

NATGUG NEWS

Volume 9, Issue 5 November 1987

```

new date (mm dd yy): 6-
format date
new date (mm dd yy): 6-19 87
format time is 7:26:33.00
new time: 7:26:33.00
DIP M
diskette in drive A has no label
directory of A \
.
DIRWIN SYS DRIVER SYS
DIRMHI EYE ASSIGN COM
DIRPAHL COM CHKDSK EYE
DIRPDI EYE DIRM EYE
DIRPKI EYE FORMAT EYE
DIRPLI COM LABEL EYE
DIRSBI SYS RECOVER EYE
DIRSICI EYE SORT EYE
DIRVII COM BASIC EYE
          COM KEYTR COM
          SU File(s)      222208 bytes free
          RANDISK SYS
          APPEND EYE
          DEBIC COM
          EXE2BIN EYE
          MODE EYE
          REPLACE EYE
          SYS EYE
          KEYSYP COM
          KEYTFD COM
          SSIT COM
          SELECT COM
          DISKCOMP EYE
          FIND EYE
          GRAPHICS EYE
          TRIE COM
          MORE COM
          RESTORE COM
          TIE EYE
          NEYFR EYE
          LPDRVR SYS
          COMMAND COM
          PATCH COM
          DISKCOPY EYE
          PC EYE
          JOIN EYE
          PRINT EYE
          SHARE EYE
          XCOPY EYE
          KEXTUN EYE
          LF COM

```

OFFICIAL JOURNAL OF THE

***National Amstrad, Tandy
&
General User Group***

INFORMATION ON THE GROUP

Membership of the group is by subscription to the Newsletter which is published at regular intervals – application forms are available from the secretary. Membership is open to anyone with an interest in computers but special emphasis is placed on equipment within the Amstrad, Tandy and MS-DOS range.

Details of the group's accounts and constitution are available from the Treasurer – please ensure that your requests are accompanied by a S.A.E.

Members requiring assistance with problems related to the machines specified should contact the P.R. Officer who will endeavour to put them in touch with possible advisors.

Sub-groups exist in many areas and their secretaries are invited to forward details to our Editor/Publisher for inclusion in the magazine. The back page is being reserved for this purpose.

Public domain software libraries are maintained in five separate collections : Model 1, Model 4, CP/M, Amstrad and MS-DOS. Names of the appropriate librarians are available from the secretary. There is a copying charge of 1-00 per disk or tape. (see also Vol.8, Iss.10)

Back numbers of the magazine, in 6 month volumes, are available at the price indicated on the application forms.

The group has no paid Officers or employees, and the issue of the magazine depends on contributions from Members, who are also invited to submit responses to questions raised in the previous issue. To allow legible print, we prefer contributions to be submitted on 5.25" disk, direct to the Editor – ASCII files are perfectly acceptable but please indicate the disk format used (SS,DS,SD,DD, track count, DOS etc.). Your disk will be returned if you enclose an addressed label, normally within 7 days. The Editor will accept written or typed articles where members insist - publishing will depend on readability.

Newsletter Editor/Publisher	Secretary	Treasurer	P.R. Officer
Gordon Collins 11 Elizabeth Road Sutton Coldfield West Midlands B73 5AR. 021-354-3299	David Washford 6 Houston Way Frome BA11 3EU (0373) 72739	Roger Storrs Oakfield Lodge Ram Hill Coalpit Heath Bristol BS17 2TY (0454) 772920	John Kilpatrick 3a Gainsborough Street Sudbury Suffolk CO10 6ET (0787) 79504

CONTENTS

Editorial	3
Sexnotes	6
A Cobol Data Base	7
Hypercube Arrays	22
Using the Model 4's 64K Memory from BASIC	24
Review of XTREE Version 2.00	29
Using the Microsoft Mouse with Supercalc 4	33
Reader's Forum	34
MSDOS Monthly - by Dave Holman	38
Using SYS13/SYS	39
Lower Case Mode for the Model I	42
Local Club News (& answers required please)	44

EDITORIAL

Thank you to all the people who have sent in articles, they are still coming in regularly (though not so many this last month). I have to admit to having not included three articles which I had when I sent the copy out for Vol. 9, Iss. 4, they are included in this copy. This was due to two peculiar reasons.

I called into the PCW show at Olympia as I was passing on the Friday. Our PR Officer was manning NATGUG Section of the stand. Our thanks go to John for all the work he put in there for six days and evenings, we are fortunate to have people like John who are willing and able to volunteer to do this type of work for us. Our thanks also to Leon who volunteered to have us on a stand which he had booked for QUANTA. John also used the opportunity to AGAIN try and have the entries for NATGUG that are on various data bases, corrected. These include PRESTEL and ACC. The committee thank all the people who have pointed out these to us. We have tried a number of times to have them corrected, but as someone else has already said on these pages, "There is no such thing as bad advertising." A NATGUG stand has been booked for next years show, it is not too early to let the committee know if you are prepared to spend some time manning the stand please.

I got very angry the other day when a member phoned to tell me what has happened to him. The story briefly is that, when LDOS 5:3 was announced, the member ordered, in good faith, from Molimerx, this DOS. The disk arrived with a note saying that this was a temporary product. Many phone calls have been made to Molimerx, as the member wished to have a 'permanent' product. As Molimerx has now ceased trading, he wrote to Misosys. The outcome is that according to Misosys, the LDOS 5:3 with an serial number of TCOA0020, sold to the member by Molimerx, is a COPY of a disk sold to someone in Canada. (Yes they can check on where each copy was sold to), Roy Soltoff, of Misosys is, understandably, also not very happy. Mr. John Harding if you read this and wish to reply to me, as we do not seem to be able to contact you, I will make space available in this Newsletter for your reply. Does any member know how Mr. John Harding can be contacted please? We do know of the PO Box 10, address at Bexhill. It appears that Mr. Harding is now ex-directory. If any Member who has purchased LDOS 5:3, and it has

the serial number as given above, would they please write to me giving details.

UPDATE. The following is in addition to the above, and is an item that I understand that will appear in the next Misosys Quarterly, which is in response to a customers question.

Molimerx and LDOS 5.3

"Fm MISOSYS: Molimerx no longer is involved with the TRS-80 software market, to my knowledge. Molimerx was also never authorized to distribute LDOS 5.3 nor were any copies ever sold to them by MISOSYS. We are the only suppliers of LDOS 5.3. The evidence we have pinpoints the copies of LDOS 5.3 which Molimerx was (or is) selling are copies of the LDOS 5.3 upgrade kit we sold to a Canadian company. That serial number was TCOA0020 and was registered to J&J Electronics Ltd. There was never any authorization for that nor has MISOSYS received any funds for those unauthorized copies sold by Molimerx. Funny how dozens of copies of a disk we sold to a Canadian company have wound up in Great Britain. Such is life ! And folks still complain about me not wasting my time on a Model I release !!!"

Some Members who send in articles are spending a lot of time formatting those articles. Rightly or wrongly, as in ignorance, it may be me doing something that I need not do, I find that to have a standard format that will fit into the pages of NATGUG News, I have to reformat. I have been using a margin of zero and a line length of 71. This was found to be the most economical format, consistent with readability. This could now change a little, if we find that a dot matrix printer output is acceptable, this will be used with 17 c.p.i., so as to get screen dumps (80 chrs/line), for example, across a page without having to chop them about. I have found out that I can control the pitch of the line feeds on the printer, so that the pages towards the back of the last issue were printed with 48 lines per page in the same page length, instead of 43, as I had been using. So the message is, Please do not format, just send in the ASCII File. There have been three members who have sent in LeScript files, those have been easy for me to reformat. If I do not put all articles through the same process I cannot number the pages as you have seen them up to now, the only other way is to cut-and-paste.

Thank you to those members who have answered my request to share the answers to other members problems through the pages of NATGUG News, this does benefit the whole of NATGUG.

UPDATE - Advertising. We are still pursuing selling advertising space in NATGUG News, Tandy has agreed, and we should have their artwork shortly. Does any Member know of anyone else we can approach. The payments received from advertising save the committee from increasing subscriptions. Get enough advertising and the committee could (maybe) cut the membership subscription to a nominal amount.

Did you have a look at the label which was stuck on the envelope bringing your Newsletter ? This now includes a date to advise YOU when YOUR membership subscription expires, if you do not agree with this date, Roger will be very happy to receive your written reason why. DO NOT THROW AWAY THE ENVELOPE WITHOUT FIRST CHECKING THIS DATE, NATGUG would not like to loose you because we had not given you the opportunity to renew your membership in time.

UPDATE - TANDY. I have had a further meeting at Tandy [U.K.], this was to take the forms to be given out from Tandy Stores telling about NATGUG. Also, to agree about certain arrangements and to discuss new points. Those who attended Swindon will have seen Tandy's representatives there.

UPDATE - 80 MICRO. We have now had two letters from 80 Micro. One asked us for all details about NATGUG, as they are going to run a listing of user groups in their December issue, and second we have been advised to apply to become dealers for 80 Micro. More on this when we have made further enquiries.

Those Members who use LS-DOS 6.3 might like to read an article starting at page 101 of the August '87 issue of 80 Micro.

If any Member has any suggestions, such as alternate venues, or on any other theme, and would like your committee to follow it up, please get in touch with one of us.

Well done to John with his page 56.75 in last months NATGUG News, I don't know who the character was with the hammer raised above his

head, must be another clanger - due to sound. Could this be my reward after SILENTLY pointing out to him, one of his errors.

I hope all those who attended the Swindon Workshop enjoyed themselves. Some people left their booking to late and found the hotel fully booked, the moral is - BOOK EARLY.

Gordon Collins, Editor. **

SEXNOTES

Short and sweet this month since a) we've promised that Officers won't take up too much of your space, and b) I'm having to write this BEFORE Swindon, when so much News is likely to break.

Have you seen that Tandy stores are now selling the Spectrum +2 ? I can assure you that we are NOT likely to change our name to cater for this - after all, lack of care might turn us into NASTyGUG !!! Anyway, haven't a couple of Members already set up a User Group for Sinclair ?

More seriously, three people so far have asked me to explain the IGEL shown in my recent Newdos BASIC listings. Since this would probably occupy at least three magazine pages I would like to know how many members are interested, since it might serve your interests better for me to write to them individually and leave the magazine free for other articles. Once again, I ask you to let me know.

