Next Meeting: Tuesday, April 3rd
7:30 P.M. (Details on Page 20)

VOICE OF THE '80 NEWSLETTER

April 1984
Volume 4
Number 11

APPLE's Macintosh Revealed!

By John Krause

Aha! The Macintosh makes its debut. Look out IBM! I haven't really seen one yet, but I just read an article about it. First some background about it. This machine will be a departure from many of the new machines popping up lately in that it is absolutely incompatible with the IBM PC or PC/jr. This is quite a gamble for APPLE. They are trying to establish an alternative to the PC, in essence saying, *"If we fail, IBM may take over."* Even Steve Jobs, Chairman of APPLE, and John Scully, President and CEO, admit that the future of APPLE rests on this machine. Scully's feelings are that *"This industry survives on constant innovation. Unless APPLE does it {innovate}, no one will be able to except IBM. But if we are successful, it means there is still a place for new ideas."*

Departures from IBM are many. First of all, gone are the 5 1/4 inch disks. The Macintosh uses the new 3 1/2 inch technology. {No word on the amount of storage space per disk yet}. Also, the newest APPLE uses a **32 BIT** Motorola 68000 microprocessor. That's right! 32 Bit!! Four times the actual capacity of the PC, which has a 16 Bit processor, but only an 8 Bit Data Path. The Macintosh also uses the Mouse. Just like the dying LISA. This lets users move around a cursor and point at graphic representations of the activity they wish to perform, like a trash can to kill a file. Cute huh? Using a graphics package called MacPaint, you can draw your own pictures.

Apple has had prototypes of the Macintosh in the hands of software houses for a year now, {so the article claims}, and they say 80 of them have agreed to write software for it. The real biggies among them are Microsoft and LOTUS Development. A version of LOTUS 1-2-3 is scheduled to be available for the new machine soon. The 500 programs expected to be ready by year end will also run on the Second Generation LISA, also due to be realeased on January 24th. The new LISA is said to list from $3,495 to $5,495 depending on memory configurations.

The Macintosh will list for $2,500. This gets you the keyboard, Mouse, high-resolution video monitor, {doesn't say if it's color or B/W}, 128K memory and one 3 1/2 inch Disk Drive. The Disk Drive is built into the video cabinet, and the keyboard is sepparate, but attached by a cable. No infra red job like the PEANUT-PC/jr. They claim that a similar IBM would be $1,000 more. APPLE hopes that people will buy them for the price, and because "*it's a cuddly machine*" (SIC). But because the home and hobbyist market is currently seen as almost saturated {do you believe that one?}, the analysts expect APPLE to go head to head with IBM in the small to medium business market. They are not given a lot of hope by the analysts. Personally, I hope they succeed.

Disk drives now available for the Model 100!

By George Saladino, Jr.

Tandy is coming out with disk drives for the Model 100! The first drive will be with a disk drive/TV interface and you can buy a second drive. The first drive with the interface is $799.00. The second drive $239.00.

By the time you read this, the drives should be out and I hope to see them and tell you more about them in the near future.

I can not stress enough that Model 100 owners should get on Micronet. The Model 100 board on there is very good and there are loads of good programs available to download from this board for free. One I have downloaded is "Locker.100", that allows me to shut off the Model 100 and no one can get on with out knowing my password. The reset button and the break key do not work! The only way one can get on is to do a cold restart and that would wipe out anything I did not want someone to see in memory.

This is one of the little "Gems" that one can find on the Model 100 board. I have put several programs on it my self. One late one is called "Set.100" that allows one to set the date and time of the Model 100 easily. When one sets the date, the day is automatically set by a formula I use in the program. Some slight modifications were made to my program and have been put on the board as "Timdat.bas".

