REGISTERED BY AUSTRALIA POST PUBLICATION No. TBH 0917 CATEGORY "B"

ISSUE NO.
3 - 4

DATE
JAN - 1986

ANNUAL SUBSCRIPTION	
AUSTRALIA	$20.00
OVERSEAS	$25.00
OVERSEAS AIRMAIL ...	$30.00
YEAR BOOK	*$20.00*

CONTENTS

NEWSLETTER CORRESPONDENCE

S.A.U.G.,
P.O. BOX 191,
LAUNCESTON SOUTH,
TASMANIA, 7249.

(003) 442493

LIBRARY CORRESPONDENCE

S.A.U.G. LIBRARY,
1 CONRAD AVENUE,
GEORGE TOWN,
TASMANIA, 7253.

(003) 822919

INTRODUCTION

I hope you all entered 1986 in high spirits.

I also hope the prize winners in the S.A.U.G. software competition have received their prizes. As yet not one has bothered to contact us and say all arrived well.

1985 YEAR BOOK

Advanced orders are now being taken for the NEW Year Book. This book will be printed in limited numbers only and to ensure you get yours please order early. $20.00 is all we ask. the Year book will be available towards the end of FEBRUARY.

NOTE 1983/84 Year book is not available now. we will only reprint if enough interest is generated.

The cost of the Year Book is extreemly high and the group cannot afford to outlay such money unless we know we can recoup.

Well thats all from me hope you enjoy this months newsletter.

X'PRESS OWNERS

There is a slight bug in the X'press MSX-DOS. Most owners will not be affected, however a fix will be given next month.

Exploring Basic Pt-15

by L.A. Dunning.

This part deals with disk drives and how to make the most of them from basic. It does not deal with standard ways of using disk statements as these are already fully covered in users manuals and the like. There are a number of approaches which are not quite apparent upon reading such books.

SEQUENTIAL FILES

The enormous advantage that a disk drive has over a cassette drive, is not so much time, as ease of access. In both cases you put the medium inside the drive and read/write from it, however with disk drives you don't need to press any keys, rewind any tape or similar physical tasks, just sit back and listen to the head eat through the disk. Any sector of a formatted disk can be accessed to be read or written to.

Now let's consider the means via basic of doing this. Firstly, we have the statements that mimic a cassette drive. These are OPEN-FOR-AS, PRINT#, INPUT#, SAVE, LOAD and CLOSE. Now a file on disk is created in the same manner regardless of content, be it a binary/ASC save or data file. In either case, the file starts at the first sector on a track. It continues using sequential sectors until all sectors on the track are used, then a new and unallocated track is picked and the sequence continues until you either run out of room on the disk or the file ends. A linked list on the File Allocation Table (FAT) keeps track of which track follows which on the the file. If a file uses less than 17 sectors on a track, the remaining sectors cannot normally be used, unless for that file.

PRINT# actually transfers the data from its source not to the disk but to a buffer. When the buffer is full or the file closed, the contents of the buffer are dumped to a sector on the disk, the next sector is chosen and pointers reset to the start of the buffer. This process continues with the buffer being filled with data and then dumped to disk, until the last dump is made.

Since PRINT# and INPUT# are designed to imitate the cassette functions, it means that the data read/written advances along the sectors in "one direction". That is, it cannot read/write over previous information without either opening another file or closing and reopening the existing file.

RANDOM FILES

Now we come to statements that do not imitate cassette functions. These are OPEN-AS, GET, PUT, DSKO# & DSKI. The main difference with a random file is that the order of reading/writting to the disk does not have to be in sequence. When the disk is read or written to, only that sector is affected, not any previous sectors.

The structure of a random file on the FAT is identical to that of a sequential file. Now each record in a random file has a sector number, given in the second argument on a GET/PUT statement. The first track of a random file contains the first 17 records of that file, the 2nd track contains records 18 to 34, the 3rd contains records 35 to 51 and so on. When you GET or PUT a higher record than previously used, the system checks to see if that record is contained in the tracks already allocated to the file. If it is, it merely alters the end of file marker on the FAT and reads/writes to the sector. If it is not, extra tracks are added to the file until the record can be accomodated. The contents of those new tracks, with the exception of the accessed sector, are not altered because of this.

Since each number in a GET/PUT statement equals a sector, the highest number you can go to on a normal BASIC disk is 612 sectors. On a disk that has the system tracks deleted (see previous issue) you can go up to 663 records. In either case, you would have just the one random file + directory track on disk.

DSKI$ & DSKO$

Both DSKI$ and DSKO$ statements function in a similar manner as GET and PUT however you do not need to open a file to use them. Whereas with GET and PUT, a single record number is required, DSKI$ and DSKO$ require the numbers of disk, track and sector to be accessed. Both these statements work by reading and writing between buffer #0 and the sector in question. Both allow you to asign string variables, either to be dumped to or read. In this case the variable is being read to or from the buffer. The buffer is 256 bytes long and a variable is at most 255 bytes long. The last byte is ignored or lost!

The solution is to FIELD some dummy variables to buffer #0. Something like:

```
        FIELD #0,128 AS B$(0),128 AS B$(1)
```

When using DSKO$ to write back to disk, dump the data to the buffer and then, when using the statement, don't include a string expression. This enables you to write a full 256 bytes to a sector.

Since you can use these statements to access any sector on a formatted disk, you don't need a directory track. This gives you an extra 17 sectors to access, even though you can't tell using FILES what is on the disk. There are two disadvantages to using DSKI$ and DSKO$ statements instead of GET and PUT. You need physical parameters instead of just a record number. This could be a bonus if you want to encode data by its location, but a disadvantage in that it has to be calculated first. You also have to use buffer #0. This buffer, unlike buffers #1 and above, is used and cleared by the system on a regular basis and so is unreliable. The solution here is to use a second buffer and then dump the contents to/from buffer #0 before each access.