And related to that plea, several of you know that I make great use of an ansaphone. Please, don't be afraid of it - just leave your name and address (since replying by post is much cheaper than by phone) and tell the machine what your query is - it's a very good listener. I get really upset when I come home and the machine says it had 8 calls, but 7 of them just hung up !

Happy Computing - hope that I'll have met a lot more of you at this Swindon meeting.

David Washford, 6 Houston Way, FROME, BA11 3EU 0373 72739

A Cobol Data Base.

=====

Last week I took a couple of days out to write the following data base program. I do not like those Data Base programming languages (Such as dBase II) as they usually don't do quite what I want them to. Hence I usually write my own program for a data base.

This program was designed for me to keep record's of Job applications that I am sending out. It consists of two programs:

CREATE/CBL Is the program to Create and Initialize the Data file that I am going to use to store the data.

LOOK/CBL Is the program to Access the Data File.

I record name and address of the company, along with the job name. I then store four dates concerned with the application. The date I send for the application form. The date I send in the application form. The date I receive a reply from the company (Some don't even bother to reply). Finally the date of my Interview with the company (If I get that far).

Now for anybody who has done an elementary course in Cobol, to alter these programs to your own ends is quite simple. Indeed for anybody who has no knowledge of Cobol then it should not provide to great a headache. One note is that Full Stops are VERY important in Cobol, do not add or remove one it could change the flow of the program !

I am printing this for two reasons:-

- a) This program has taken me several days to write and I think it might be useful to other members (well maybe not you but your family).
- b) It uses the Indexed Random file access mode in addition to the printer. Many people find using a Indexed file in Cobol confusing, and I certainly find using the Printer from Tandy Cobol very confusing. So I am using this program to demonstrate programming ideas for Cobol people.

If you (the Reader) would like me to do a quick course in cobol for you, then ask the Editor and we will see what I can do.

000100 IDENTIFICATION DIVISION.

000110 PROGRAM-ID. CREATE-DATA-FILE.

000120 AUTHOR. PETER KNAGGS.

000130 DATE-WRITTEN. 10/07/87.
000140
000150 ENVIRONMENT DIVISION.
000160 CONFIGURATION SECTION.
000170 SOURCE-COMPUTER. TRS-80-MODEL-3.
000180 OBJECT-COMPUTER. TRS-80-MODEL-3.
000190 INPUT-OUTPUT SECTION.
000200 FILE-CONTROL.
000210 SELECT MASTER-DISC-FILE
000220 ASSIGN TO RANDOM, "DATAFILE/APL"
000230 ORGANIZATION IS INDEXED
000240 ACCESS MODE IS DYNAMIC
000250 RECORD KEY IS MASTER-KEY.
000260
000270 DATA DIVISION.
000280 FILE SECTION.
000290 FD MASTER-DISC-FILE
000300 LABEL RECORDS ARE STANDARD
000310 DATA RECORD IS MASTER-REC.
000320
000330 01 MASTER-REC.
000340 02 MASTER-KEY PIC X(20).
000350 02 ADDRESS.
000360 03 ADDR-1 PIC X(20).
000370 03 ADDR-2 PIC X(20).
000380 03 ADDR-3 PIC X(20).
000390 03 COUNTY PIC X(10).
000400 03 POST-CODE PIC X(10).
000410 02 JOB-NAME PIC X(20).
000420 02 DATES.
000430 03 LETTE PIC X(8).
000440 03 APPL PIC X(8).
000450 03 REPLY PIC X(8).
000460 03 INTER PIC X(8).
000470 02 JOB.
000480 03 OFFER PIC X.
000490 03 TAKE PIC X.
000500
000510 WORKING-STORAGE SECTION.
000520
000530 PROCEDURE DIVISION.
000540 START-UP.
000550 OPEN OUTPUT MASTER-DISC-FILE
000560 MOVE SPACES TO MASTER-REC.
000570 WRITE MASTER-REC.
000580 CLOSE MASTER-DISC-FILE
000590 STOP RUN.
000600 END PROGRAM.

000100 IDENTIFICATION DIVISION.
000110 PROGRAM-ID. JOB-APPLICATIONS-DATA-BASE.
000120 AUTHOR. Peter J Knaggs.
000130 DATE-WRITTEN. 8/7/87-10/7/87.
000140
000150* Menu Driven data base system
000160
000170 ENVIRONMENT DIVISION.
000180 CONFIGURATION SECTION.
000190 SOURCE-COMPUTER. TRS-80-MODEL-4.
000200 OBJECT-COMPUTER. TRS-80-MODEL-4.
000210 INPUT-OUTPUT SECTION.
000220 FILE-CONTROL.
000230 SELECT MASTER-DISC-FILE
000240 ASSIGN TO RANDOM, "DATAFILE/APL"
000250 ORGANIZATION IS INDEXED
000260 ACCESS MODE IS DYNAMIC
000270 RECORD KEY IS MASTER-KEY.
000280 SELECT PRINTER
000290 ASSIGN TO PRINT, "PRINTER".
000300
000310 DATA DIVISION.
000320 FILE SECTION.
000330 FD MASTER-DISC-FILE
000340 LABEL RECORDS ARE STANDARD
000350 DATA RECORD IS MASTER-REC.
000360
000370 01 MASTER-REC.
000380 02 MASTER-KEY PIC X(20).
000390 02 ADDRESS.
000400 03 ADDR-1 PIC X(20).
000410 03 ADDR-2 PIC X(20).
000420 03 ADDR-3 PIC X(20).
000430 03 COUNTY PIC X(10).
000440 03 POST-CODE PIC X(10).
000450 02 JOB-NAME PIC X(20).
000460 02 DATES.
000470 03 LETTE PIC X(8).
000480 03 APPL PIC X(8).
000490 03 REPLY PIC X(8).
000500 03 INTER PIC X(8).
000510 02 JOB.
000520 03 OFFER PIC X.
000530 03 TAKE PIC X.
000540
000550 FD PRINTER
000560 DATA RECORD IS PRINT-LINE.
000570

```

000580 01 PRINT-LINE          PIC X(79).
000590
000600 WORKING-STORAGE SECTION.
000610
000620 01 DASHES              PIC X(63) VALUE ALL "-".
000630 01 STARS               PIC X(63) VALUE ALL "**".
000640 01 BLANKS              PIC X(63) VALUE ALL SPACES.
000650 01 EOF                 PIC 9 VALUE 0.
000660 88 END-OF-FILE         VALUE 1.
000670 01 DUMMY              PIC X.
000680 88 DUMMY-VALID          VALUES ARE " ", "@".
000690 88 RETRY               VALUE " ".
000700 88 ABORT               VALUE "@".
000710 01 MENU-OPTION        PIC X.
000720 88 VALID              VALUE "A", "D", "E",
                                "V", "I", "Q", "@",
                                "P", "X", "L".
000730
000740
000750 88 LEAVE              VALUE "Q", "@".
000760 01 TEMP-8             PIC X(8).
000770 01 TEMP               PIC X(20).
000780 01 NAME               PIC X(20).
000790 01 TEMP-LINE.
000800 02 PART-1             PIC X(16) JUSTIFIED RIGHT.
000810 02 FILLER             PIC X VALUE IS SPACE.
000820 02 PART-2            PIC X(20).
000830 02 FILLER            PIC X(5) VALUE IS SPACES.
000840 02 PART-3            PIC X(22) JUSTIFIED RIGHT.
000850 02 FILLER            PIC X VALUE IS SPACE.
000860 02 PART-4            PIC X(8).
000870
000880 01 PRINT-TEST         PIC 9 VALUE 0.
000890 88 PRINTED            VALUE 1.
000900 01 LINE-COUNT        PIC 99.
000910 01 TEMP-INDEX.
000920 02 FILLER            PIC X(5) VALUE SPACES.
000930 02 NAME-1            PIC X(20).
000940 02 FILLER            PIC X(10) VALUE SPACES.
000950 02 NAME-2            PIC X(20).
000960
000970 01 SYSTEM-DATE.
000980 02 SYSTEM-YEAR        PIC 99.
000990 02 SYSTEM-MONTH      PIC 99.
001000 02 SYSTEM-DAY       PIC 99.
001010 01 DISPLAY-DATE.
001020 02 TODAY-DAY        PIC Z9.
001030 02 FILLER          PIC X VALUE "/".
001040 02 TODAY-MONTH      PIC Z9.
001050 02 FILLER          PIC X VALUE "/".