Below is my "Set" program.

```
10 CLS
20 ON ERROR GOTO 440
30 PRINT @ 82,"Time is ";TIME$;" of ";DATE$;", "DAY$
40 PRINT:PRINT"  Do you want to make any changes?"
50 A$=INKEY$
60 PRINT @ 90,TIME$;
70 IF A$="" THEN 50
80 IF A$="y" OR A$="Y" THEN 90 ELSE END
90 CLS
100 PRINT @82,"Which would you like to change?"
110 PRINT:PRINT"<1> for time change"
120 PRINT"<2> for date change"
130 PRINT"(Day is changed with date.)"
140 INPUT A
150 IF A<>1 AND A<>2 THEN 90
160 IF A=1 THEN INPUT"Input new time HH:MM:SS";T$ELSE 200
170 TIME$=T$
180 CLS: PRINT @ 82,"Time is ";TIME$;" of ";DATE$;", "DAY$
190 GOTO 250
200 IF A=2 THEN INPUT"Input new date MM/DD/YY";D$
```

```
210 DATE$=D$
220 GOTO 300
230 DAY$=DA$
240 CLS: PRINT @ 82,"Time is ";TIME$;" of ";DATE$;", "DAY$
250 PRINT:PRINT" Any other changes?"
260 C$=INKEY$
270 PRINT @90,TIME$;
280 IF C$=""THEN260
290 IF C$="Y" OR C$="y" THEN90ELSE END
300 M=VAL(LEFT$(D$,2)) :D=VAL(MID$(D$,4,2))
    :Y=VAL(RIGHT$(D$,2)) :Y= Y+1900
310  IF M>2 THEN 340
320 M=M+12
330 Y=Y-1
340 N=D+2*M+INT(.6*(M+1))+Y+INT(Y/4)-INT(Y/100)+INT(Y/400)+2
350 N= INT((N/7-INT(N/7))*7+.5)
360 IF N=0 THEN DA$="SAT"
370 IF N=1 THEN DA$="SUN"
380 IF N=2 THEN DA$="MON"
390 IF N=3 THEN DA$="TUE"
400 IF N=4 THEN DA$="WED"
410 IF N=5 THEN DA$="THU"
420 IF N=6 THEN DA$="FRI"
430 GOTO 230
440 CLS:PRINT@ 80,"ENTRY error. Use correct time or date
     format!"
450 FOR T=1 TO 1000:NEXT
460 RESUME 10
```

Be careful when you type in lines 300 - 350. These are the formulas that calculate the correct day for the date.

If you want the shorter version, "Timdat", you will have to go on Micronet.

I have bought the Bar Code Reader, but have not been able to use it. The one I got was defective. I hope to have another in a week.

Six programs are given with the bar code reader. Three are machine language programs to read the Universal Product code, 3 of 9 code and Plessey bar code. The other three programs are in basic and are demo programs to show you how to use the bar code reader.

Well, till next time, have fun with your computer. I hope you all have a Model 100. I find it is very handy on my travels as a field man.

Master Directory Utilities

By John Krause

Well, as promised in my last article, here are the MERGE and special print routines for the Newsletter Master Directory program. This is also my first contribution to The Voice since my move to Rochester. All is well, we have finally settled in to our house, and are currently watching the third snowfall since our arrival cover the cars and driveway. I have naturally sought out the local Radio Shack stores to see what type of setup they have, all ten of them!!!!! I have only visited 3 so far, and as expected, only the local Computer Center has anything worth playing with. I have made nice with the manager as I did with Thom Seaman at the Norwalk store. I was very friendly with all of the staff down there, and I seem to have established that same rappor again here.

But anyway, down to business. The following programs give a little more flexibility to the Newsletter Master Directory program published in February. They allow you to merge the sepparate files together into two {or more} alphabetically sorted files and then print out this monster database. If you are running on a Model 4 with 128k memory, then you might be able to get several years worth of newsletter data in memory at once for printing and sorting. But with only 48k on a Model 1, I needed a way to organize all that data. So, MERGER/BAS and DIRPRT/BAS were born. MERGER/BAS will read in your data files and put them in two other files depending on their alphabetical sequence. {You can change the number of files, if you need, to very easily.} These sort files are then CLOSED and re-read into memory one at a time. This allows the NEWDOS/80 CMD "O" sort to resort the data into a continuous flow. {This will also work with Model III TRSDOS. MULTIDOS Users substitute CMD "Q"} The files are then re-written to disk. You now have two {or more} data files in alphanumeric order for printing.

This is where DIRPRT/BAS comes into play. This program will read in your data files and print them with appropriate page headings. The output will look just like that from DIRECTRY/BAS, but much longer. If you have visited the FREEBIE table lately, you will have seen the finished product. Both progrms will need to be tailored to your needs if you chose to use more than 2 Sort files.