BUFFERS

Each buffer occupies 265 bytes in memory. The starting location of each buffer can be found by using VARPTR(#n) where n is the number of the buffer. The first 9 bytes are used for housekeeping by the system. The remainding 256 bytes (not 128, as stated elsewhere) are the contents of the buffer proper. Let's examine the first 9 bytes in detail.

Byte 0 is used to indicate the file mode. The values are (in octal):

001	Sequential File - INPUT
002	" " - OUTPUT
010	" " - APPEND
004	Random File
020	Delete File
200	Binary Save - BASIC PROGRAM

The first four values of this list are fairly staightforward. A Binary save is one where the basic program is saved in tokenised form instead of ASC form. What does Delete File mean? I don't know. Perhaps it is used when truncating or deleting files but such a pointer would hardly seem neccesary. Perhaps someone else would like to enlighten me on this. I'll certainly pass on the information.

Byte 1 points to the first entry for the file on the FAT. This is also the first track used for the file.

Byte 2 indicates the last track accessed by the buffer. This is not neccesarily the last track used by the file. It also points to the current entry used on the FAT.

Byte 3 indicates the last sector accessed on the track indicated by byte 2.

Byte 4 indicates which disk is used by the buffer.

Byte 5 indicates the size of the buffer. This is normally set 0 which indicates a buffer size of 256. A shorter size might be given at the close of a sequential file (which is given by a control-Z on the disk sector.

Byte 6 gives the current PRINT position in the buffer. This is the location that PRINT# will start printing at or INPUT# read to.

Byte 7 is a flag indicating the following (in octal):

002	Buffer is empty
004	PUT has been done and no GET done yet
010	Buffer has been altered by PRINT#
020	File is write protected - "P"
100	File is set to read after write - "R"

The buffer empty flag is needed to distinguish between being full and empty, which are both represented by 0 in byte 6. The PUT flag is there because doing a PUT before before either PRINT#/INPUT# are supposed to give errors unless a GET is done (see later).

Byte 8 indicates the number of bytes in the buffer/file since the last TAB or unsuppressed PRINT#. An unsuppressed PRINT# is one without a ";" behind it. This produces an extra carraige return (0DH) and linefeed (0AH) in the buffer. A TAB produces a 09H character in the buffer. These are dumped to/from disk as is.

MANIPULATING BUFFERS

According to certain manuals, you can open the same file under different names and then manipulate that file in different ways. This is almost true. If you try to open two files with the same name you will get an error. If you try to open two files with the same buffer you will also get an error. There are two ways around this.

Firstly, you could alter the Directory Table (DIR, see last issue) so that a file had two entries. Both would point to the same entry on the FAT. You could then open two seperate buffers on the same material, using INPUT for one and OUTPUT or APPEND for the other. The problem here, is that the results would be unpredictable. The physical records of the file in question would be dependent upon the last buffer to be closed. You could lose the file entirely by having bad entries on the FAT.

The second way is to only open a random file buffer for the file. Instead of OPEN"filename"FORactionAS#1, use OPEN"filename"AS#1. The reason is, that both PRINT# and INPUT# both work in SV random files. The GET statement also works for sequential files, although PUT does not. Whenever a GET or PUT is performed, bytes 2 & 3 in the buffer point to the last sector accessed and byte 6 (the buffer offset) is set to 0. PRINT#s and INPUT#s made after this start at the beginning of the buffer with the sector as set by the last GET/PUT.

The advantage here, is that you can do both an INPUT# and PRINT# in the same buffer.

You may also use GET to position the buffer in sequential files. The only problem is that GET will dump the sector to the buffer and destroy any contents therein. Also, performing a GET greater than the length of the file will produce and error. The solution is to use LOF(#n) to determine the length of file, where n is the buffer used. This gives the number of records in the file.

VARIABLE FILE LENGTHS

One thing lacking with SV random files is the convenience of having variable lengths or records. On other basics you can set your record length between 1 and 256 records. On the SV, it is set to 256. If you have a random file and a record length less than 256, then the excess bytes are wasted if you only use GET and PUT. Assuming you only used 129 bytes for each entry, then the remainding 127 bytes are blanked out and lost.

There are several ways around this. If the file was one that was only to be read, you could use a combination of GET and INPUT$() to read the data. Assume the record length to be "L" where L is less than 256 and the record number to be "N". Use PRINTUSING# to place the records on disk originaly with the length of the control string being equal to L. To retrieve data, do:

```
            GET#1,(N*L\256)
            POKE VARPTR(#1)+6,(N*L)MOD256
            D$=INPUT$(L,#1)
```

To record data, do the first two statements and then substitute

```
            PRINTUSING#1,DX$;D$,nn... ...nn;
```

where DX$ is the control string for PRINTUSING and D$/nn are the variables printed. The first one will work for reading data, the second should work all the time however I do not guarantee it.

The second way is to subdivide the buffer into subfiles. If the record length is 128 records or less, you can fit 256\L records in a sector. Assuming C to be the number of whole records you can fit into a sector and the original statement to be something like:

```
  FIELD#1,L1asD$,L2asE$,L3asF$
```

then you alter it to be:

```
  FIELD#1,L1asD$(0),L2asE$(0),L3asF$(0),L1asD$(1),L2asE$(1),L3asF$(1)...
```

and so on until you reach D$(C-1),E$(C-1) & F$(C-1). This might take more than one statement and you can use dummy variables to pad out any further FIELD statements.

Assuming N to be the record to be accessed you do the following statements:

```
                GET#1,(N\C)
                P=N MOD C: D$=D$(P):E$=E$(P):F$=F$(P)
```

here D$, E$ & F$ are the variables used by the program. To alter the record, you must do the above and also:

```
                LSETD$(P)=D$:LSETE$(P)=E$:LSETF$(P)=F$
                PUT#1,(N\C)
```

The important thing is, that the whole sector must be read before altering any part of it otherwise information contained in the other records will be lost.