```

```
001060      02  TODAY-YEAR      PIC 99.
001070
001080 01  SYSTEM-TIME.
001090      02  SYSTEM-HOUR      PIC 99.
001100      02  SYSTEM-MINUTE    PIC 99.
001110      02  FILLER          PIC 999.
001120 01  DISPLAY-TIME.
001130      02  HOUR             PIC Z9.
001140      02  FILLER          PIC X   VALUE ".".
001150      02  MINUTES         PIC 99.
001160      02  FILLER          PIC X   VALUE SPACE.
001170      02  NOON            PIC X.
001180      02  FILLER          PIC X   VALUE "m".
001190
001200 PROCEDURE DIVISION.
001210 MAINLINE-CODE.
001220     PERFORM INITIALIZATION.
001230     PERFORM PROGRAM-BODY UNTIL LEAVE.
001240     PERFORM CLOSE-DOWN.
001250     STOP RUN.
001260
001270* Initialize the system.
001280
001290 INITIALIZATION.
001300     OPEN OUTPUT PRINTER,
001310         I-O  MASTER-DISC-FILE.
001320     PERFORM CONVERT-DATE.
001330     PERFORM CONVERT-TIME.
001340
001350* Convert Date into English format
001360 CONVERT-DATE.
001370     ACCEPT SYSTEM-DATE FROM DATE.
001380     MOVE SYSTEM-YEAR  TO TODAY-YEAR.
001390     MOVE SYSTEM-MONTH TO TODAY-MONTH.
001400     MOVE SYSTEM-DAY   TO TODAY-DAY.
001410
001420* Convert time into English
001430 CONVERT-TIME.
001440     ACCEPT SYSTEM-TIME FROM TIME
001450     MOVE SYSTEM-HOUR   TO HOUR.
001460     MOVE SYSTEM-MINUTE TO MINUTES.
001470     IF HOUR > 12
001480         MOVE "p" TO NOON
001490         SUBTRACT 12 FROM SYSTEM-HOUR GIVING HOUR
001500     ELSE
001510         MOVE "a" TO NOON.
001520
001530* Close the system down
```

```
001540
001550 CLOSE-DOWN.
001560     DISPLAY "*****" LINE 6.
001570     DISPLAY "***          ***".
001580     DISPLAY "*** End of Program ***".
001590     DISPLAY "***          ***".
001600     DISPLAY "*****".
001610     CLOSE PRINTER,
001620             MASTER-DISC-FILE.
001630     DISPLAY SPACE ERASE LINE 1.
001640
001650* Display menu and distribute selection
001660
001670 PROGRAM-BODY.
001680     PERFORM DISPLAY-MENU.
001690     PERFORM GET-SELECTION UNTIL VALID.
001700     PERFORM DISPLAY-TITLE.
001710
001720     IF MENU-OPTION = "A"
001730             PERFORM ADD-A-RECORD THRU ADD-EXIT.
001740     IF MENU-OPTION = "D" PERFORM DELETE-A-RECORD.
001750     IF MENU-OPTION = "E" PERFORM EDIT-A-RECORD.
001760     IF MENU-OPTION = "I" PERFORM VIEW-INDEX.
001770     IF MENU-OPTION = "L" PERFORM LIST-FILE.
001780     IF MENU-OPTION = "P" PERFORM PRINT-A-RECORD.
001790     IF MENU-OPTION = "V" PERFORM VIEW-A-RECORD.
001800     IF MENU-OPTION = "X" PERFORM PRINT-INDEX.
001810
001820* Display Options Menu.
001830 DISPLAY-MENU.
001840     PERFORM DISPLAY-TITLE.
001850     DISPLAY "A - Add    an Entry" POSITION 7 LINE 5.
001860     DISPLAY "E - Edit   an Entry" POSITION 7.
001870     DISPLAY "D - Delete an Entry" POSITION 7.
001880     DISPLAY "V - View   an Entry" POSITION 7 LINE 9.
001890     DISPLAY "I - List   the Index" POSITION 7 LINE 10.
001900     DISPLAY "P - Print  an Entry" POSITION 37 LINE 9.
001910     DISPLAY "X - Print  the Index" POSITION 37 LINE 10.
001920     DISPLAY "L - List   all Entries" POSITION 37 LINE 11.
001930     DISPLAY "Q - Quit out of program"
001940                                     POSITION 20 LINE 13.
001950     MOVE " " TO MENU-OPTION.
001960
001970* Wait for key selection from Menu.
001980 GET-SELECTION.
001990     MOVE " " TO MENU-OPTION.
002000     DISPLAY "Make Selection =>" POSITION 5 LINE 15.
002010     ACCEPT MENU-OPTION LINE 15 POSITION 23.
```



```
002020      IF VALID
002030          DISPLAY MENU-OPTION POSITION 23 LINE 15
002040          DISPLAY "Confirm"    POSITION 25 LINE 15
002050          MOVE " " TO DUMMY
002060          ACCEPT DUMMY LINE 15 POSITION 33
002070          IF DUMMY NOT = " " AND DUMMY NOT = MENU-OPTION
002080              MOVE "@" TO MENU-OPTION.
002090          DISPLAY "          " LINE 15 POSITION 23.
002100
002110* Display Title (Screen) Mask
002120 DISPLAY-TITLE.
002130     PERFORM CONVERT-DATE.
002140     PERFORM CONVERT-TIME.
002150     DISPLAY "Date:" LINE 1 POSITION 50 ERASE.
002160     DISPLAY "Time:" LINE 2 POSITION 50.
002170     DISPLAY DISPLAY-DATE LINE 1 POSITION 56.
002180     DISPLAY DISPLAY-TIME LINE 2 POSITION 56.
002190
002200     DISPLAY "Job Applications Data Base"
002210             LINE 1 POSITION 12.
002220     DISPLAY "Written in RM-COBOL by P. Knaggs. (c) 1987"
002230             LINE 2 POSITION 4.
002240     DISPLAY DASHES LINE 3 POSITION 1.
002250     DISPLAY STARS LINE 16 POSITION 1.
002260
002270* Select Record to Edit, View, Print, Delete.
002280 SELECT-RECORD.
002290     PERFORM DISPLAY-RECORD-MASK.
002300     MOVE SPACES TO NAME.
002310     ACCEPT NAME LINE 5 POSITION 18 TAB PROMPT.
002320     IF NAME = SPACES GO TO SELECT-RECORD.
002330     MOVE NAME TO MASTER-KEY.
002340     READ MASTER-DISC-FILE
002350         KEY IS MASTER-KEY
002360         INVALID KEY GO TO NOT-FOUND.
002370     PERFORM DISPLAY-RECORD.
002380
002390* Record not found.
002400 NOT-FOUND.
002410     DISPLAY "Entry is not in the system"
002420             LINE 14 POSITION 19.
002430     PERFORM RETRY-ABORT.
002440     IF RETRY GO TO SELECT-RECORD
002450         ELSE GO TO PROGRAM-BODY.
002460
002470* Display the screen mask for the record.
002480 DISPLAY-RECORD-MASK.
002490     DISPLAY "Company Name:"    LINE 5 POSITION 4.
```

```

002500      DISPLAY "Address:"          LINE 6 POSITION 9.
002510      DISPLAY "County:"          LINE 9 POSITION 10.
002520      DISPLAY "Post Code:"       LINE 10 POSITION 7.
002530      DISPLAY "Job Description:"  LINE 12 POSITION 1.
002540      DISPLAY "Letter Sent:"     LINE 6 POSITION 43.
002550      DISPLAY "Application:"     LINE 7 POSITION 43.
002560      DISPLAY "Reply Received:"  LINE 8 POSITION 40.
002570      DISPLAY "Interview:"       LINE 9 POSITION 45.
002580      DISPLAY "Job Offered:"     LINE 10 POSITION 43.
002590      DISPLAY "Job Accepted:"    LINE 11 POSITION 42.
002600
002610* Display the contents of the record on the screen.
002620 DISPLAY-RECORD.
002630      DISPLAY MASTER-KEY  LINE 5 POSITION 18.
002640      DISPLAY ADDR-1     LINE 6 POSITION 18
002650          ADDR-2         LINE 7 POSITION 18
002660          ADDR-3         LINE 8 POSITION 18
002670          COUNTY        LINE 9 POSITION 18
002680          POST-CODE      LINE 10 POSITION 18.
002690      DISPLAY JOB-NAME   LINE 12 POSITION 18.
002700
002710      DISPLAY LETTE LINE 6 POSITION 56.
002720      DISPLAY APPL  LINE 7 POSITION 56.
002730      DISPLAY REPLY LINE 8 POSITION 56.
002740      DISPLAY INTER LINE 9 POSITION 56.
002750
002760      IF OFFER = "Y" DISPLAY "Yes" LINE 10 POSITION 56
002770          ELSE DISPLAY "No " LINE 10 POSITION 56.
002780      IF TAKE  = "Y" DISPLAY "Yes" LINE 11 POSITION 56
002790          ELSE DISPLAY "No " LINE 11 POSITION 56.
002800
002810* Wait for a Space or @
002820 RETRY-ABORT.
002830      DISPLAY "Press <Space> to re-try, or <@> to abort"
002840          LINE 15 POSITION 12.
002850      ACCEPT DUMMY LINE 15 POSITION 54.
002860      IF NOT DUMMY-VALID GO TO RETRY-ABORT.
002870      DISPLAY BLANKS LINE 14.
002880      DISPLAY BLANKS LINE 15.
002890
002900*Add a Record to the system
002910 ADD-A-RECORD.
002920      DISPLAY " Add a Record " LINE 16 POSITION 24.
002930      PERFORM DISPLAY-RECORD-MASK.
002940      ACCEPT NAME LINE 5 POSITION 18 TAB PROMPT.
002950      MOVE NAME TO MASTER-KEY.
002960      READ MASTER-DISC-FILE
002970          KEY IS MASTER-KEY