MERGER/BAS

```
10 CLS :PRINT "Directory Master , Sort / Merge Utility." :PRINT
```

```
20 CLEAR 10000 :DEFINT A-Z :DIM  CT , X , FI$(9) :INPUT "How
Many Data Files To Read "; FC :FOR X = 0 TO FC - 1 :PRINT "Name
Of File "; X + 1 ; :INPUT FI$(X) :NEXT
30 CLS :OPEN "O" , 2 , "ATON/DAT :1" :OPEN "O" , 3 , "MTOZ/DAT
:1" : 'Change to suit needs.
40 FOR X = 0 TO FC - 1 :OPEN "I" , 1 , FI$ (X) :PRINT@ 260 ,
"Reading File "FI$ (X); :GOSUB 110 :CLOSE 1 :NEXT X :CLOSE
50 LPRINT "Total Count ";CT :LPRINT "A to N "AN :LPRINT "M to Z
"MZ :'Change LPRINT to PRINT if you don't want printed balancing
figures.
55 'ZERO Arrays, Re-Dimension to read file...
60 PRINT@ 260 , "Now Reading A to N File  "; :CLEAR 20000 :DIM
IT$ (550) , CT , X :OPEN "I" , 1 , "ATON/DAT" :CT = 0 :GOSUB 140
:GOSUB 100 :CLOSE :LPRINT CT "Records Read"
70 PRINT@ 260 , "Now Re-writing A to N File"; :OPEN "O" , 1 ,
"ATON/DAT" :FOR X = 0 TO CT - 1 :PRINT#1 , IT$ (X) :NEXT :CLOSE
:LPRINT CT" Records written back to A to N File." ' Change
LPRINT to PRINT if desired.
75 'ZERO Arrays, Re-Dimension to read next file...
80 PRINT@ 260 , "Now Reading M to Z File  "; :CLEAR 20000 :DIM
IT$ (550) , CT , X :OPEN "I" , 1 , "MTOZ/DAT" :CT = 0 :GOSUB 140
:GOSUB 100 :CLOSE :LPRINT CT "Records Read"
90 PRINT@ 260 , "Now Re-writing M to Z File"; :OPEN "O" , 1 ,
"MTOZ/DAT" :FOR X = 0 TO CT - 1 :PRINT#1 , IT$ (X) :NEXT :CLOSE
:LPRINT CT " Records written back to M to Z File." :END
95 'Repeat lines 80 & 90 if more files used, and change names of
files.
100 CMD "O" , CT , IT$(0) :RETURN :'Use CMD "Q" for MULTIDOS
110 PRINT@ 460 , CT; :PRINT@ 470 , AN; :PRINT@ 480 , MZ; :IF EOF
(1) THEN RETURN
120 LINEINPUT#1 , IN$ :CT = CT+1 :IF LEFT$ (IN$ , 1) < "M" THEN
PRINT#2 , IN$ :AN = AN + 1 :GOTO 110
130 PRINT#3 , IN$ :MZ = MZ + 1 :GOTO 110
140 IF EOF (1) THEN RETURN ELSE LINEINPUT#1 , IT$ (CT) :CT = CT
+ 1 :GOTO 140
```

To change MERGER/BAS to allow more Sort files, simply change line 30 to open more files, appropriately named of course. You will also need to allocate more file buffers upon invoking BASIC. Then change the subroutine at line 110 to look for your selected first letters. This routine presumes that all entries will start with a capital letter. A valid presumption I think, since all of our articles do begin with a capital letter. If this annoys you, simply change it. Some other logical file names might be ATOI/DAT, JTOR/DAT and STOZ/DAT. This would give you three Sort files. Again, you would only need to do this if you think your Sort files will be larger than available memory.

DIRPRT/BAS does expect you to have a printer, but you can use DIRECTRY/BAS without a printer. DIRECTRY/BAS allows output to Video as well as the printer. The reason for DIRPRT/BAS is only to print the output files of MERGER/BAS with the nice headings from DIRECTRY/BAS. If you don't have an MX-80, or if your printer codes are different, omit all occurances of LPRINT CHR$(7), and the Emphasized Mode commands in line 60.