There is no reason why different parts of a buffer have to contain duplicate records. You could have two seperate files sharing the same physical record, by using dummy strings in FIELD statements. This is a variation on the above technique. Assuming you have file A that requires 100 bytes per record, file B that needs 80 bytes per record and file C that also requires 80 bytes then you could open a file called "DATA" and do the following.

When you require File A:

```
FIELD#1,100asA$,156asD1$
```

When you require File B:

```
FIELD#1,100asD1$,80asA$,86asD2$
```

When you require File C:

```
FIELD#1,180asD1$,80asA$,6asD2$
```

here A$ is the variable used to read the file and D1$ and D2$ are dummy variables. Once again, you have to GET a record before PUTting it to ensure that the other files aren't destroyed. The difference between this and the previous technique is that it is designed to be used by seperate programs.

All the above attempt to conserve space on the disk at the cost of extra handling, still you can't have everything. It's an old question of speed vs versitility.

IPL

This months listing is IPL. This is the program that I normally use with the IPL command when the disk is booted up. It produces a title for the disk, executes a FILES and sets colour and function keys. When all this is done, it NEWs itself. This uses one track on the disk and no memory. I normally save this as the first program on disk and call it "ipl" so it is immediately recognizable. An interesting variation is to call it "   " and save it as an ASC file. This produces no name when FILES is done, only a size listing. No doubt you have your own version of such a program. I'm including it for those who are new to disks; modify it for your own needs.

NEXT MONTH

In next months part, I'll show how to handle possible and probable errors using disks and how to repair files.

IPL

by : L.A. Dunning

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV. 84.) or The Year Book.

```
BG      10 REM  IPL
JJ      20 REM  Save program to disk as "ipl"            then type:
MJ      30 REM  IPL"RUN"+CHR$(34)+"1:ipl"                and reboot computer
BO      40 MAXFILES=4:RS$=CHR$(27)+"p":RE$=CHR$(27)+"q":QU$=CHR$(34):CL$=CH
           R$(12):CR$=CHR$(13):CU$=CHR$(21)
CA      50 COLOR15,1,3:CLS:READA$:PRINTRS$:PRINTUSING"<< \"+SPACE$(23)+"\ F
           REE ## >>";A$;DSKF(1):PRINTRE$:FILES
EK      60 DATA <---Name of Disk------>
IN      70 KEY 1,CU$+"FILES"
CO      80 KEY 2,"LOAD"+QU$+"1:"
FH      90 KEY 3,"SAVE"+QU$+"1:"
EB     100 KEY 4,CU$+"list"
HM     110 KEY 5,CU$+"run"+CR$
IG     120 KEY 6,CU$+"color15,1,5"+CR$
GH     130 KEY 7,"MERGE"+QU$+"1:"
AF     140 KEY 8,"KILL"+QU$+"1:"
EB     150 KEY 9,CU$+"list."+CR$
EG     160 KEY10,CL$+"run "+CR$
DG     170 NEW
END
```

