have an original 8K PET expanded to 32K fitted with BASIC3 ROMs. As I do not have expansion ROM sockets I have used a customised version of Calco Software's DISK-O-PRO, running in RAM to test the BASIC4 capabilities of the new DOS2.1.

The first thing one notices is that on turning on the disk unit the drives revolve for a period. Rather irritating this as one cannot immediately put the disks in the drive. Perhaps Commodore could explain the purpose of this annoying feature. However this is the only grouse I have. Without BASIC4 capabilities the DOS2.1 adds two new capabilities. Firstly you no longer have to initialise disks which is more useful than might be thought. Secondly one can copy a complete disk by e.g. for copying drive 0 to drive 1:

```
OPEN15,8,15:PRINT#15,"C1=0":CLOSE15
```

This is extremely useful when you are copying somebody else's disk! More importantly it provides a quick way of copying a DOS1 disk to a DOS2.1 formatted disk. This is necessary because DOS2.1 will not write to a DOS1 disk (it generates a wrong format error). DOS2.1 will read a DOS1 disk and vice versa.

The second noticeable thing is on reading the disk directory one notices an extra 2A on the header. This indicates a DOS2.1 formatted disk. DOS1 disks have a 1 and disks formatted on the 8050 have 2C. So why didn't they call it DOS2A I ask myself. To make life confusing I suppose! It should be said here that an 8050 disk cannot be used with DOS2.1 and vice versa.

Well, what about the BASIC4 capabilities. Firstly one has a completely new type of file available—Relative Record. This is a file of fixed length lines (records) and

one can point to any position on the file before reading or writing. e.g. to position to the 5th byte of the 9th record and then read up to the first 'carriage return' one would type:

```
RECORD#1,9,5:INPUT#1,A$
```

Relative record files can be opened for reading and writing at the same time so one can amend a file without writing to another file and then copying back. Obviously this is a great improvement on sequential access but if a true indexed file is required one will still need to hold the index in another file. A disk access will then be made by consulting the index, pointing to the correct record and byte and then accessing the relative record file.

The other BASIC4 capabilities are just more readable and memorable alternatives to the usual commands. e.g.:

```
COPY 0,"FILE1" TO 1,"FILE2"
```

is equivalent to:

```
OPEN15,8,15:PRINT#15,"C1:FILE2=0:FILE1":CLOSE15
```

I found this capability more useful than I thought and I particularly liked the handling of SCRATCH. Firstly, in direct mode you have to confirm a scratch is required and with my DISK-0-PRO at least, it tells you how many files are scratched after the command is executed. I'm not sure if this happens with BASIC4 but it is very useful.

One of my worries before fitting DOS2.1 was whether I would make useless all the programs and utilities I have got which access disks. In fact this stopped me fitting DOS2.1 for some period. I need not have worried. I have only found one problem. Jim Butterfield's DISK RENAME utility did not work. The solution was simple. Mount the disk to be renamed in drive 1 instead of drive 0 and modify the program to send the disk commands to drive 1. From this I deduce that DOS2.1 stores the drive 1 directory in RAM where DOS1 stored the drive 0 directory. Perhaps somebody could confirm. [Yes, this is so - Ed].

Finally, one must mention the bugs. With retrofitted 2040 or 3040 disk units one must use U1 and U2 instead of B-R and B-W. Secondly, one is advised by Commodore not to open 2 channels for writing to the same relative record

file. This is not a bug but a practical restriction. I have heard that one cannot open 2 relative record files at the same time. If this is true it must represent a serious problem for business programs. I have also heard that only 8 relative record files are allowed on a disk. This reminds me that compared to the DOS1 there is slightly less storage space on a disk (6 blocks less) and one can only have 144 files on a Format 2A disk (155 for Format 1).

One cannot review a Commodore product without mentioning the instruction manual. My DOS2.1 came with a new disk manual which was considerably better than that with my 2040 disk. It even indicates where directory info is stored on the disk. One minor point. It would help to page numbers given against the entries on the contents page !