```

```
002980             INVALID KEY GO TO CREATE-REC.
002990     DISPLAY "Entry already exists !"
003000             LINE 14 POSITION 21.
003010     PERFORM RETRY-ABORT.
003020     IF RETRY GO TO ADD-A-RECORD.
003030     IF ABORT GO TO ADD-EXIT.
003040
003050* Create a new Record.
003060 CREATE-REC.
003070     MOVE SPACES TO ADDRESS,
003080             JOB-NAME,
003090             DATES,
003100             JOB.
003110     PERFORM EDIT-REC.
003120     MOVE NAME TO MASTER-KEY.
003130     WRITE MASTER-REC.
003140
003150 ADD-EXIT.
003160     EXIT.
003170
003180*Edit a record in the system
003190 EDIT-A-RECORD.
003200     DISPLAY " Edit a Entry "
003210             LINE 16 POSITION 25.
003220     PERFORM SELECT-RECORD.
003230     DISPLAY "Is this the correct Entry <Y/N> ? "
003240             LINE 15 POSITION 15.
003250     MOVE "X" TO DUMMY.
003260     PERFORM GET-KEY UNTIL DUMMY-VALID OR
003270             DUMMY = "Y" OR
003280             DUMMY = "N".
003290     IF DUMMY = "N" GO TO EDIT-A-RECORD.
003300     IF DUMMY = "Y" OR RETRY
003310             PERFORM EDIT-REC
003320             REWRITE MASTER-REC.
003330
003340* Edit a record in memory (Excluding Name)
003350 EDIT-REC.
003360     DISPLAY BLANKS LINE 15.
003370     MOVE ADDR-1 TO TEMP.
003380     ACCEPT ADDR-1 LINE 6 POSITION 18 TAB.
003390     IF ADDR-1 = SPACE MOVE TEMP TO ADDR-1.
003400     MOVE ADDR-2 TO TEMP.
003410     ACCEPT ADDR-2 LINE 7 POSITION 18 TAB.
003420     IF ADDR-2 = SPACE MOVE TEMP TO ADDR-2.
003430     MOVE ADDR-3 TO TEMP.
003440     ACCEPT ADDR-3 LINE 8 POSITION 18 TAB.
003450     IF ADDR-3 = SPACE MOVE TEMP TO ADDR-3.
```

```
003460      MOVE COUNTY TO TEMP.
003470          ACCEPT COUNTY LINE 9 POSITION 18 TAB.
003480          IF COUNTY = SPACE MOVE TEMP TO COUNTY.
003490      MOVE POST-CODE TO TEMP.
003500          ACCEPT POST-CODE LINE 10 POSITION 18 TAB.
003510          IF POST-CODE = SPACE MOVE TEMP TO POST-CODE.
003520      MOVE JOB-NAME TO TEMP
003530          ACCEPT JOB-NAME LINE 12 POSITION 18 TAB.
003540          IF JOB-NAME = SPACE MOVE TEMP TO JOB-NAME.
003550
003560      MOVE LETTE TO TEMP-8.
003570          MOVE 6 TO LINE-COUNT.
003580          PERFORM READ-DATE.
003590          MOVE TEMP-8 TO LETTE.
003600      MOVE APPL TO TEMP-8.
003610          MOVE 7 TO LINE-COUNT.
003620          PERFORM READ-DATE.
003630          MOVE TEMP-8 TO APPL.
003640      MOVE REPLY TO TEMP-8.
003650          MOVE 8 TO LINE-COUNT.
003660          PERFORM READ-DATE.
003670          MOVE TEMP-8 TO REPLY.
003680      MOVE INTER TO TEMP-8.
003690          MOVE 9 TO LINE-COUNT.
003700          PERFORM READ-DATE.
003710          MOVE TEMP-8 TO INTER.
003720
003730      MOVE OFFER TO TEMP-8.
003740          MOVE 10 TO LINE-COUNT.
003750          PERFORM READ-YN.
003760          MOVE DUMMY TO OFFER.
003770      MOVE TAKE TO TEMP-8.
003780          MOVE 11 TO LINE-COUNT.
003790          PERFORM READ-YN.
003800          MOVE DUMMY TO TAKE.
003810
003820      READ-YN.
003830          MOVE "X" TO DUMMY.
003840          DISPLAY "***" LINE LINE-COUNT POSITION 56.
003850          DISPLAY "Answer Yes or No <Y/N> ? "
003860              LINE 15 POSITION 19.
003870          PERFORM GET-KEY UNTIL DUMMY = "Y" OR
003880              DUMMY = "N" OR
003890              DUMMY-VALID.
003900          IF DUMMY-VALID MOVE TEMP-8 TO DUMMY.
003910          IF DUMMY = "Y"
003920              DISPLAY "Yes" LINE LINE-COUNT POSITION 56
003930          ELSE
```

```

003940          DISPLAY "No " LINE LINE-COUNT POSITION 56.
003950      DISPLAY BLANKS LINE 15.
003960
003970 READ-DATE.
003980      DISPLAY "**/**/**" LINE LINE-COUNT POSITION 56.
003990      DISPLAY "Set to Todays date <Y/N> ? "
004000          LINE 15 POSITION 18.
004010      MOVE "X" TO DUMMY.
004020      PERFORM GET-KEY UNTIL DUMMY-VALID OR
004030          DUMMY = "Y" OR
004040          DUMMY = "N".
004050      IF DUMMY = "Y"
004060          MOVE DISPLAY-DATE TO TEMP-8.
004070      IF ABORT GO TO READ-DATE.
004080      DISPLAY BLANKS LINE 15.
004090      DISPLAY TEMP-8 LINE LINE-COUNT POSITION 56.
004100
004110* Delete a record from the system
004120 DELETE-A-RECORD.
004130      DISPLAY " Remove an Entry "
004140          LINE 16 POSITION 22.
004150      PERFORM SELECT-RECORD.
004160      DISPLAY "Is this the correct Entry <Y/N> ? "
004170          LINE 15 POSITION 15.
004180      MOVE "X" TO DUMMY.
004190      PERFORM GET-KEY UNTIL DUMMY-VALID OR
004200          DUMMY = "Y" OR
004210          DUMMY = "N".
004220      IF DUMMY = "Y" OR RETRY
004230          PERFORM DELETE-REC.
004240      IF DUMMY = "N" GO TO DELETE-A-RECORD.
004250
004260 DELETE-REC.
004270      DELETE MASTER-DISC-FILE RECORD.
004280
004290*Print a record on the printer
004300 PRINT-A-RECORD.
004310      DISPLAY " Print an Entry "
004320          LINE 16 POSITION 23.
004330      PERFORM SELECT-RECORD.
004340      MOVE "X" TO DUMMY.
004350      DISPLAY "Is this the Correct Entry <Y/N> ? "
004360          LINE 15 POSITION 15.
004370      PERFORM GET-KEY UNTIL DUMMY-VALID OR
004380          DUMMY = "Y" OR
004390          DUMMY = "N".
004400      IF DUMMY = "N" GO TO PRINT-A-RECORD.
004410      IF DUMMY = "Y" OR RETRY

```

```
004420             PERFORM PRINT-REC.
004430
004440 PRINT-REC.
004450     MOVE 1 TO PRINT-TEST.
004460     MOVE "Company Name:" TO PART-1.
004470         MOVE MASTER-KEY TO PART-2.
004480         MOVE SPACES TO PART-3, PART-4.
004490         WRITE PRINT-LINE FROM TEMP-LINE.
004500     MOVE "Address:" TO PART-1.
004510         MOVE ADDR-1 TO PART-2.
004520         MOVE "Date of Letter:" TO PART-3.
004530         MOVE LETTE TO PART-4.
004540         WRITE PRINT-LINE FROM TEMP-LINE
004550             AFTER ADVANCING 1 LINE.
004560     MOVE SPACES TO PART-1.
004570         MOVE ADDR-2 TO PART-2.
004580         MOVE "Date of Application:" TO PART-3.
004590         MOVE APPL TO PART-4.
004600         WRITE PRINT-LINE FROM TEMP-LINE.
004610     MOVE ADDR-3 TO PART-2.
004620         MOVE "Date Reply received:" TO PART-3.
004630         MOVE REPLY TO PART-4.
004640         WRITE PRINT-LINE FROM TEMP-LINE.
004650     MOVE "County:" TO PART-1.
004660         MOVE COUNTY TO PART-2.
004670         MOVE "Date of Interview:" TO PART-3.
004680         MOVE INTER TO PART-4.
004690         WRITE PRINT-LINE FROM TEMP-LINE.
004700     MOVE "Post Code:" TO PART-1.
004710         MOVE POST-CODE TO PART-2.
004720         MOVE "Job Offered:" TO PART-3.
004730         IF OFFER = "Y" MOVE "Yes" TO PART-4
004740             ELSE MOVE "No" TO PART-4.
004750         WRITE PRINT-LINE FROM TEMP-LINE.
004760     MOVE SPACES TO PART-1, PART-2.
004770         MOVE "Job Accepted:" TO PART-3.
004780         IF TAKE = "Y" MOVE "Yes" TO PART-4
004790             ELSE MOVE "No" TO PART-4.
004800         WRITE PRINT-LINE FROM TEMP-LINE.
004810     MOVE "Job Description:" TO PART-1.
004820         MOVE JOB-NAME TO PART-2.
004830         MOVE SPACES TO PART-3, PART-4.
004840         WRITE PRINT-LINE FROM TEMP-LINE
004850             BEFORE ADVANCING 2 LINES.
004860
004870
004880* Display a Record on the screen.
004890 VIEW-A-RECORD.
```

```
004900      DISPLAY "  View a Entry  "
004910          LINE 16 POSITION 24.
004920      PERFORM SELECT-RECORD.
004930      MOVE 0 TO EOF.
004940      PERFORM VIEW-MENU.
004950      IF DUMMY = "R" GO TO VIEW-A-RECORD.
004960
004970 VIEW-MENU.
004980      DISPLAY
004990          "<P>rint, <E>dit, <D>elete, <N>ext, <R>epeat or <@>"
005000          LINE 15 POSITION 8.
005010      MOVE "X" TO DUMMY.
005020      PERFORM GET-KEY UNTIL DUMMY = "P" OR
005030                          DUMMY = "E" OR
005040                          DUMMY = "D" OR
005050                          DUMMY = "R" OR
005060                          DUMMY = "N" OR
005070                          DUMMY-VALID.
005080      DISPLAY BLANKS LINE 15.
005090      IF DUMMY = "N" AND NOT END-OF-FILE
005100          PERFORM GET-NEXT-RECORD
005110          GO TO VIEW-MENU.
005120      IF DUMMY = "P" PERFORM PRINT-REC.
005130      IF DUMMY = "D" PERFORM DELETE-REC.
005140      IF DUMMY = "E" PERFORM EDIT-REC
005150          REWRITE MASTER-REC.
005160
005170 GET-NEXT-RECORD.
005180      READ MASTER-DISC-FILE NEXT RECORD
005190          AT END MOVE 1 TO EOF.
005200      PERFORM DISPLAY-RECORD.
005210
005220* List the Index of the file on the video
005230*      a screen at a time.
005240 VIEW-INDEX.
005250      DISPLAY "  View the Index  "
005260          LINE 16 POSITION 23.
005270      PERFORM OPEN-SEQUENTIAL.
005280      MOVE 5 TO LINE-COUNT.
005290      MOVE "X" TO DUMMY.
005300      PERFORM VIEW-A-SCREEN UNTIL END-OF-FILE OR ABORT.
005310
005320 VIEW-A-SCREEN.
005330      PERFORM READ-INDEX.
005340      DISPLAY TEMP-INDEX LINE LINE-COUNT.
005350      ADD 1 TO LINE-COUNT.
005360      IF LINE-COUNT = 14 OR END-OF-FILE
005370          DISPLAY "Press <Space> for next page"
```

```
005380                LINE 15 POSITION 18
005390                MOVE "X" TO DUMMY
005400                PERFORM GET-KEY UNTIL DUMMY-VALID
005410                PERFORM DISPLAY-TITLE
005420                DISPLAY " View the Index "
005430                LINE 16 POSITION 23
005440                MOVE 5 TO LINE-COUNT.
005450
005460* Print the Index of the file on the printer
005470*                correctly paged (if required).
005480 PRINT-INDEX.
005490                DISPLAY " Print the Index "
005500                LINE 16 POSITION 22.
005510                DISPLAY "*** Printing ***"
005520                LINE 8 POSITION 25.
005530                IF PRINTED
005540                    WRITE PRINT-LINE FROM SPACE
005550                    BEFORE ADVANCING PAGE.
005560                WRITE PRINT-LINE FROM SPACE
005570                AFTER ADVANCING 2 LINES.
005580                MOVE 60 TO LINE-COUNT.
005590                PERFORM OPEN-SEQUENTIAL.
005600                PERFORM PRINT-PAGE UNTIL END-OF-FILE.
005610                MOVE 0 TO PRINT-TEST.
005620
005630 PRINT-PAGE.
005640                PERFORM READ-INDEX.
005650                WRITE PRINT-LINE FROM TEMP-INDEX.
005660                SUBTRACT 1 FROM LINE-COUNT.
005670                IF LINE-COUNT = 0 OR END-OF-FILE
005680                    WRITE PRINT-LINE FROM SPACES
005690                    BEFORE ADVANCING PAGE
005700                WRITE PRINT-LINE FROM SPACES
005710                BEFORE ADVANCING 2 LINES
005720                MOVE 60 TO LINE-COUNT.
005730
005740* Read the next two Index values into TEMP-INDEX.
005750 READ-INDEX.
005760                MOVE SPACES TO TEMP-INDEX.
005770                READ MASTER-DISC-FILE NEXT RECORD
005780                AT END MOVE 1 TO EOF.
005790                IF NOT END-OF-FILE
005800                    PERFORM READ-INDEX-2.
005810
005820 READ-INDEX-2.
005830                MOVE MASTER-KEY TO NAME-1.
005840                READ MASTER-DISC-FILE NEXT RECORD
005850                AT END MOVE 1 TO EOF.
```



```
005860      IF NOT END-OF-FILE
005870          MOVE MASTER-KEY TO NAME-2.
005880
005890* List all records to the printer.
005900 LIST-FILE.
005910      PERFORM OPEN-SEQUENTIAL.
005920      IF PRINTED
005930          WRITE PRINT-LINE FROM SPACE
005940              BEFORE ADVANCING PAGE.
005950      PERFORM READ-PRINT UNTIL END-OF-FILE.
005960      WRITE PRINT-LINE FROM SPACES
005970          BEFORE ADVANCING PAGE.
005980      MOVE 0 TO PRINT-TEST.
005990
006000 READ-PRINT.
006010      READ MASTER-DISC-FILE NEXT RECORD
006020          AT END MOVE 1 TO EOF.
006030      IF NOT END-OF-FILE
006040          PERFORM PRINT-REC.
006050
006060 OPEN-SEQUENTIAL.
006070      MOVE SPACES TO MASTER-KEY.
006080      READ MASTER-DISC-FILE
006090          KEY IS MASTER-KEY
006100          INVALID KEY GO TO INVALID-FILE.
006110      MOVE 0 TO EOF.
006120
006130* File incorrectly created (Must have spaces key!)
006140 INVALID-FILE.
006150      PERFORM DISPLAY-TITLE.
006160      DISPLAY "*** Data file has been corrupted" LINE 7.
006170      DISPLAY "*** Or has not been CREATED properly".
006180      CLOSE PRINTER,
006190          MASTER-DISC-FILE.
006200      STOP "File Indexing Error".
006210
006220* Wait for a key from the keyboard
006230 GET-KEY.
006240      ACCEPT DUMMY LINE 15 POSITION 63.
006250      DISPLAY SPACE LINE 15 POSITION 63.
006260
006270 END PROGRAM.
```