MASTER MIND

by : G. Trevathan

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV. 84.) or The Year Book.

```
FG    100 REM □□□ from SVI-MSX USER GROUP □□□
EM    110 REM □□□ SOFTWARE LIBRARY,  1986 □□□
AM    120 F=2
BK    130 COLOR15,1,1:SCREEN1:BEEP
BE    140 LOCATE1ØØ,12Ø:PRINT"Mastermind"
CC    150 LOCATE75,154:PRINT"By ... G.Trevathan"
BK    160 ONSTOPGOSUB170:STOPON:IFPEEK(&H4788)=&H3ETHENDEFUSR=&H4788:A=USR
          (Ø)ELSEFORI=ØTO1ØØØ:NEXT
EC    170 COLOR 2,1,1:SCREEN2:STOPOFF
AM    180 FORX=1TO3
CE    190 FORY=2TO15:COLORY
BJ    200 LOCATE5Ø,5Ø:PRINT"MASTER"
CG    210 LOCATE5Ø,9Ø:PRINT" MIND"
FM    220 FORZ=1TO1Ø:NEXTZ
DH    230 NEXTY
DH    240 NEXTX
BI    250 SCREEN1
BJ    260 LOCATE3Ø,1ØØ:PRINT"DO YOU REQUIRE INSTRUCTIONS [Y/N]"
AC    270 I$=INKEY$:IFI$=""THEN27Ø
BJ    280 IFI$="Y"ORI$="y"THEN31Ø
PO    290 IFI$="N"ORI$="n"THEN48Ø
AL    300 GOTO27Ø
CL    310 COLOR1,15,15:SCREEN1
DI    320 LOCATE1ØØ,Ø:PRINT"RULES"
GE    330 COLOR2:LOCATE1Ø,2Ø:PRINT"This is a game of cunning and logic
              played by me and you."
DI    340 LOCATE1Ø,45:COLOR4:PRINT"I will choose a combination of 5
              colours and place them behind a green      box at the top of
          the screen."
CK    350 COLOR6:LOCATE1Ø,8Ø:PRINT"You must try to duplicate the exact
              colours and their positions."
BD    360 LOCATE1Ø,1Ø5:COLOR7:PRINT"To choose your colours look at the
               colour chart on the left hand side of      the board. Wait
          for the prompt at the      bottom."
DB    370 LOCATE1Ø,145:COLOR1Ø:PRINT"Now key in the numbers (colours) whic
          h     you have chosen. I will then tell you      how many you ha
          ve right."
DF    380 COLOR13:LOCATE1Ø,18Ø:PRINT"Press any key to continue"
AA    390 I$=INKEY$:IFI$=""THEN39Ø
BB    400 SCREEN1
GH    410 COLOR1:LOCATE1Ø,1Ø:PRINT"Look at the small box at the side of
              board. There are 5 peg holes here         which correspond
          to the hole numbers       where your colours are shown."
IP    420 COLOR3:LOCATE1Ø,5Ø:PRINT"If you have guessed correct then the
              circle will be cleared.  If you have       guessed the corre
          ct colour but in the       position the circle will have been
               filled in."
EE    430 COLOR4:LOCATE1Ø,1ØØ:PRINT"If your guess is completely wrong the
               circle will remain the same. Now enter     your next guess.
          "
EN    440 COLOR1Ø:LOCATE9Ø,15Ø:PRINT"Good Luck!!!"
```

```
AO    450 COLOR13:LOCATE60,180:PRINT"Press any key to begin."
AI    460 I$=INKEY$:IFI$=""THEN460
CE    470 COLOR15,1,1
BJ    480 SCREEN1
EA    490 IFF=1THENC=C+1
EH    500 IFF=0THENA=A+1
DA    510 IFA=5ORC=5THEN2610
GD    520 LINE(60,11)-(208,182),15,B
EA    530 LOCATE90,2:PRINT"MASTER MIND"
BP    540 LINE(110,20)-(200,30),2,BF
AO    550 LINE(103,40)-(103,160),15
BA    560 LINE(60,40)-(208,40),15
AP    570 LINE(60,60)-(208,60),15
AO    580 LINE(60,80)-(208,80),15
CJ    590 LINE(60,100)-(208,100),15
BL    600 LINE(60,120)-(208,120),15
BI    610 LINE(60,140)-(208,140),15
BF    620 LINE(60,160)-(208,160),15
CJ    630 CIRCLE(65,55),2,15
CL    640 CIRCLE(73,55),2,15
CN    650 CIRCLE(81,55),2,15
CE    660 CIRCLE(89,55),2,15
CG    670 CIRCLE(97,55),2,15
AP    680 LOCATE79,44:PRINT"6"
CB    690 CIRCLE(65,75),2,15
CD    700 CIRCLE(73,75),2,15
BA    710 CIRCLE(81,75),2,15
CH    720 CIRCLE(89,75),2,15
CJ    730 CIRCLE(97,75),2,15
AH    740 LOCATE79,64:PRINT"5"
CE    750 CIRCLE(65,95),2,15
CG    760 CIRCLE(73,95),2,15
CI    770 CIRCLE(81,95),2,15
BP    780 CIRCLE(89,95),2,15
CB    790 CIRCLE(97,95),2,15
AB    800 LOCATE79,84:PRINT"4"
AK    810 CIRCLE(65,115),2,15
AI    820 CIRCLE(73,115),2,15
AG    830 CIRCLE(81,115),2,15
AP    840 CIRCLE(89,115),2,15
AN    850 CIRCLE(97,115),2,15
DB    860 LOCATE79,104:PRINT"3"
AO    870 CIRCLE(65,135),2,15
AM    880 CIRCLE(73,135),2,15
AK    890 CIRCLE(81,135),2,15
AI    900 CIRCLE(89,135),2,15
AG    910 CIRCLE(97,135),2,15
DD    920 LOCATE79,124:PRINT"2"
AH    930 CIRCLE(65,155),2,15
AF    940 CIRCLE(73,155),2,15
AD    950 CIRCLE(81,155),2,15
AM    960 CIRCLE(89,155),2,15
AK    970 CIRCLE(97,155),2,15
CK    980 LOCATE79,144:PRINT"1"
AF    990 CIRCLE(115,50),2,15
CO   1000 CIRCLE(135,50),2,15
DB   1010 CIRCLE(155,50),2,15
```

```
DE    1020 CIRCLE(175,5Ø),2,15
DH    1030 CIRCLE(195,5Ø),2,15
CO    1040 CIRCLE(115,7Ø),2,15
DB    1050 CIRCLE(135,7Ø),2,15
DE    1060 CIRCLE(155,7Ø),2,15
DH    1070 CIRCLE(175,7Ø),2,15
DK    1080 CIRCLE(195,7Ø),2,15
DB    1090 CIRCLE(115,9Ø),2,15
CJ    1100 CIRCLE(135,9Ø),2,15
CM    1110 CIRCLE(155,9Ø),2,15
CP    1120 CIRCLE(175,9Ø),2,15
DC    1130 CIRCLE(195,9Ø),2,15
AC    1140 CIRCLE(115,11Ø),2,15
AF    1150 CIRCLE(135,11Ø),2,15
AI    1160 CIRCLE(155,11Ø),2,15
AL    1170 CIRCLE(175,11Ø),2,15
AO    1180 CIRCLE(195,11Ø),2,15
AJ    1190 CIRCLE(115,13Ø),2,15
AB    1200 CIRCLE(135,13Ø),2,15
AÉ    1210 CIRCLE(155,13Ø),2,15
AH    1220 CIRCLE(175,13Ø),2,15
AK    1230 CIRCLE(195,13Ø),2,15
AF    1240 CIRCLE(115,15Ø),2,15
AI    1250 CIRCLE(135,15Ø),2,15
AL    1260 CIRCLE(155,15Ø),2,15
AO    1270 CIRCLE(175,15Ø),2,15