DIRPRT/BAS

```
10 DEFUSR = -578 :X = USR(1000) :IF X <> 0 THEN PRINT "Not
Enough Space" :STOP :'For TRASHMAN.....
20 CLEAR 1 :CLS :PRINT CHR$ (23) :PRINT@ 460, "> Directory
Master <" :CLEAR 10000 :DEFINT A-Z :ON ERROR GOTO 190 :GOSUB 210
:FM = FRE("")
30 EX = 0 :CLS :PRINT@ 10, "> Newsletter Directory Print Routine
<" :PRINT@ 66,BX$; :PRINT@ 195, STRING$ (58, 140); :PRINT@ 165,
"Free Memory  : ";
40 PRINT@ 521, "<P>rint Directory."; :PRINT@ 800, "<Q>uit";
50 PRINT@ 179,FM; :GOSUB 230 :LPRINT CHR$ (7); :GOSUB 180
:PRINT@ 975, "Enter Selection"; :LC = 993 :GOSUB 170 :ON INSTR
("PpQq", IN$) + 1 GOTO 50, 60, 60, 150, 150
60 GOSUB 230 :LPRINT CHR$ (27); CHR$ (69) :GOSUB 180 :PRINT@
960, "  Paper At Top Of Page, Press <CLEAR >"; :LC = 998 :GOSUB
170 :GOSUB 140 :'LPRINTs put EPSON MX-80 in Emphasized Mode.
70 PRINT@ 517, "-->"; :FS$ = "ATON/DAT"
80 GOSUB 180 :PRINT@ 975, "     Loading ";FS$; :OPEN"I",1,FS$
90 IF EOF (1) THEN CLOSE :PRINT@ 261, "   "; :IF FS$ =
"ATON/DAT" THEN FS$ = "MTOZ/DAT" :GOTO 80 ELSE 30 'Change here
if more than 2 Sort files.
100 LINEINPUT#1 , DR$
110 LPRINT TAB (8) DR$ :IF PC = 55 THEN PC = 0 :GOSUB 130
120 PC = PC + 1 :IF INKEY$ = "@" THEN 30 ELSE 90 :'IF <@> key
pressed, cop out.
130 FOR LF = 1 TO 5 :LPRINT " " :NEXT LF
140 LPRINT " "  :LPRINT TAB (25)  "Newsletter Master Directory":
LPRINT " " :LPRINT TAB (8) "Subject-------------------------Vol.-
No.---Month-------Year--Page" :LPRINT TAB (8) STRING$ (64, 172)
:LPRINT " " :RETURN : 'Change Dashes to Spaces.
150 GOSUB 180 :PRINT@ 960, "Are You Sure You're Done (Y/N)"; :LC
= 991 :GOSUB 170 :IF IN$ <> "Y" AND IN$ <> "y" THEN 30 ELSE CLS
:END
160 GOSUB 180 :PRINT@ 965, "Directory File Not Found."; :PRINT@
517, "   "; :FOR Y = 0 TO 1000 :NEXT :GOTO 30
170 IN$ = INKEY$ :IF IN$ < CHR$ (31) THEN PRINT@ LC,CHR$ (RND
(63) + 128); :GOTO 170 ELSE PRINT@ LC,IN$; :RETURN :'INKEY$
Routine.  Prints random graphic to prompt for input.
180 PRINT@ 960, CHR$ (31); :RETURN :'Clears message line.
190 IF ERR/2 + 1 = 54 PRINT@ 970, "File Not Found"; :GOTO 50
200 CLS :PRINT@ 450, "Unexpected Error ";ERR/2 + 1;" In Line
";ERL :STOP
210 DIM DR$, IN$, LC, Z, FS$, PC :'FASTER Method...
220 BX$ = CHR$ (151) + STRING$ (58,131) + CHR$ (171) + CHR$
(196) :FOR X = 1 TO 11 :BX$ = BX$ + CHR$ (149) + CHR$ (250) +
CHR$ (170) + CHR$ (196) :NEXT :BX$ = BX$ + CHR$ (181) + STRING$
(58,176) + CHR$ (186) :RETURN
230 IF ( PEEK ( 14312 ) AND 240 ) <> 48 GOSUB 180 :PRINT@ 975,
"Ready Printer!"; :GOTO 230 ELSE RETURN
```

WHERE ARE THOSE /VC FORMULAS?

by Don Pearson, D D P assoc.70065,226 - 914-669-9622

Have you users of VISICALC ever wanted to ponder the structure of the formulas that you have so laborously entered? AND, where did they really /Replicate to? There was always that undocumented

/SS<ent> :P

that would print the formulae (thanks, Stan Tishler). But that produced streams of paper and from the bottom up (??). Unless you had a very weird template the formulas that related one to another were each on a different page and the positions didn't seem relate either. What really would be nice -- formulas in normal cell format!

Several weeks ago, I was having a chat with Tom Ettinger (a longstanding METRO (nyc) USERS GROUP member, as is Stan) and he proceded to demo an idea whose TIME had COME; looky:

```
. . .
. . .
>D7:/FG.25
>C7:@NPV(D7,E7...X7)
>B7:1800
>A7:"ROUTER
>H6:930
. . .
. . .
/GFI
/GC9
/X>A1:>A1:
```

To produce this display save your VC template, with formulas (and input data, if you like). Bring up SCRIPSIT or other well built word processor. LOAD your file (the one you just saved from VC), Notice the format:

```
>C5:@NPV(D7,E7...H7)
>A7:"ROUTER
```

The first part of each entry must have been supplied by the VISICALC program - we didn't type in the '>' or the 'C' or the '5' or the ':' did we? Also note the quote in the A7 label type entry. THERE! that's the idea...replace all the colons <:> with colon/quotes <:">..makes VC think all cells are labels. It's easy - we're in SCRIPSIT, so do a repeat/replace (make sure you're at the top of the listing ...):

...enter a: <cntl R>
...msg asks: REPEAT HOW MANY TIMES?,.. <ent> (255*)
...next msg asks: ENTER REPEAT COMMAND
...you enter: <brk>.. R>:>:"

...(*)'enter' gives max number of repeats (255).Repeat the process as necessary to get to the bottom of listing. OOPS, note that the last few lines aren't really cells (no letter/number giz before the colons)-they're global format instructions for the whole template so take out those few quotes after the colons that we just put in these lines. Now save the whole thing as a new file. Go back into VC and load this new file. The display now looks like this, along with normal R/C headers:

A1 C 22

	A	B	C
1			
2			
3	/FR"EQUIP	/FR"COST	/FR"NPV
4	"DR PRESS	1200	@NPV(D4,E4...H4)
5	"LATHE	2700	@NPV(D5,E5...H5)
6	"MILL	2500	@NPV(D6,E6...H6)
7	"ROUTER	1800	@NPV(D7,E7...H7)
8			
9			
10			
11			
12			

Change the column width (global) as necessary, to view the whole formulas. Printout also in the usual manner (probably fewer cols/pg than before).