DOS 2 from Mike Todd

Many users of the Commodore disk drives may be contemplating upgrading to DOS2 in order to have the additional facilities that this makes available. As part of that decision, the user will have to decide whether or not to upgrade the computer at the same time and install BASIC4 and have all the DOS2 facilities at his fingertips. At first sight it seems an obvious choice, however what is not generally known is that all DOS2 facilities (including the full random access facilities) are available using BASIC2/3 provided that the user is prepared to use rather long winded commands. The only command not fully implementable is DIRECTORY which is a unique function of BASIC4 and not of DOS2. Similarly the disk status variables DS and DS\$ are not available.

For the discussion that follows, it is assumed that the command channel has been opened using OPEN1,8,15. It might also be pointed out that by using these BASIC2/3 formats for DOS2 access allows compatibility with BASIC4 since all these formats are valid in BASIC4 - it's just that there are easier ways of implementing them!

DLOAD and DSAVE are the same as LOAD and SAVE in BASIC2/3 except that the normal BASIC2/3 syntax must be used. Similarly DOPEN and DCLOSE are the same as OPEN and CLOSE - although DCLOSE without any file number will close all active files, CLOSE without any file number will result in a syntax error.

Other obvious command translations are HEADER, which is the same as PRINT#1,"N...."; COLLECT which is the same as PRINT#1,"V...."; BACKUP which is the same as PRINT#1,"D...."; RENAME is the same as PRINT#1,"R...."; SCRATCH is the same as PRINT#1,"S...." and COPY which is the same as PRINT#1,"C....". I will not elaborate on these commands as they are self explanatory to anyone already using DOS1.

However, DOS2 adds a few extra facilities such as the APPEND command which opens a sequential file for writing and writes the new data on to the end of the existing file. This is easy to implement with BASIC2/3 it just requires the OPEN filename to be in the format: "filename,A" which performs the same function as the APPEND command in BASIC4. Note that BASIC4 does not require the channel number to be specified in the DOPEN or APPEND commands whereas the user will have to supply these in the normal way if using BASIC 2/3.

One of the main advantages of DOS2 is that it allows decent random access files to be created and this is normally accessed in BASIC4 using DOPEN#x,"filename",Ly where L indicates record length specifier and y the actual record length. When DOS2 sees this command it automatically opens the necessary random access file. BASIC2/3 can do the same thing by using "drive:filename,L,"+CHR\$(y) as the filename in the OPEN command.

Once the random access file is opened, it is necessary to select the record required and this is done using the RECORD command. BASIC2/3 can do this through the command channel as follows PRINT#1,"P";CHR\$(c);CHR\$(Lo);CHR\$(hi) and this will select record number $hi*256+lo$ (ie lo and hi are the low and high bytes of the actual record number) and c is the channel number allocated to the random access

file. Note that DOS2 also allows the selection of a specific byte within a record by the addition of the byte position in the RECORD command. BASIC2/3 can do this by adding;;;CHR\$(b) to the command above, where b is the byte position from which you wish to read data.

The other command is the CONCAT command. This is also implementable in BASIC2/3 using the normal COPY format as follows:

```
CONCAT D0,"NAME1" TO D1,"NAME2" becomes
PRINT#1,"C1:NAME2=1:NAME2,0:NAME1.
```

That then is how to use DOS2 commands from a BASIC2/3 environment. However, there is one DOS2 facility that has not been publicised, but which is available for use by BASIC2/3/4 users. It can only be accessed using the format described below. I might add that this command also exists in DOS1 although I have never been able to check its reliability either in DOS1 or DOS2.