BB    1280 CIRCLE(195,15Ø),2,15
FB    1290 LOCATE15,1Ø:COLOR15:PRINT"COLOUR"
EJ    1300 LOCATE18,2Ø:COLOR15:PRINT"CHART"
EE    1310 CIRCLE(4Ø,5Ø),5,2
EJ    1320 CIRCLE(4Ø,7Ø),5,4
EO    1330 CIRCLE(4Ø,9Ø),5,6
AD    1340 CIRCLE(4Ø,11Ø),5,7
DC    1350 CIRCLE(4Ø,13Ø),5,1Ø
DE    1360 CIRCLE(4Ø,15Ø),5,13
DF    1370 CIRCLE(4Ø,17Ø),5,15
DI    1380 PAINT(4Ø,5Ø),2
DF    1390 PAINT(4Ø,7Ø),4
CH    1400 PAINT(4Ø,9Ø),6
GI    1410 PAINT(4Ø,11Ø),7
DF    1420 PAINT(4Ø,13Ø),1Ø
DF    1430 PAINT(4Ø,15Ø),13
DG    1440 PAINT(4Ø,17Ø),15
EC    1450 LOCATE15,47:PRINT"1 ="
EC    1460 LOCATE15,67:PRINT"2 ="
EC    1470 LOCATE15,87:PRINT"3 ="
AH    1480 LOCATE15,1Ø7:PRINT"4 ="
AF    1490 LOCATE15,127:PRINT"5 ="
AO    1500 LOCATE15,147:PRINT"6 ="
AM    1510 LOCATE15,167:PRINT"7 ="
BA    1520 LOCATE22Ø,1Ø:PRINT"SCORE"
AG    1530 LOCATE22Ø,2Ø:PRINT"CHART"
AE    1540 LOCATE22Ø,5Ø:PRINT"ME"
HC    1550 LOCATE22Ø,1ØØ:PRINT"YOU"
EF    1560 LOCATE24Ø,5Ø:PRINTA
BE    1570 LOCATE24Ø,1ØØ:PRINTC
EM    1580 T=1
```

```
EH    1590 S=15Ø
EB    1600 R=155
CH    1610 W=RND(-TIME)
EK    1620 FORB=1TO5
DP    1630 C(B)=Ø
BG    1640 C(B)=INT(RND(1)*15+1)
JB    1650 IFC(B)=1 OR C(B)=3 OR C(B)=5 OR C(B)=8 OR C(B)=9 OR C(B)=11 OR C
           (B)=12 OR C(B)=14 THEN163Ø
FA    1660 NEXTB
EI    1670 FORI=1TO5
HN    1680 LOCATE158,17Ø:COLOR1:PRINT"█"
CH    1690 LOCATE68,17Ø:COLOR2:PRINT"Colour for peg";I
FE    1700 I$=INKEY$:IFI$=""THEN17ØØ
AD    1710 IFVAL(I$)<1ORVAL(I$)>7THEN17ØØELSE172Ø
AG    1720 P(I)=VAL(I$)
OO    1730 IFP(I)=1THENP(I)=2:GOTO18ØØ
OM    1740 IFP(I)=2THENP(I)=4:GOTO18ØØ
OK    1750 IFP(I)=3THENP(I)=6:GOTO18ØØ
OJ    1760 IFP(I)=4THENP(I)=7:GOTO18ØØ
JH    1770 IFP(I)=5THENP(I)=1Ø:GOTO18ØØ
JK    1780 IFP(I)=6THENP(I)=13:GOTO18ØØ
JN    1790 IFP(I)=7THENP(I)=15:GOTO18ØØ
GO    1800 CIRCLE(185,171),5,P(I):PAINT(185,171),P(I)
FM    1810 I$=""
FJ    1820 FORY=1TO1ØØ:NEXTY
DJ    1830 CIRCLE(185,171),5,1:PAINT(185,171),1
FD    1840 NEXTI
BA    1850 CIRCLE(115,S),5,P(1)
BE    1860 CIRCLE(135,S),5,P(2)
BI    1870 CIRCLE(155,S),5,P(3)
BM    1880 CIRCLE(175,S),5,P(4)
CA    1890 CIRCLE(195,S),5,P(5)
FN    1900 PAINT(115,S),P(1)
FL    1910 PAINT(112,S),P(1)
FN    1920 PAINT(135,S),P(2)
FK    1930 PAINT(132,S),P(2)
FL    1940 PAINT(155,S),P(3)
FJ    1950 PAINT(152,S),P(3)
FK    1960 PAINT(175,S),P(4)
FI    1970 PAINT(172,S),P(4)
FJ    1980 PAINT(195,S),P(5)
FH    1990 PAINT(192,S),P(5)
CO    2000 P1=P(1)
CN    2010 P2=P(2)
CM    2020 P3=P(3)
CL    2030 P4=P(4)
CK    2040 P5=P(5)
DD    2050 C1=C(1)
DC    2060 C2=C(2)
DB    2070 C3=C(3)
DA    2080 C4=C(4)
CP    2090 C5=C(5)
DN    2100 IFP1=C1THENCIRCLE(65,R),2,1:PAINT(65,R),1:P1=Ø:C1=9
EE    2110 IFP2=C2THENCIRCLE(73,R),2,1:PAINT(73,R),1:P2=Ø:C2=9
EL    2120 IFP3=C3THENCIRCLE(81,R),2,1:PAINT(81,R),1:P3=Ø:C3=9
DM    2130 IFP4=C4THENCIRCLE(89,R),2,1:PAINT(89,R),1:P4=Ø:C4=9
ED    2140 IFP5=C5THENCIRCLE(97,R),2,1:PAINT(97,R),1:P5=Ø:C5=9
```

```
KA   2150 IFP(1)=C(1)ANDP(2)=C(2)ANDP(3)=C(3)ANDP(4)=C(4)ANDP(5)=C(5)THENF
          =1:GOTO241Ø
DL   2160 IFP1=C2THENCIRCLE(65,R),2,15:PAINT(65,R),15:P1=Ø:C2=9
DO   2170 IFP1=C3THENCIRCLE(65,R),2,15:PAINT(65,R),15:P1=Ø:C3=9
EB   2180 IFP1=C4THENCIRCLE(65,R),2,15:PAINT(65,R),15:P1=Ø:C4=9
EE   2190 IFP1=C5THENCIRCLE(65,R),2,15:PAINT(65,R),15:P1=Ø:C5=9
DA   2200 IFP2=C1THENCIRCLE(73,R),2,15:PAINT(73,R),15:P2=Ø:C1=9
DF   2210 IFP2=C3THENCIRCLE(73,R),2,15:PAINT(73,R),15:P2=Ø:C3=9
DI   2220 IFP2=C4THENCIRCLE(73,R),2,15:PAINT(73,R),15:P2=Ø:C4=9
DL   2230 IFP2=C5THENCIRCLE(73,R),2,15:PAINT(73,R),15:P2=Ø:C5=9
DC   2240 IFP3=C1THENCIRCLE(81,R),2,15:PAINT(81,R),15:P3=Ø:C1=9
DF   2250 IFP3=C2THENCIRCLE(81,R),2,15:PAINT(81,R),15:P3=Ø:C2=9
DK   2260 IFP3=C4THENCIRCLE(81,R),2,15:PAINT(81,R),15:P3=Ø:C4=9
DN   2270 IFP3=C5THENCIRCLE(81,R),2,15:PAINT(81,R),15:P3=Ø:C5=9
DE   2280 IFP4=C1THENCIRCLE(89,R),2,15:PAINT(89,R),15:P4=Ø:C1=9
DH   2290 IFP4=C2THENCIRCLE(89,R),2,15:PAINT(89,R),15:P4=Ø:C2=9
CP   2300 IFP4=C3THENCIRCLE(89,R),2,15:PAINT(89,R),15:P4=Ø:C3=9
DE   2310 IFP4=C5THENCIRCLE(89,R),2,15:PAINT(89,R),15:P4=Ø:C5=9
CL   2320 IFP5=C1THENCIRCLE(97,R),2,15:PAINT(97,R),15:P5=Ø:C1=9
CO   2330 IFP5=C2THENCIRCLE(97,R),2,15:PAINT(97,R),15:P5=Ø:C2=9
DB   2340 IFP5=C3THENCIRCLE(97,R),2,15:PAINT(97,R),15:P5=Ø:C3=9
DE   2350 IFP5=C4THENCIRCLE(97,R),2,15:PAINT(97,R),15:P5=Ø:C4=9
HG   2360 T=T+1
CE   2370 IFT>=7THENGOTO241Ø
EF   2380 S=S-2Ø
EE   2390 R=R-2Ø
AD   2400 GOTO167Ø
CC   2410 LINE(95,2Ø)-(2ØØ,3Ø),1,BF
AJ   2420 CIRCLE(115,25),5,C(1)
AL   2430 CIRCLE(135,25),5,C(2)
AN   2440 CIRCLE(155,25),5,C(3)
AP   2450 CIRCLE(175,25),5,C(4)
BB   2460 CIRCLE(195,25),5,C(5)
GB   2470 PAINT(115,25),C(1)
GB   2480 PAINT(135,25),C(2)
GB   2490 PAINT(155,25),C(3)
FG   2500 PAINT(175,25),C(4)
FG   2510 PAINT(195,25),C(5)
FF   2520 IFF<>1THENF=Ø
FB   2530 COLOR1:LOCATE68,17Ø:PRINT"██████████████████"
LI   2540 COLOR2:LOCATE73,167:PRINT"{ENTER} FOR NEW GAME"
IK   2550 COLOR2:LOCATE22Ø,135:PRINT"{ESC}"
FJ   2560 LOCATE218,15Ø:PRINT"TO END"
FE   2570 I$=INKEY$:IFI$=""THEN257Ø
BB   2580 IFI$=CHR$(13)THEN48Ø
DO   2590 IFI$=CHR$(27)THEN261Ø
AL   2600 GOTO26ØØ
FF   2610 COLOR15,4,5
END
```