N I C E ? !!

John Bethel (FCUG) suggested a nice refinement : after the above repeat/replace - do another. Replace the <""> with a <"> ..so the quote mark doesn't print with each label.

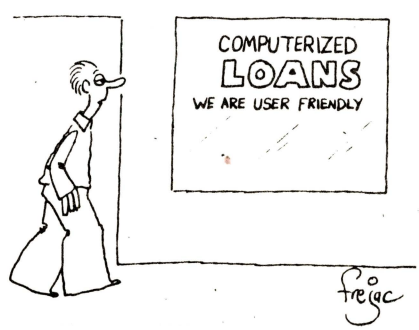
"Maintenance? My modem's started making obscene phone calls!"

"PRINTER LOWER CASE MOD"

By Dave Pederson

After purchasing a used printer (RS LP1), I had finally completed my basic full system: MODEL 1, EXP INT, DISK DRIVE, & PRINTER. At first it was great to be able to see my programs and scripsit files on hard-copy. When I proudly displayed the results to my family & friends they all thought of the wonderful use the printer would recieve. It wasn't until my daughter turned in a class report printed in all upper-case letters did I realize that in these days, reports were almost unacceptable in all upper-case. I explained to her that it was the way that the printer was designed and we had to live with it. She accepted the fact that the rest of her world would be expressed in upper-case. Shortly thereafter, her offer of preparing a school newspaper was turned down because it could not be printed in lower-case. Feeling dismayed, she questioned me for a solution. After thumbing through all of the back issues of 80-US and MICRO-80, I stumbled upon an article by Larry Panattoni, projects editor of 80-US featuring a construction project to convert the LP1 to lower-case. The project although cumbersome and requiring five ic's didn't phase me much in the construction end but bothered me in the fact that the heart of project was a ROM character generater available only from a british manufacturer at a cost of $62.00.

While discussing the project with John Roy, I took his advice to design a character generator to replace the original. Under John's supervision, I researched IC manuals until one matched the parameters needed. A 2732 EPROM 4K X 8 would work but not directly replaceable as the original ROM was 28 pins and the 2732 was outfitted with 24 pins with a completly different pinout. This problem was rectified by designing a transition harness approximatly 6 inches long with a 28 pin plug on one end and a 24 pin IC socket on the other. Once completed, the original ROM was removed and the transition harness inserted in it's place. The majority of time was spent in determining the correct address and data to be programed into the EPROM to obtain the required ASCII output format. Now that the printer is pumping away with lower and upper case, my daughter asked me the possibility of GRAPHICS. My reply was, " *Lets go out and look at the EPSON RX-80*"!



ROM CHECK (ROMCHK) REVISITED.....BY J. ROY

ROMCHK is a very simple program that will obtain a check sum of a specified range of ROM locations. Actually all the program does is to add each succeeding ROM location to obtain a cumulative total. If this number agrees with a previously recorded number it is reasonable to assume the ROM's are good.

This method is not fool proof and some short comings must be noted. It is possible that two or more ROM locations could be bad in such a manner that the summation of these bad locations could result in the same total number. Another possible problem is a failure in ROM that will prevent this BASIC program from running.

In its simplest form the ROMCHK program listed can be executed to obtain a cumulative check sum of all of ROM. Because of individual requirements, the start address, end address, and increment is selectable. The ability to establish start and end points in addition to increment information will allow pinpointing ROM differences to specific locations. FIGURE 1 shows an increment every 1024 locations. This number was picked as a good compromise for evaluation and explanation purposes. A bad increment can be linked to the specific failing ROM. This feature can be very useful to obtain other intermediate check sums for comparison with some previously known information or with the results of a friends computer. The resolution of the increment can be adjusted to a point of the actual failing address.

The end address will change depending on your computer model. The model 1 has 12K of ROM (3-4K ROMS) as compared to 14K (1-8K, 1-4k, & 1-2K ROMS) for the Model 3 and 4. Therefore the end address on a Model 1 will be 12288, and Model 3 and 4 will be 14336.

Figure 1 is a collection of known versions of ROM's, with "A" being the oldest for the Model 1. Other versions may exist but these are the only ones I have information on.