HYPERCUBE ARRAYS

=====

In response to your appeals, I have been trying to think of something to write about for the NATGUG news letter. The trouble is this: in the old days everything I wanted to do on my Video Genie was an adventure. To produce a usable result involved solving unfamiliar problems. If I found the solution, it was often worth sharing with other members. If I failed, then perhaps some other user had been cleverer, and there was nothing to lose by using the news letter to ask them.

Since last December I have had my Amstrad PC1512 SD mono and life is different. Everything works the way the book says. I can get cheap and powerful public-domain or user-supported packages. PC-Write, and PC-Calc, serve nearly all of my needs. Basic2 is fine, and Turbo Pascal is fast and flexible (even if annoyingly pedantic about data types). MS-DOS is not as terrible as people make out, because I need to use and learn only a fraction of its apparent facilities in order to do all I want to do.

GEM is less useful, and its OUTPUT function can't be made to work on my minimum configuration (unless I have a dud version) - but GEM can be run under MS-DOS, enabling the <PrtSc> key to produce graphic output. Electric Studio's "IMPRINT" package is an inexpensive and handy alternative to GEM OUTPUT. My Epson MX80 does all I ask of it, just as it did when it was connected to the Genie. So what is there to write about?

Well, the last issue had a note from Leon Heller about arranging Transputers in hypercube arrays. It occurred to me that many of your readers may never have seen a 4-dimensional object, so for their edification I got out the listing of an old TINY PASCAL program I once wrote for my Genie. It turned out to be very easy to adapt to Turbo Pascal on the PC, and here it is:-

```
{program to draw a hyperobject}
```

```
var key: char; var d: integer;
```

```
procedure hyper(d,x,y: integer); {draws a d-dimensional hyper object at x,y}
```

```
const hres=320; vres=200;
```

```
var x2,y2: integer;
```

```
function xoffset(d:integer):integer; {returns a suitable x-offset for the
next level of drawing}
```

```

begin if (d mod 2)=1 then xoffset:=hres*(d+4) div 25
      else xoffset:=hres*(d-2) div 20 end;

function yoffset(d:integer):integer; {ditto for y-offset}

begin if (d mod 2)=1 then yoffset:=vres*(d-1) div 15
      else yoffset:=vres*(d+3) div 24 end;

procedure link(d,x,y,x2,y2: integer); {draws links between two d-
dimensional objects at x,y and x2,y2}

var xos,yos: integer;

begin if d=1 then draw(x,y,x2,y2,15)
      else begin link(d-1,x,y,x2,y2); xos:=xoffset(d-1);
            yos:=yoffset(d-1); link(d-1,x+xos,y+yos,x2+xos,y2+yos);
            end {else}; end {proc link};

begin {main part of procedure hyper}

  if d=0 then plot(x,y,15) else begin
    hyper(d-1,x,y); x2:=x+xoffset(d); y2:=y+yoffset(d); hyper(d-
    1,x2,y2); link(d,x,y,x2,y2);
  end {else}

end {procedure hyper};

begin {main prog}

  graphcolormode;

  repeat writeln('enter number of dimensions'); read(d)
  until (d>0) and (d<10);

  graphcolormode; hyper(d,10,10); write(d,'-dimensional object');

  {at this point the program waits for key Q to be pressed to quit display
  of final drawing.  Meanwhile, use of PrtSc key will get the drawing
  printed provided that MS-Dos GRAPHICS command has previously been set
  up}

  repeat read(kbd,key)
  until (key='Q') or (key='q');

end {program}.

```


Bank 0 ... Normal Memory 0000H-7FFFH
 Bank 1 ... Normal Memory 8000H-FFFFH
 Bank 2 ... Alternate Memory
 Bank 3 ... Alternate Memory

Banks 0 and 1 are always fitted, Banks 2 and 3 are optional additions.

Bank switching is achieved by sending a control byte to PORT 84H (write only), called the OPTION port. This port also controls some video functions. The meaning of each bit in this control byte is as follows:

Bits 0-1 select memory configuration 0-3. Normal configuration for Model 4 mode is 1, and for Model III mode it is 0. There are two other modes also, but they are not relevant here.
 Bit 2 when =0 selects 64x16 video, when =1 selects 80x24 video.
 Bit 3 when =0 selects normal video, when =1 selects inverse video.
 Bits 4-5 select the memory bank to switch:
 0=Bank 0, 1=Bank 1, 2=Bank 2, 3=Bank 3.
 Bit 6 determines where to connect the selected bank:
 when =1, connect selected bank to 0000H-7FFFH.
 when =0, connect selected bank to 8000H-FFFFH.
 NOTE that these appear to be the wrong way round!
 Bit 7 selects the video page to use (page 0 or 1).

Switching a bank in simply connects it to the addresses chosen in place of the one previously there, and leaves the previous one unconnected; the content of the memory in any bank is NOT altered by switching it in or out. Information stored in switched-out banks is maintained unless the computer is switched off.

Since the DOS and the running program (or at least part of it) are resident in Bank 0 at addresses 0000H-7FFFH, we must normally confine any bank switching operations to banks 1,2, and 3 and addresses 8000H-FFFFH.

The control byte to send depends on whether we are using Model 4 or III mode, and on the precise switching to be done:

To swop Bank 2 with Bank 1, control byte=23H (20H for Model III)
 To swop Bank 3 with Bank 1, control byte=33H (30H for Model III)
 To swop Bank 1 back again, control byte=13H (10H for Model III)

Other swops are not normally possible due to the need to keep Bank 0 intact.

The program to carry out the swop needs to be able to swop, copy data from the new bank into bank 0, and then swop back again. Then the existing basic program can be used to read the data from where it has been stored in Bank 0. It has to be a machine code program, and it is vital that it is located in

Bank 0 so that it does not get switched out halfway through the process. I have managed to get it down to only 20 bytes:

```

0000      00001      ORG      0           ;see text following.
0084      00002 OPTION EQU      84H
0001      00003 BANK  EQU      $+1
0000 3E00      00004 ENTRY LD      A,0           ;poke here bank code
0002 D384      00005      OUT      (OPTION),A
0005      00006 BYTES EQU      $+1
0004 010000      00007 LD      BC,0           ;poke here no of bytes
0008      00008 SOURCE EQU      $+1
0007 210000      00009 LD      HL,0           ;poke here source
000B      00010 DESTN EQU      $+1
000A 110000      00011 LD      DE,0           ;poke here destn
000D EDB0      00012 LDIR           ;copy
000F 3E13      00013 EXIT LD      A,13H        ;use 10H for Mod III
0011 D384      00014      OUT      (OPTION),A   ;to restore Bank 1
0013 C9      00015      RET
0014      00016 BUFFER EQU      $
0000      00017      END      ENTRY

```

This program is completely relocatable and can be put anywhere in memory. It can be used in either Model 4 or Model III mode, but needs to be in different places. Since it is so short, it is easy to insert it in a data statement and read it from basic.