BANK SWITCHING

by : T. Colverd

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV. 84.) or The Year Book.

```
ED        5 REM ├├├ THIS PROGRAM WILL NOT WORK UNDER DISK BASIC ┤┤┤
EF        6 REM ├├├     ONLY FOR SV-328 OR 318 WITH 64K RAM     ┤┤┤
BO      100 CLEAR 200,&HF480
AL      110 DEFUSR=&HF480:DEFINT I
BE      120 FOR I=&HF480 TO &HF4F9
DB      130 READ A$:POKEI,VAL("&H"+A$)
AI      140 NEXT I:I=USR(0):NEW
AL      150 DATA 22,5E,FE,E5,3E,C3,32,57,FF,21,B4,F4,22,58,FF,ED
AH      160 DATA 73,5C,FE,F3,3E,0F,D3,88,DB,90,32,64,FE,E6,FD,D3
DE      170 DATA 8C,21,00,80,11,00,00,01,00,80,ED,B0,3A,64,FE,D3
CK      180 DATA 8C,FB,E1,C9,FE,C9,C0,ED,73,5C,FE,DD,2A,03,FA,23
AI      190 DATA 22,5E,FE,F3,3E,0F,D3,88,DB,90,32,64,FE,E6,FD,D3
DF      200 DATA 8C,21,00,00,11,00,80,4E,EB,46,71,EB,70,23,13,7A
BM      210 DATA B3,20,F4,3A,64,FE,D3,8C,FB,ED,7B,5C,FE,CD,50,37
CC      220 DATA DD,22,03,FA,2A,5E,FE,7E,C1,C9
END
```

This program allows you to use the SVI SWITCH command with out the need of a second Bank of 64K ram.

Please note that the program only works with the disk drives turned off.

Anyone with a fix for this please call us.

LIBRARY NOTES

By. J. Collins

Welcome to the New Year to all members and especially to any new members. It's time for my regular Library News and Update on Library Software. For the new members some information on just what the library is and what it contains etc, and for everyone, some news of an exciting software package offer.