RX 80 REFERENCE SHEET

My reference sheet for the current version of the Epson RX 80 printer appears on pages 12 and 13.

$$$

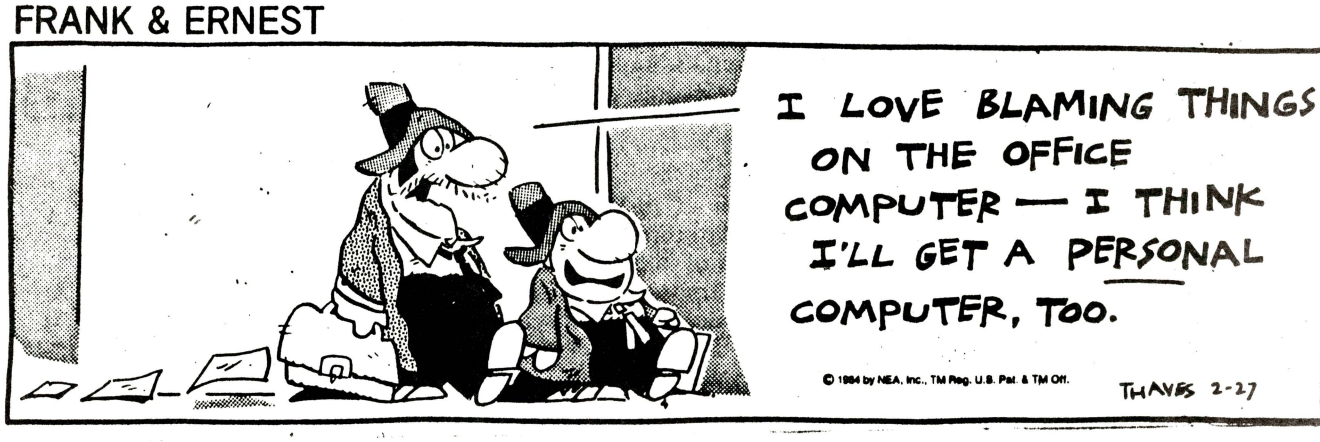
<FIGURE 1>

ROMCHK TABLE FOR KNOWN VERSIONS OF ROMS

ADDRESS	MOD 1 VER A	MOD 1 VER B	MOD 1 VER C	MOD 3 VER A	MOD 3 VER B	MOD 4 VER A
0000- 1024	108780		109316	105410	105410	105410
1025- 2048	107648			114446	114446	114446
2049- 3072	109188					
3073- 4096	112240					
4097- 5120	110971					
5121- 6144	111743					
6145- 7168	107316			107364	107364	107364
7169- 8192	119063					
8193- 9216	117901	117227	117151	117151	117151	117151
9217-10240	122133					
10241-11264	121776			121693	121693	121693
11265-12288	113550	114186	114082	114283	114283	114283
12289-13312	------	------	------	99989	99957	97808
13313-14336	------	------	------	108909	108801	104831
0000-12288	1362309	1362271	1362627	1365685	1365685	1365685
0000-14336	-------	-------	-------	1574583	1574443	1568324

```
10 REM ROM CONTENT CHECK ROUTINE (ROMCHK) 2-26-84 J. ROY
15 REM STANDARD VALUES FOR: START, END, INCREMENT
16 REM          MODEL 1= 0,12288,1024
17 REM          MODEL 3= 0,14436,1024
18 REM          MODEL 4= 0,14436,1024
30 DEFDBL Z: CLS: INPUT"START,END,INCREMENT";S,E,I
40 PRINT"START=";S;"END=";E;"INCREMENT=";I
50 FOR N=S TO E: Z1=PEEK(N): C=C+1
80 Z2=Z1+Z2:Z3=Z1+Z3
90 IF C<>I THEN 140
100 S=S+I
110 PRINT"ADD=";S;TAB(15)"INC CKSUM=";Z2;TAB(36)"CUM CKSUM=";Z3
120 C=0: Z2=0: Z1=0
140 NEXT N
```

FRANK & ERNEST



RX 80 PRINTER REFERENCE SHEET 2-29-84 By: J. Roy

SPECIAL TEST MODES

Self test.....turn power on with LF switch depressed
Hex dump.....turn power on with LF & FF sw depressed

INITIALIZE PRINTER CONDITIONS.....(27)(64)

Conditions are as defined by dip-switches at power-up

PRINT STYLE CODES

Select Standard ASCII Characters.....(27)(53)
" Italic " " .....(27)(52)
" International " " .....(27)(82)(n)
n= 0-USA 1-France 2-Germany 3-England 4-Denmark
5-Sweden 6-Italy 7-Spain 8-Japan 9-Norway

SPECIAL PRINT MODES

Double-strike printing.....(27)(71)
Cancel double-strike printing.....(27)(72)
Emphasized printing.....(27)(69)
Cancel emphasized printing.....(27)(70)
Print with underline.....(27)(45)(1)
Cancel underline.....(27)(45)(0)
Print superscript.....(27)(83)(0)
Print subscript.....(27)(83)(1)
Cancel super- or subscript.....(27)(84)
Print unidirectionally.....(27)(60) or (27)(85)(1)
Print bidirectionally.....(27)(85)(0)
Half speed.....(27)(115)(1)
Cancel half speed.....(27)(115)(0)