Bearing in mind that it must be in Bank 0, I have located it as follows:
 For Model 4 mode, in the spare device driver space in low memory.
 For Model III mode, in a Dummy File Buffer

The spare device driver space in TRSDOS 6 is not explained in any book I have seen except the "Programmer's Guide to LDOS/TRSDOS" by Roy Soltoff, but a pointer to it is contained in bytes 0206-7H. For DOSPLUS the space is documented as User Memory and there is a pointer to it in @FLAGS. For the programs below I have ignored all this for simplicity, and have assumed that an unmodified DOS is being used, i.e. without any extra devices, exactly as booted from scratch. If you want to use extra devices, you will have to follow the rules, which I have skipped because I think they will confuse this simple demo. I'll be pleased to help anyone on this if needed. For the simple approach I install the program at 0FF4H for TRSDOS and 180AH for DOSPLUS. The following Basic program works for either, the first line identifying the DOS and setting the right load point:

```

10 IF PEEK(&H4DC)=74 THEN ENTRY=&HFF4 ELSE ENTRY=&H180A
20 DATA 62,0,211,132,1,0,0,33,0,0,17,0,0,237,176,62,19,211,132,201
30 FOR A=0 TO 19:READ B:POKE ENTRY+A,B:NEXT
40 DEF USRO=ENTRY
50 BUFFER=ENTRY+20:BHI=INT(BUFFER/256):BLO=BUFFER-256*BHI
52 B$=STRING$(236,0):POKE VARPTR(B$)+1,BLO:POKE VARPTR(B$)+2,BHI
60 BANK=ENTRY+1:BYTES=ENTRY+5:SOURCE=ENTRY+8:DESTN=ENTRY+11
70 '
80 ' *****
90 ' * To use, first POKE the required values into BANK, *
100 ' *   BYTES, SOURCE and DESTN, and then do X=USRO(). *
110 ' * *
120 ' * The rest of this program ia just a demonstration of *
130 ' * one way to use the facility. It is really up to the *
140 ' * user to work out ways that suit him/her best. This *
150 ' * way simply defines a start address and a length of *
160 ' * string to be put at that address.
170 ' * Note that when reading you must specifiiy the right *
180 ' * number of bytes to read or you will get rubbish for *
190 ' * everything beyond the string you put there! *
200 ' *****
210 '
220 'FNUC$(A$) converts A$ to upper-case.
230 DEF FNUC$(A$)=CHR$(ASC(A$)+32*(A$)="a" AND A$<="z")
240 '
250 'FNDH!(A$) converts a HEX string to single-prec number
260 DEF FNDH!(A$) =INSTR("123456789ABCDEF",MID$(A$,1,1))*4096
                +INSTR("123456789ABCDEF",MID$(A$,2,1))*256
                +INSTR("123456789ABCDEF",MID$(A$,3,1))*16
                +INSTR("123456789ABCDEF",MID$(A$,4,1))
270 '
279 CLS
280 PRINT @(2,0),CHR$(30)"Read, Write or Quit (R/W/Q) ? ";
290 RW$=INKEY$:IF RW$=""THEN 290 ELSE RW$=FNUC$(RW$):PRINT @(2,28),RW$;
300 ON INSTR("RWQ",RW$) GOTO 310,400,570:GOTO 280
310 '---read routine---
320 GOSUB 490
330 PRINT@(2,0),CHR$(31);:INPUT"Number of Bytes to Read ";NU
340 POKE BYTES,NU
350 POKE BANK,BK:POKE SOURCE,ALO:POKE SOURCE+1,AHI
360 POKE DESTN,BLO:POKE DESTN+1,BHI
370 X=USRO(0)
380 CLS:PRINT@(6,0), LEFT$(B$,NU);
390 FOR A=1 TO 1500:NEXT:GOTO 280
400 '---write routine---
410 GOSUB 490

```

```

420 PRINT@(2,0),CHR$(31);:INPUT"String to Write: ";ST$:LSET B$=ST$
430 POKE BYTES,LEN(ST$)
440 POKE BANK,BK:POKE SOURCE,BLO:POKE SOURCE+1,BHI
450 POKE DESTN,ALO:POKE DESTN+1,AHI
460 X=USR(0)
470 GOTO 280
480 ;
490 '---subr to input address---
500 PRINT@(2,0),CHR$(31);
510 INPUT"Address to use (as a HEX string eg 2A00): ";AD$
520 IF LEN(AD$)<>4 THEN PRINT"Input must be 4 HEX characters!":GOTO 510
530 FOR A=1 TO LEN(AD$):MID$(AD$,A,1)=FNUC$(MID$(AD$,A,1)):NEXT
540 AD!=FNDH!(AD$):IF AD!<32768! THEN AD!=AD!+32768!:BK=35 ELSE BK=51
550 AHI=INT(AD!/256):ALO=AD!-256*AH!
560 RETURN
570 END

```

For Model III mode make the following changes:

```

5 CLEAR 2000'---allocate string space
10 OPEN"R",1,"DUMMY1/RND":FIELD 1, 20 AS A$,236 AS B$:CLOSE
15 ENTRY=PEEK(VARPTR(A$)+1)
20 DATA 62,0,211,132,1,0,0,33,0,0,17,0,0,237,176,62,16,211,132,201
    (note changed 17th byte)
then lines 30 onwards as above, except modify:
lines 280,330,420,500 to read PRINT @(128), .....
line 290 to read ..... :PRINT @(156),RW$
line 380 to read PRINT @(384), .....
line 540 to read ..... :BK=32 ELSE BK=48

```

When working in Model 4 mode both Pages 0 and 1 of the screen are in use. Since bit 7 of the control byte switches pages, the screen will blink conspicuously if the cursor is on the second page (page 1) at the time the subroutine is called. I have avoided this by keeping the cursor high on the screen (hence all the PRINT@'s used). This problem does not occur with Model III as only page 0 is in use.

Finally, do bear in mind that this is only a demonstration program to show how the idea works. Don't try and add any more to it, and in particular don't add device drivers since they will over-write the program in low memory unless you first amend the pointers stored at 0206H (TRSDOS) or in 0FLAGS (DOSPLUS). A lot more can be done, but you must understand what you are doing !

Henry Shepherd, 3 White House Close, Shippon, Abingdon, OXON. OX13 6LP
Tel: (0235) 21921

REVIEW OF XTREE Version 2.00

=====

I thought it would never happen. Me, desert Tandy, never! After all, the Model 4 was everything a good honest clean living chap like myself could want. C/PM and Newdos80 V2. could do almost everything. Wordstar, Supercalc2 and dBasell with a bit of Pascal and Basic thrown in coped with it all. Who in their right mind would want one of those costly I*M clone things?. Not me!

Well, blame Mr. Sugar for what happened next. Drop the price of the clone to within reach of ordinary computerists, add a bit of hype and wave it in front of me. Bang - I'm hooked. (What excuse shall I give the wife? - Well dear, you see, in work we are using computers like this one and if I had one then you've heard it before ?).

Anyway, here I am, sold the Model 4 and all the gubbins for a song and sitting proudly in front of a PC1512 HD20 with 640k memory occasionally feeding the mouse. What a difference the hard disk makes. Megabytes upon megabytes to save files on. (Come on -- get to the point). Soon those megabytes dwindle with hundreds of files in dozens of sub-directories.

Herein lies a problem. With all those sub-directories it becomes a pain to copy, delete, move, rename etc. The full directory path of the source and destination files must be typed and even the use of wildcards is limited if there is little or no similarity in the filenames or extensions. With CP/M there was NSWEEP which allowed mass copying, deleting etc of files. Surely there must be some thing similar for MSDOS. I think I've found the nicest utility to match, or dare I say it, surpass NSWEEP. I am referring to XTREE.

XTREE consists of three files, two of which are concerned with installation and customising and XTREE.EXE. Installation is very simple. Run XTREEINS and set up the monitor type as an IBM PC compatible (Quick Display) then change the colour scheme to suit your taste. I like a royal blue background with white text.

XTREE is one of those programs which deserve a place in the sub-directory where you keep all the utilities and which is included in the PATH command, ie, it will be found no matter which sub-directory you happen to be in at the time.

Once loaded, XTREE displays a screen with several windows. The largest window shows a representation of part of the directory tree structure with the root shown as a backslash \. A vertical line represents the first sub-directory level with branches off it. As only part of the tree is shown, it is possible to scroll up or down using the mouse, arrows, PgUp, PgDn, Home and End keys. The cursor is displayed as a rectangular block which moves over each of the directory names.

-----				FILE: *.*
\				-----
! _CC				DISK: C:FIXED_DISK
! _DB				Available
! ! _DBFILES				bytes: 7,823,360
! ! _LABELS				-----
! _DOSPLUS				DISK Statistics
! _GAMES				Total
! _GEMAPPS				Files: 961
! _GEMBOOT				Bytes: 12,274,481
! _REFLEX				Matching
! _TURBO				Files: 961
! ! _PASCAL				Bytes: 12,274,481
! ! _TOOLBOX				Tagged
! ! _SOURCE				Files: 0
AUTOEXEC.BAT	CONFIG .SYS	DOS .SYS		Bytes: 0
COMMAND.COM	DB .BAT	DOSPLUS.COM		Current Directory
COMMS .BAT	DM .BAT	DOSPLUS.SYS	\	
GEM .BAT	IO .SYS	MSDOS .SYS	Bytes: 204,008	

DIR	Available	Delete	Filespec	Log Disk
COMMANDS	^Showall	^Tag	^Untag	Volume eXecute
scroll	RETURN	file commands	ALT menu	F1 quit F2 help

At this stage, before proceeding to describe the operations within the full file display window, I shall mention the other windows displayed at the beginning. On the right hand side is a small window which shows the disk drive which you are logged onto and its volume name, if any, together with the Available Bytes remaining on the disk. The FILE: *.* above it means that all files are displayed.

Below this is a larger window showing various disk statistics such as the total number of files on the disk and the space they occupy.

The reference to 'Matching' means that because we have the wildcard *.* specified then naturally all the files on the disk match. Tagged files refers to a process of 'tagging'. More of this shortly. The Current Directory shown in the diagram above is the root shown as \ and contains files totalling 204,008 bytes.

Along the bottom three lines are various commands. These are directory commands.

As the cursor is moved over a directory name, the files within that directory are displayed in a window below the main one. This window is not large enough to show all files. To display them, simply press the Enter key twice, first to move the cursor into the lower window then to expand it to almost fill the screen.