The staff of the Library is Jim Collins...he is assisted by Jim Collins...and in turn these two can call on Jim Collins ! I don't get paid for doing any of the work, and it must be fitted in with any other task such as being a husband and a father. I am willing to answer the phone at any reasonable hour after working hours but would prefer calls between 6.00pm and 8.00pm. You can call during daylight hours at the weekend but I don't have to be here. I will answer any technical or programming query that I am capable of and can soon pass you on to someone else if I don't have the answer you want. If you are writing for information I would appreciate a stamped addressed envelope for replies. If you are sending for disks or cassettes of software you should use a Padded Jiffy-Bag but don't put two hundred staples in the top. When ordering for the first time from the library please print your name and address very clearly. I will send back some official order forms with your order for future use and these should then be used exclusively for any orders. Enough now, or this wont fit all in one issue.....more later.

GROUP AUTHOR SOFTWARE

Items in this group are programs written by group members and sold through the Library on a Royalty basis. Any member may submit any of his own programs to the Program Evaluation Committee for possible inclusion in the For Sale category. Programs must be your own work, fully debugged, working correctly, and with full documentation. The documentation may be in the form of built-in instructions which can be accessed via the program, or they can be in the form of written instructions which we can duplicate and include with each copy sold. Authors are requested to advise on the actual price they wish to receive for each copy sold, bearing in mind that copying and handling charges must be added to this amount to cover the costs of the library. Once agreement has been reached there is no need for the author to do anything other than wait for Royalties to come in. As explained in earlier articles the Royalties are paid out to the authors at regular intervals as dictated by volume of sales. Anyone with suitable programs should make first contact with the Librarian at the address given. If you require any further info on this facet of the Library then ask the Librarian.

CP/M PUBLIC DOMAIN LIBRARY SOFTWARE

Items listed in this category are suitable for 318/328 owners who have an expander unit and one or two disk drives. Also suitable for MSX owners with disk drives and CP/M capability. (not all MSX machines support CP/M). DO NOT ASK FOR CP/M SOFTWARE ON CASSETTE.

The label "PUBLIC DOMAIN" refers to programs written and then released in order to be used by anyone, for no charge, by the author. The charges we make for these programs reflect the direct cost to the library in terms of labour, time, and resources. This represents the cheapest possible way of getting high-class utility software. All disk-drive owners should make full use of this section of our library. At the moment I can supply CP/M software on the following disk formats.

STANDARD	**SINGLE-SIDED DOUBLE DENSITY**	**(early 318/328)**
STANDARD	**DOUBLE-SIDED DOUBLE DENSITY**	**(later 318/328)**
CUSTOM	**DOUBLE-SIDED DOUBLE DENSITY**	**(altered BIOS)**
SVI-MSX	**DOUBLE-SIDED DOUBLE DENSITY**	**(5.25inch disk)**
XPRESS	**SINGLE-SIDED DOUBLE DENSITY**	**(3.5 inch disk)**

IMPORTANT NOTICE

When supplying disks for CP/M software you should FORMAT each disk and SYSGEN an image of your CP/M system onto each disk, and remember when ordering to clearly state if this has been done and in what format. If you don't do this you may have to wait longer for your order. I can not put a CP/M system on your disks as this would contravene copyright laws.

And now on with the rest of this article and away from CP/M but please read carefully the updated list of library items available for those who choose to use this facility.

NEW ITEM - NEW ITEM - NEW ITEM...SOFTWARE COMPETITION PROGRAMS

All those programs which were submitted for inclusion in our recent competition have been combined into three disks or (three cassettes) and are offered for sale as follows :-

Single Disks or Cassettes containing twelve or more programs will cost $18.00 per disk or $16.00 per cassette.

All three disks, containing over 36 programs, will cost $45.00. A saving of $9.00 on single disk price.

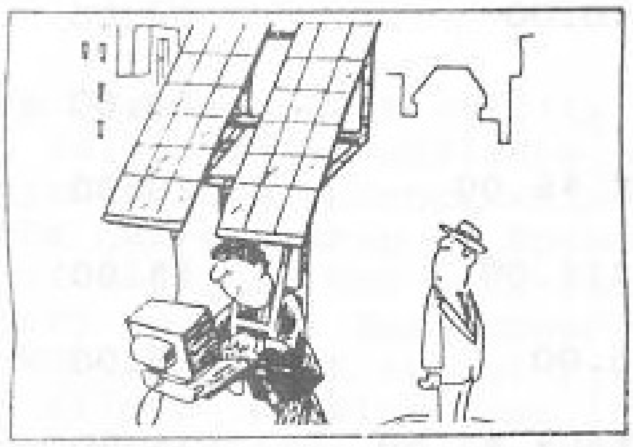
All three cassettes, containing over 36 programs, will cost $40.00. A saving of $8.00 on single cassette price.

IN THIS OFFER WE RESERVE THE RIGHT TO SUPPLY THE DISKS OR CASSETTES AND TO CHOOSE THE PROGRAMS WHICH WILL BE ON EACH DISK. Do not send your own disks or cassettes and do not request particular programs inside this Special Offer. Either buy a single disk or cassette or order the package. This is the only way the software will be available. When ordering please refer to the disk or package you want as follows :-

COMPETITION DISK NUMBER ONE/CASSETTE NUMBER ONE
COMPETITION DISK NUMBER TWO/CASSETTE NUMBER TWO
COMPETITION DISK NUMBER THREE/CASSETTE NUMBER THREE
COMPETITION DISK PACKAGE (3 disks)
COMPETITION CASSETTE PACKAGE (3 cassettes)

Here's your chance to build up your library of Games and utilities with programs going for less than a dollar each. When you consider that cost of media is also included in this bonus price it must rate as a real bargain. RUSH YOUR ORDERS NOW

MSX owners please note that this offer does not directly apply to you at the moment..................All the software submitted was for 318 or 328 computers and although most could be converted to MSX we don't have the time to do this just now. Perhaps later but don't take this to be a firm commitment. Of course there is nothing to stop you buying the software and tackling the conversions yourself. Most would require little change and the changes are relatively simple. If any MSXers decide to go this route then simply order what you want as shown above. Same prices and conditions of sale apply of course.