If you are writing to a sequential file and then accidentally or purposefully open the same file as a READ file you will get the error light and error #60 (WRITE FILE OPEN). However, if you open the READ file using the format: "filename,M" for the filename in the OPEN command, the DOS will allow you to read back the file at the same time as you are writing to it. I have tried it on several occasions and have found that it even reads back data not yet actually written to the disk and only sitting in one of the buffers. As yet, however, I have been unable to find a use for it!

--o0o--

COMPENDIUM ERRATUM

While most users of the IPUG Compendium extoll its virtues, nobody seems to have noticed that the ROM routines to transfer between the two floating point accumulators are transposed, i.e. DB08 is #2 to #1, DB18 is #1 to #2. In addition \$C46F <C468> exits with X=FF, Y=01.

--o0o--

MIKRO REVISITED

Since reviewing the MIKRO assembler chip last issue I have spent a considerable time using it for an extensive machine-code program. In consequence a number of discoveries have been made that may be worthy of note.

MIKRO uses a space as a delimiter and therefore labels cannot contain an embedded space. In addition a label may not contain an embedded 'GO' if it follows a relative branch because BNE GO<line number> is legal, whereas BNE GOBACK is not, 'BACK' not being numeric.

It took some while to fathom this one: COUNT BYT 0;;;COMMENT results in NO OP-CODE ERROR whereas BYT 0 ;COMMENT gives ILLEGAL LABEL ERROR. The 'error' is that comment fields are illegal with pseudo-ops.

One can define quantities such as VALUE=2, but not NULL=0. Negative values are also inadmissible.

Unless defined to the contrary, code is assembled at \$033A up, so that if your program is long enough it will overlap \$0400 (1024). Invoking the !A command to start assembly produces the message: OBJECT CODE WILL CORRUPT TEXT ABORT (Y/N)? and if No, your previously saved text (you did save it, didn't you?) will be corrupted as assembly progresses. It also transpires that since MIKRO assembles code in high memory (\$7000-up for 32K machines) and your code intrudes upon this, then all is corrupt. For a while I used the Programmers' Friend in lieu of the Toolkit with MIKRO, but I experienced a system crash with a long program and conclude that assembly in this high memory region had corrupted the 'Friend' (long here means a module greater than 328 bytes).

The start and end addresses for your code that are printed when assembly is complete are those to use to save your object code with the monitor, i.e. end+1. During assembly the keyboard is inactive and pressing the <STOP> key has no effect. On a long program one is never sure if

the process is continuing or the PET has 'hung' until either assembly is complete or stops with an error message (the Toolkit's HELP is far more useful here than for BASIC programs). If a printout is requested, a CBM printer will skip perforations (if aligned first), but if long comments force a new line, each 'extra' line upsets the form-feed counting.

MIKRO will treat a '?' in a comment field as a token and it will appear as PRINT in your listings. The chip does not treat the comment field like a REM statement, even though the code to do so could have been simply included. If your comment starts with REM or quotes (") it will prevent this.

I have yet to assemble more code than can be accommodated in the reserved space at top of memory all in one go, but I suspect that something dodgy will happen when the time comes.

MIKRO does not print a symbol table and after some investigation I discovered why - it doesn't create one! The chip works like an interpreter as opposed to a compiler. It creates code line by line. When a label is encountered, the whole source text is searched for it. This explains why there is no reasonable limit to the label length.

Finally, each chip has a unique serial number embedded therein and Supersoft keep a record of everyone dispatched. Not a disk file, but meticulously entered in a ledger with a quill, so rumour has it.

R.D.G.

---o0o---

WORDS OF WARNING

Despite a variety of advertisements, and being an Approved Product, no publication appears to mention that there are three versions of the KC Netkit, and that the appropriate version must be specified when ordering: 3000, 4000, or 8000 series.

---o0o---



PHOTOCOPIED

Printed, published and distributed by Carlton Press, 10 Bernard Road, Gorleston,
Gt. Yarmouth, Norfolk NR31 6EG, England, UK. Tel: Gt. Yarmouth (STD 0493) 61331