FONT PITCH CODES

Pica (Normal Mode) (10 char/inch).....(18) or (27)(80)
Elite (12 char/inch).....(27)(77)
Condensed (17 char/inch).....(15) or (27)(15)
Enlarged Mode (one-half of the above above pitches)
Not cancelled at end of line.....(27)(87)(1)
Cancelled at end of line (by CR).....(14) or (27)(14)
Return to unelarged mode (10, 12, or 17 CPI)....(27)(87)(0)
Return to Normal Mode (10 CPI, not enlarged).....(20)

FORM FEEDING

Advance the form to the next top-of-form.....(12)
Set the form length to n lines 0<n<127.....(27)(67)(n)
Set the form length to n inches 0<n<22.....(27)(67)(0)(n)
Skip-over-perforation (advance to next top-of-form
when n lines are left) 0<n<127.....(27)(78)(n)
Cancel skip-over-perforation.....(27)(79)

HORIZONTAL & VERTICAL POSITIONING

```
Carrage return.....................................(13)
Set the left hand print margin...Ø<n<78........(27)(1Ø8)(n)
Set the right hand print margin...2<n<8Ø........(27)(81)(n)
Move the head to next H-tab position...................(9)
Set H/V skip position setting...Ø<m<127.....(27)(1Ø2)(n)(m)
Set H/V tab unit positions..................(27)(1Ø1)(n)(m)
  n= Ø-Horz  1-Vert    m= multiplier (default = 8)
Move head back 1 position..............................(8)
Advance form to next vertical tab.....................(11)
```

LINEFEEDING

```
Execute a single linefeed..............................(1Ø)
Set linefeed to 1/8 inch.........................(27)(48)
Set linefeed to 7/72 inch........................(27)(49)
Set linefeed to 1/6 inch.........................(27)(5Ø)
Set linefeed to n/72 inch  Ø<n<85...............(27)(65)(n)
Set linefeed to n/216 inch   Ø<n<255............(27)(51)(n)
Feeds n/216 inch line spacing (1 line) Ø<n<255..(27)(74)(n)
```

GRAPHICS

In the following commands:

n1 & n2 = number of bit image data to be sent
n1= remainder of (number of data) over 256
n2= quotient of (number of data over 256
b1,b2...bx are the decimal m-values, which, when translated into their binary equivalents, show which dots will be printed. In an 8-bit system the dots are valued, from top to bottom, at 128,64,32,16,8,4,2,1

Within this command, the ninth pin cannot be fired

```
Select special char generator..................(27)(1Ø9)(n)
  n= 4-graphic characters   Ø-control codes
Bit image mode selection (27)(42)(m)(n1)(n2)(d1)(d2)...(dx)
  m= densities below
Ø Normal density............(27)(75)(n1)(n2)(d1)(d2)...(dx)
1 Dual density..............(27)(76)(n1)(n2)(d1)(d2)...(dx)
2 Dual density / 2X speed...(27)(89)(n1)(n2)(d1)(d2)...(dx)
3 Quad density..............(27)(9Ø)(n1)(n2)(d1)(d2)...(dx)
```

OTHER FUNCTION CODES

```
Delete last character sent (except function codes)....(127)
Activate the printer's buzzer for Ø.2 sec...............(7)
Disable printer's "paper-out" detector.............(27)(56)
Enable printer's "paper-out" detector..............(27)(57)
```

Ask Alan
by
Alan Abrahamson

Q: From the prez, "will you do a demo this month?"
A: *Yes, what subject?*
Prez: Assembly language.
A: *Why don't you get Dick Carricato or someone else more expert than I?*
Prez: I want to keep it SIMPLE!
A: *Well SIMPLE is what you will get since that is all I am up to. After over 2 years of trying to learn this stuff, this demo will probably disclose my true understanding of assembly language programming.*

Now for all of you who will unfortunately attend this meeting, I am suggesting to you that you bring along your newsletter to the meeting. No, I'm not signing autographs. I think that you might be able to better follow the demo if you have the program sheets in front of you and no handouts will be prepared. The monitors are not always "seeable" from where you are, especially the fine print. Remember this is "INTRODUCTION TO ASSEMBLY LANGUAGE" so don't look for anything fancy. Perhaps I can lay a few "CONCEPTS" on you and then you will be well on your way to creating SUPER UTILITY 99.9999.