GROUP AUTHOR SOFTWARE-LIBRARY PROGRAM LIST

ASKING PRICE		OUR MEDIA		YOUR MEDIA	
Includes Pack and Post.		CASSETTE	DISK	CASSETTE	DISK
3D-MAZE $5.00		9.00	11.00	5.00	5.00
CALENDARS $3.00		7.00	9.00	3.00	3.00
MURDER $10.00	##	14.00	16.00	10.00	10.00
MYSTERIOUS MANOR $5.00		9.00	-----	5.00	-----
COUNT DRACULAR $10.00		14.00	-----	10.00	-----
CRUNCH $10.00	##	14.00	16.00	10.00	10.00
DISASSEMBLER $5.00	##	9.00	11.00	5.00	5.00
ELIZA $10.00	##	14.00	16.00	10.00	10.00
MARVYN $10.00	##	-----	16.00	-----	10.00
MIGHTY MORMAR $5.00		9.00	11.00	5.00	5.00
HOUSE OF FRANKENSTEIN $5.00		9.00	11.00	5.00	5.00
PACMAN $10.00		14.00	16.00	10.00	10.00
SUPER IMP/ED $10.00		14.00	16.00	10.00	10.00
JOYSTICK SPRITE $10.00	##	14.00	16.00	10.00	10.00
FILES $5.00		9.00	11.00	5.00	5.00
RUBIKS CUBE $10.00		14.00	16.00	10.00	10.00
X'BERT $10.00		14.00	16.00	10.00	10.00
FIVE GAME PACK $6.00		10.00	12.00	6.00	6.00
ASMED/LOADER $11.00		15.00	-----	11.00	-----
WP318/WP328 $5.00		9.00	11.00	5.00	5.00
MSX GAMES PACKAGE $7.50		11.50	13.50	7.50	7.50
BASIC UTILITIES $6.50		-----	12.50	-----	6.50
DRAW-2 $7.50		11.50	13.50	7.50	7.50
SVI ARTIST $7.50		11.50	13.50	7.50	7.50

COMPETITION DISK ONE	-----	18.00	-----	-----
COMPETITION DISK TWO	-----	18.00	-----	-----
COMPETITION DISK THREE	-----	18.00	-----	-----
THREE DISK PACKAGE DEAL	-----	45.00	-----	-----
COMPETITION CASSETTE ONE	16.00	-----	-----	-----
COMPETITION CASSETTE TWO	16.00	-----	-----	-----
COMPETITION CASSETTE THREE	16.00	-----	-----	-----
THREE CASSETTE PACKAGE DEAL	40.00	-----	-----	-----

The programs shown with two hash marks after the name are available for both SVI318/328 computers and also have been ported across to MSX. Please be sure when ordering that you specify which computer you want the software for.

CP/M PUBLIC DOMAIN SOFTWARE

ADVENT.COM *	$5.00	MODEM7.COM *	$5.00	YAM.COM *	$5.00
XDIR.COM	$5.00	ZCPR2	$5.00	CATALOG.COM	$5.00
CP/M GAMES DISK	$6.00	GRAF.COM *	$15.00	NULU.COM	$10.00

All CP/M software is suitable for DISK use only and the prices shown DO NOT INCLUDE PRICE OF DISK. If you don't supply disks please add normal $6.00 media charge to the price shown against the item you want. Those CP/M items with an asterisk after the program name fill a normal SS/DD disk and other items can not be included on the disk. Example......if you want MODEM7 and XDIR.COM you will need two disks.

There are two new CP/M utility disk offers in this listing. GRAF.COM is a full disk of software which will allow you to produce graphical displays and add enhancements to your printouts when combined with any of the current crop of Epson compatible printers. This is a powerful set of routines and well worth having in any CP/M library. There is a very powerful demo program on the disk to show you how it is done. NULU.COM is a set of routines which will allow you to Squeeze your CP/M files, combine them in libraries, and store them on disks in this compressed form. On the disk with NULU.COM is NSW.COM which is probably the most powerful disk utility available. These two programs are worth their weight in gold for any CP/M user. Full documentation files are on the disk for NULU and NSW and these files printout to give full operating instructions for both programs.

buy, trade & sell

FOR SALE

1 X SV601 EXPANDER
1 X SV801 DISK CONTROLLER
1 OR 2 80TK DRIVES 384K

PRICE

W/- 1 DRIVE $595
W/- 2 DRIVES $835

1 X 80 COL CARD $180
1 X CENTRONICS INTERFACE +
CABLE $95

CONTACT

G.C. JACOBI,
13 DALBY ST.,
NANANGO 4315

CALL 071-631776

HELP!

PLEASE HELP. IS THERE ANYONE OUT THERE WHO HAS THE GAMES, THE LOST PRINCESS AND PLANET OF ZANDELL.

CONTACT

ROGER LANDRIAT
3/64 TOWNSVILLE RD.
INGHAM,
NTH QLD 4850

FOR SALE

MODEM EXECUTIVE PROGRAM (MEX) PUBLIC DOMAIN FULLY CONFIGURED TO RUN ON THE SVI-738. IT CAN BE PASSWORDED IF YOU WANT IT. HELP FILE INCLUDED.

PRICE

$15 INCLUDES DISK

CONTACT

ALFRED BUATTI
16 MERRIGUM CRESC.
CLAYTON SOUTH 3169

FOR SALE

SV 328 ,CASSETTE, MINI EXPANDER, CENTRONICS INTERFACE, 2 JOYSTICKS, PRINTER CABLE, AND SOFTWARE.

PRICE

$400 WILL CONSIDER SPLITTING.

CONTACT

PETER BRUMBY,
212 OPOSSUM RD,
NORWOOD, 7250

CALL

(003) 444646

USER GROUP

ANYONE INTERESTED IN A USERS GROUP IN SYDNEY ??

CONTACT

888-7664 AFTER 7PM.