123 .BAT	NW .BAT	FILE: *.*
AUTOEXEC.BAT	PAT .BAT	-----
COMMAND.COM	RPED .EXE	DISK: C: ROBERTS
COMMS .BAT	SC3 .	Available
CONFIG .BAK	SC3BKUP .BAT	Bytes: 6,866,944
CONFIG .SYS	SK .BAT	-----
DB .BAT	T .BAT	DIRECTORY Stats
DM .BAT	WS .BAT	Total
DOS .SYS	WSBKUP .BAK	Files: 29
DOSPLUS.COM	WSBKUP .BAT	Bytes: 204,075
DOSPLUS.SYS		Matching
FLS .BAT		Files: 29
FS .BAT		Bytes: 204,075
GEM .BAT		Tagged
IO .SYS		Files: 0
MC .BAT		Bytes: 0

MSDOS	.SYS	Current File	
NVR	.EXE	123	BAT
NVAL	.BAK	Bytes:	91

This is the window where most of XTREE's operations take place and there are a set of specific commands available. These are for Single File Operations

- Attributes - View or modify the DOS file attributes of the current file.
- Copy - Copy the current file to a new disk and/or path.
- Delete - Delete the current file.
- Filespec - Selects a new group of files.
- Log disk - Log on to a different disk drive.
- Move - Move the current file to a different directory.
- Print - Print the contents of the current file.
- Rename - Rename the current file.
- Tag - Tag the current file.
- Untag - Remove the tag from the current file.
- View - View the contents of the current file on the screen.
- eXecute - Execute a program or DOS command.

The View command is very useful. With it you can look at the text in the file if it is ASCII or Wordstar format without displaying the control codes. EXE and COM files can be examined by using VIEW's 'H' subcommand to give a HEX/ASCII display just like some disk editor programs.

With the eXecute command most EXE and COM programs can be run from XTREE and when they are finished you are placed back in XTREE.

In addition, by holding down the ALT key a further menu of commands is available in the Directory Window.

These commands are used by holding down the ALT key, which displays the ALT Directory Commands menu, and pressing the first letter of the command name.

- ALT/File display - Change the format and amount of file information XTREE displays in the File Window.
- ALT/Sort criteria - Change the order XTREE uses when it displays the list of files in the File Window.
- ALT/Tag - Tag any file on the disk whose file attributes match a specified setting.
- ALT/Untag - Remove the tag from any file on the disk whose whose file attributes match a specified setting.
- ALT/eXecute - Execute a program or DOS command. When you return to XTREE, it does not re-log the disk.

Highly recommended, especially for hard disk users.

David Roberts. Prestel MBX 247462564 (MBX's welcome).

[illegible]

USING THE MICROSOFT MOUSE WITH SUPERCALC 4

Several lunch-hours were spent reading the programmers guide for the mouse and after a little bit of experimentation I have solved the problem.

The following small program should be prepared by any text editor and saved in ASCII. The non-document mode of Wordstar is ideal. Call the file CURSOR.DEF.

```
lf:      TYPE 0,75      ;simulate the left cursor key
rt:      TYPE 0,77      ;simulate the right cursor key
up:      TYPE 0,72      ;simulate the up cursor key
dn:      TYPE 0,80      ;simulate the down cursor key
ent:     TYPE ENTER     ;simulate the enter key
```

You will be prompted for the file to convert. Enter `CURSOR.DEF` <cr>

To activate the program type: MENU CURSOR <cr> and bingo! you have full mouse control. That's all there is to it. As you will probably want to use the mouse in all possible situations it is a simple matter to include MENU CURSOR in the autoexec.bat file just after installing MOUSE.COM. Having said that, you must of course have MENU.COM on the disk. MAKEMENU.COM is not necessary after the conversion process.

David Roberts Tel 0247-462564 Prestel MBX. 247462564

[illegible]

READER'S FORUM

THIS IS WHERE YOU ARE INVITED TO ASK ALL THOSE QUESTIONS THAT YOU HAVE
NEEDED ANSWERS FOR, OR TO MAKE ANY COMMENTS, OR ???

FROM:- 6 Plantation Road, Ballymacormick Road, Bangor,
Co Down. BT19 2AF

Dear Gordon,

Its nice to be back in the fold of NATGUG again after a short lapse.
(Aftermath of selling the Model 4 for an Amstrad PC.)

It is some time since I wrote an article for NATGUG so I am enclosing a disk and hardcopy of a review of a nice little utility called XTREE for consideration. As time permits, I may produce something on Supercalc 4 Release 1.1.

I'm afraid I haven't the time to match the volume of my friend Anon way out there in the Hebrides but I hope this will help swell the magazine. Incidentally, congratulations on the quality and quantity.

Yours Sincerely, David Roberts.

=====

FROM: 46 Field Way, Chalfont St. Peter, Bucks. SL9 9SH (0753) 882982

Dear Gordon,

Additional Drives on TRS-80 Model IV

My TRS-80 Model IV has now been giving me good service for about two years. There is however, one feature in which the system would be improved and maybe one of our members could give me hint about what to do.

When I changed my Model I, I had 2 Cumana disk drives and these I transfered to the Model IV but they never worked properly.

I find attempting to copy from the internal drives to the Cumanas or to run programs from them, either difficult or impossible. This is so much so that I am contemplating disposing of these additional drives. However I would be glad to keep them if I could get them to work reliably as they did on the Model I.

Any suggestions would be gratefully received either in the NATGUG News or by phone on (0753) 882982 or by letter to the above address.

Eric Brandes

=====

From: The Oaks, 12 Grailands, BISHOP'S STORTFORD, Herts, CM23 2RG

Dear Gordon,

Although I have had a Tandy for about 10 years now, I am not a writer of programs at all, just a user of the computer and what it can do for me. I use it with programs written by other people either specially for my purpose or standard ones like Scripsit, modified by Geoff Smith.

I upgraded my original Model 1 with cassette, through expansion boards to finally two external disc drives. The next stage was to a Blandford modified Model 111 with two double sided quad density discs. This was inspired by seeing one at a Milton Keynes workshop. This reduced my data discs from 14 down to 2. However growth has now got me up to 4 discs. I have always had 2 back-up of each disc so with such large discs back-up alone is a slow business.

I am now having to consider with the still increasing data bank that a hard disc of at least 10 MB and probably 20 MB will have to be the next step.

During this time I progressed through the little Tandy printer to Anadex, then to Epson MX80 and now to an Epson LQ800.

My reason for writing this letter is to ask whether someone can give me advice as to what programs I ought to be searching for to accompany my, by now, fairly large data bank. I guess that the next computer will be IBM compatible.

The data bank consists of the details of a car club to which I belong and records the names, addresses and cars of the members. I am registered under the Data Protection Act for this purpose. The reason for the continual growth is that, for reasons relevant to the club's policy, we do not remove a members name or cars even when he no longer pays his subscription or even when he dies. We also do not remove his cars even when he no longer owns them. Over the years the number of new members has always exceeded the number of resignations. Recently the rate of growth has increased so that the current total is about 1 200 names. The number of members past and present is about 4 000 and about 6 000 cars. To make this at all manageable the owners in Great Britain are split onto two discs one from A to L and the second from M to Z, the overseas owners are now also on two discs.

To simplify the description of my procedure let us imagine that one name list, say GBA, is in complete alphabetical order. If I add new members they can only be added to the end of the list. After I have added a few more names the listing is slow from the subsequent search procedure. A letter such as A will have about 120 names on it while B has had to be split into A-L & M-Z with about 200 names on each. Laurie Shields wrote a sort program for me some years back which I then have to use to put the whole series into alphabetical order again.

As you can see to sort the Great Britain lot needs 30 runs through and the overseas about the same.

I now ask the question I have been leading up to with this long preamble. Is it possible to write or buy a program which will insert a name such as Mr & Mrs A M Brown between Mr & Mrs A C Brown and A S Brown so that I no longer have to sort? Also now when I want to call up A S Brown I have to call up Brown and reject all the Browns before A S as "wrong" before A S Brown comes up. Could I call up A S Brown and let the program search for only this one Brown? The combination of suffixes for Brown is fairly large - - - Mr & Mrs, Mrs & Mr, Mrs & Mrs, Dr, Dr & Mrs, Capt, Admiral & Mrs and so on.

There are a number of rather special programs I use to abstract certain details from the whole data bank which possibly cannot be much improved, these two are the ones which I use most often and would save me the most time. The programs are written in Basic with Newdos 80.2 ZAP 11.

My next query covers a slightly different aspect and is to do with the Epson LQ800. As you know one of the features of this printer is the 24 pin head which produces for nearly everything a letter indistinguishable from a daisy wheel. The other feature which is one of the reasons I bought it is the availability of alternative fonts which you can plug in. For reasons to do with the final stages of the work I am doing for the club and other things I have selected a sans serif font which is also proportionally spaced

As was mentioned in Derek Traylor's article on printers in Natgug News Vol 8 Issue 11/12 using Scripsit with proportional spacing produces a ragged right margin even when using the right justification command. I am aware why this happens, you need a very large memory to cope with the large variations in spacing from commas to the letter m. The variation covers about 5 different widths. Since the Model III when Scripsit is loaded has only about 30K RAM left this is no way near enough. As was also mentioned in Derek's article, columns are difficult with proportional spacing but can be done by the use of a series of codes to the Epson.

What word processing program, which I would almost certainly have to use with a hard disc and about 500K RAM, would cope with proportional spacing to give right hand justification and easy column work? I would guess that "wysiwyg" would almost be a necessity and would certainly make the work a lot easier

With a data bank of this size I need quick access and no sorting, my typing speed is slow but it is the access time which I need to be faster.

The word processing program must be able to call up the Epson commands to give me Emphasised, Double Strike, Condensed, Expanded, Italic with the ability

Unfortunately I missed Blandford and being in Germany means that I will miss Swindon as well, the worst news is that an impending trip to a series of islands in the South Atlantic means that I will miss the next Swindon as well. This is not good as I will have been out of actual contact with group members for some considerable time. Hopefully Swindon next autumn is still on for me.

I have received no mail from MSDOS users this time so have no questions to answer but I hope to return to a mass of problems and solutions in November.

Good Luck and Happy "MSDOSing"

Dave Holman, 160 London Road West, Bath, Avon. BA1 7QU

*(Re: David's apology, I have written to David to say that this was not necessary as the reason it did not go in, was not that the article did not arrive in time, but that I had to send the copy to the printers early. Ed. **)*

USING SYS13/SYS

In Vol. 9, Iss. 3, page 20, of NATGUG News, there was an article on the above by OPTICA. Here are my ramblings on the same subject. Whilst writing I will be following the instructions on another Model 4, just to confirm that they will work.

To save anyone doing a lot of reading, and then finding that they are not able to use this, can I now say that this will only work with TRSDOS 6:2:xx and LS-DOS 6:3:xx up. Also I shall be keeping to a very narrow path whilst writing, I would get lost myself if I did not. If anyone understands more of this than me, please feel free to take detours. I shall go through this in stages so that anyone wanting to branch off can do so, you may find out more than I have. I am not a DOS expert, but I am willing to learn whilst listening to others.

My natural inquisitive instinct was aroused, when I read, against a reference to something called an "EFLAG\$", "that this byte is used to indicate the presence of an Extended Command Interpreter (ECI) program in the SYS13/SYS slot". Turning to the Mod. IV Owner's Manual revealed nothing. (Because it was written before TRSDOS 6:2 ?). The Technical Manual for TRSDOS 6:2, page 262, has 5 lines which meant nothing to me, and page 190 tells you to remove the file if you have not implemented an ECI. Have I missed something somewhere ?

Over a period of time I found out the following, which is a very brief synopsis of all my workings in the land of the unknown.

BACKUP your system disk to include SYS13/SYS. This backed up system disk should now be put in drive :0. REBOOT and then type DIR SYS13/SYS:0 (S) <ENTER>, you should see that the "File Size" is 1.5K. Make at least 28K of free space on the disk. Type PURGE :0 <ENTER> and answer <Y> to a number of files, <BREAK> when you have removed enough files (I answered <Y> to: BREF/CMD, DOS/HLP, HELP/CMD and TAPE100/CMD), then type FREE :0 <ENTER> as a check. Using Model IV, VC/CMD - which I think most people have - on a disk in drive :1, (I am using this file as an example only, as almost any /CMD file can be copied to SYS13/SYS,) type in the following:

```
COPY VC/CMD:1 TO SYS13/SYS.LSIDOS:0 (C=N) <ENTER>
```

As a check type DIR SYS13/SYS :0 (S) <ENTER>, you should now see "File Size" as 27.00K. Still with me ?. Type, DIR :0 (S,I) <ENTER> you should not have found VC/CMD on your disk. NOW at DOS READY, type <*> <ENTER>, yes just an asterisk and <ENTER>, VisiCalc should arrive on your screen. You have just Executed an ECI Program.

This is where it gets a bit more difficult for me. The above is not going to move mountains, but please keep reading. Whilst in VisiCalc <ENTER>, </>, <S>, <Q>, <Y>, you should now be at DOS Ready; you could have just rebooted. At DOS Ready type MEMORY (ADD="E") <ENTER>, the forth line on the screen should read:-

```
X'006E' = 110 (X'0000', ... etc.
```

What we are interested in are the first two X'????'s, this shows that the EFLAG\$ is not set. The EFLAG\$ being at MEMORY ADDRESS X'006E' and is X'00'. Again at DOS Ready type:-

```
PATCH SYS0/SYS.LSIDOS:0 (D00,7A:FF:F00,7A:00) <ENTER>
```

This has now permanently zapped the EFLAG\$ byte in a file called SYS0/SYS and since SYS0/SYS is always in memory from bootup you will now have to reboot to effect this zap, so now please REBOOT. If you did that, VisiCalc should have appeared on your screen. This is another way of executing an ECI. Whilst SYS0/SYS remains zapped on this disk, on each BOOT'ing, DOS will want to load SYS13/SYS, you will not see "DOS Ready" again.

Just a comment here before moving on. In OPTIKA's article mention was made of having two system disks. On one disk LESCRIPT/CMD has been copied to SYS13/SYS and on the other, VC/CMD. OPTIKA has also AUTO'ed DDUTY/CMD. On bootup DDUTY is loaded, and also SYS13/SYS. OPTIKA then changes system disks and upon going into the second partition of DDUTY, a second SYS13/SYS is loaded. (you could stay with the first system disk and upon then going to the second partition of DDUTY, that SYS13/SYS would load again. A quick way of

getting two programs up and running with a few key strokes. N.B. LeScript, when in SYS13/SYS has to be zapped at record X'12', find "LESCRIPT/CMD" and change to SYS13/SYS, as LeScript looks for its own name before loading, or you could leave the file LESCRIPT/CMD on the disk. Another N.B., is that when using LeScript with DDUTY, note the comment on Page 97 of the LeScript manual.

Now back again. Whilst VisiCalc is on the screen please type the following </>, <S>, <E>, and in the edit line type, MEMORY (ADD="E") <ENTER>, you will now see that the byte at memory address X'006E' has changed to X'FF'. Hit any key to reformat the screen. We now have one big problem, try exiting VisiCalc by typing, </>, <S>, <Q>, <Y>, as we did above. So something just wants to keep loading the ECL, its that EFLAG\$ byte we zapped, so with VisiCalc now on screen type, </>, <S>, <E>, and in the edit line, MEMORY (ADD=X'6E',B=X'00'), you will now see that we have rezapped MEMORY ADDRESS X'6E' to X'00'. Hit any one key to reformat the screen and exit VisiCalc as above, and you should get to DOS Ready.

The next stage that I have got to, and here I give my thanks to Peter Knaggs who worked up a zap for me, for VC/CMD, is that I do not remain in that endless loop on exiting VC/CMD. This zap is for VC/CMD, record X'0068' and starting at byte 00. You should find:

02 02 00 30 6D B6 6D B6 6D

change these nine bytes to: 01 03 6E 00 00 02 02 00 30. This zap resets address X'6E' in memory before VC/CMD returns to DOS Ready. If someone could work up and send me an Lcode patch to cover the above, I will publish that. I would also publish any patches to other /CMD files that resets the EFLAG\$ memory byte, which I receive. I would hope that the above will lead others to experiment and send their findings to me for publication.

Whilst we have been doing the above it has been necessary to enter passwords for files. If you, like me, wish to save on typing, type the following and <ENTER>:

For TRS-DOS 6:2; PATCH SYS2/SYS.LS1DOS (D02,33=18:F02,33=28)

For LS-DOS 6:3; PATCH SYS2/SYS.LS1DOS (D02,50=18:F02,50=28)

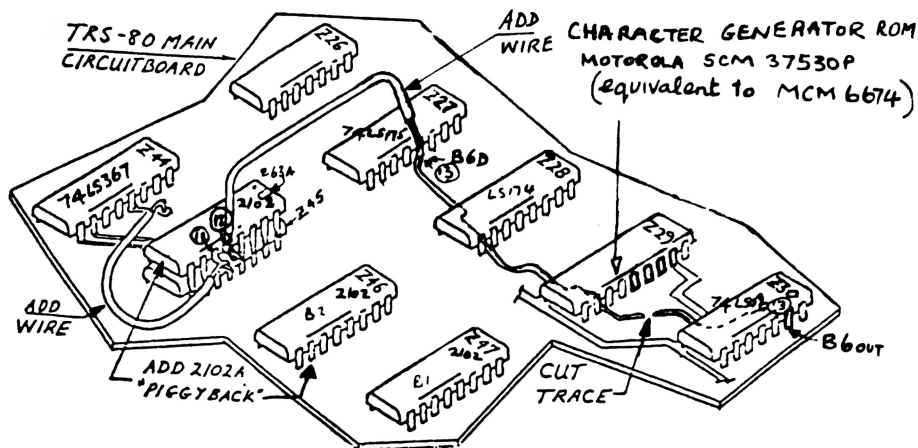
The following are for LS-DOS 6:3 only. Sorry about this one Roy. Who would prefer to use "KILL" not "REMOVE" ? Then type and <ENTER> the following with NO passwords anymore: (you did do the above PATCH didn't you ?

PATCH SYS1/SYS (D02,15="KILL ":F02,15="REMOVE")

(there are two spaces after "KILL ")

Those who are not bothered about having the correct time and do not wish to see "Time HH:MM:SS ?" after booting:

MODEL I LOWERCASE MODIFICATION

Notes:

- 1) The character generator chip shown on the drawing is often in a socket and produces decoders on lower case.
- 2) One of the 2102 video memory chips is also socketed on later keyboards. Piggyback two 2102L chips except for pins 11 and 12 on the top one. Attach thin wires to these two pins and fit the combination in place of the 2102N in the socket. This is much easier than soldering "on the board".
- 3) Cut the track that comes from the 74LS02 (Z30) and goes between pins 11 and 12 on the 18 character ROM (Z29).
- 4) Solder the wire from the piggybacked chip, pin 11, to pin 13 of the 74LS367 (Z44). Solder the wire from pin 12 to pin 13 of the 74LS175 (Z27).
- 5) Reassemble the keyboard and switch on. Load one of the lower case drivers, or a word-processor that has one built in, and Viola or whatever her name is ?

Dave Martin. G810J. 37 Longworth Avenue, Tilehurst, Reading,
Berks. RG3 5JU. (0734) 413688

LOCAL CLUB NEWS

BOURNEMOUTH

1st & 3rd WEDNESDAY at Kinson Community Centre 7.30pm.
Ring Carl Rabe on (0202) 730617

MS.DOS User Group. 2nd & 4th Wednesday 8 pm in rear room Sports Centre,
Hurn. Contact: Mark Austin (0202) 428856.

CHELMSFORD

1st WED. of each month at 7.30pm. Contact: Richard Creak, Woodcote,
59D Little Baddow Road, Danbury, Chelmsford, CM3 4NT. (0245) 413725

MILTON KEYNES

Alternate Sundays, October to March. Brian Pain (0908) 564271

LONDON

Nth East London-80; 77 Old Church Road, Chingford, LONDON. E4 6ST.
Meets first Sunday every month.

Nth West London; Central Common Rooms, Northwick Park Hospital. 1st
Sunday of month 10.30-1.30. Geof Smith, 01-950 6345 after 8 pm.

SUDBURY

2nd Wed. at:- 3a Gainsborough Street. John Kilpatrick (0787) 79504.

SWINDON WEEKEND '88 next meeting March 11th-13th.
Do not wait - book now !!! ring (0793) 28282

If your club is not mentioned above then its your fault, not ours !
For your FREE advert, write with details to:- Gordon Collins, Editor.