WHITEOUT/BAS

In BASIC we might code this task like this:

```
 5 CLS
10 FOR X = 0 TO 127 'Loop 128 counts
20    FOR Y = 0 TO 47  'Inner loop 48 counts
30       SET(X,Y) 'Set point on CRT at X,Y
40  NEXT Y,X 'Repeat Loops till done
```

WHITEOUT/SRC

In Assembler we might code this task this way:

```
100              ORG     0F00H    ;Memory location of Object code
120              LD      DE,3C00H;Load DE with screen location
130              LD      BC,400H ;Number of bytes to move
140  LOOP        LD      A,BFH   ;Load source byte graphic block
150              LD      (DE),A  ;Load DE's location with block"
170              INC     DE      ;Point to next destination
180              DEC     BC      ;Decrement counter
190              LD      A,B     ;Get MSB count
200              OR      C       ;Merge LSB count
210              JR      NZ,LOOP ;Go if count NOT 0
220              RET             ;Return to BASIC
230              END
```

WHITEOUT/BPF

```
10 DEFUSR = &HF000 ' Define address of machine language code
20 FOR X = &HF000 TO &HF000 + 15 'Start loop of 16 counts
30 READ A 'Read data 1 at a time
40 POKE X,A 'Poke data in location X
50 NEXT X 'Repeat till loop finished
60 X = USR(0) 'Make USR call to F000 hex
70 DATA 17,0,60,1,0,4,62,191,18,19,11,120,177,32,247,201
   REMARK: Hex equivalents of above decimal data bytes are:
           17d = 11H = LD DE,nn
           nn  = 60 * 256 + 0 = 15360decimal or 3C00hex
            1d = 1H  = LD BC,nn
           nn  = 4 * 256 + 0 = 1024decimal or 400hex
           62d = 3EH = LD A,n
            n  = 191decimal or BFH
           18d = 12H = LD (DE),A
           19d = 13H = INC DE
           11d = 0BH = DEC BC
          120d = 78H = LD A,B
          177d = B1H = OR C
           32d = 20H = JR NZ,n
            n  = 247d = F7H (relative address back)
          201d = C9H  = RET
```

Z-80 Registers

Inside the Z-80 CPU there are some work locations called registers. These registers are what the assembly language programmer has to work with. They are called:

AF (Accumulator & Flag)
BC (16 bit register pair)
DE (16 bit register pair)
HL (16 bit register pair)
PC (Program Counter)
SP (Stack Pointer)
IX (Index Register)
IY (Index Register)
R (Refresh Register)
I (Interrupt)

and the "primed" registers: A', F', B', C', D', E', H' & L'

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$     SAVE BIG BUCKS      $
$     Send 1985 dues      $
$ prior to June to:       $
$ John Roy, Treas.        $
$ 18  Winfield Drive      $
$ Shelton, CT   06484     $
$ $20 vs $24 in June      $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Z-80 INSTRUCTIONS

Z80 OPCODE	DEC	HEX	OCT	ASCII	COMMENT
NOP	000	00	000	NUL	
LD BC,nn	001	01	001	SOH	
LD (BC),A	002	02	002	STX	
INC BC	003	03	003	ETX	
INC B	004	04	004	EOT	
DEC B	005	05	005	ENQ	
LD B,n	006	06	006	ACK	
RLCA	007	07	007	BEL	
EX AF,AF'	008	08	010	BS	b/s,erase
ADD HL,BC	009	09	011	HT	
LD A,(BC)	010	0A	012	LF	lf,car. ret
DEC BC	011	0B	013	VT	lf,no C/R
INC C	012	0C	014	FF	top of form
DEC C	013	0D	015	CR	lf,car. ret
LD C,n	014	0E	016	SO	cursor on
RRCA	015	0F	017	SI	cursor off
DJ NZ,n	016	10	020	DLE	
LD DE,nn	017	11	021	DC1	
LD (DE),A	018	12	022	DC2	
INC DE	019	13	023	DC3	
INC D	020	14	024	DC4	
DEC D	021	15	025	NAK	
LD D,n	022	16	026	SYN	
RLA	023	17	027	ETB	32 char
JR n	024	18	030	CAN	b/s cursor
ADD HL,DE	025	19	031	EM	adv cursor
LD A,(DE)	026	1A	032	SUB	dn. cursor
DEC DE	027	1B	033	ESC	up cursor
INC C	028	1C	034	FS	home cursor
DEC DE	029	1D	035	GS	cursor
LD E,n	030	1E	036	RS	erase EOL
RRA	031	1F	037	US	erase EOS
JR NZ,n	032	20	040	SP	space-blank
LD HL,nn	033	21	041	!	
LD (nn),HL	034	22	042	"	
INC HL	035	23	043	#	
INC H	036	24	044	$	
DEC H	037	25	045	%	
LD H,n	038	26	046	&	
DAA	039	27	047	'	apostrophe
JR Z,n	040	28	050	(	
ADD HL,HL	041	29	051	)	
LD HL,(nn)	042	2A	052	*	
DEC HL	043	2B	053	+	
INC L	044	2C	054	,	comma
DEC L	045	2D	055	-	dash
LD L,n	046	2E	056	.	period
CPL	047	2F	057	/	slash
JR NC,n	048	30	060	0	zero
LD SP,nn	049	31	061	1	
LD nn,A	050	32	062	2	
INC SP	051	33	063	3	
INC (HL)	052	34	064	4	
DEC (HL)	053	35	065	5	
LD (HL),n	054	36	066	6	
SCF	055	37	067	7	
JR C,n	056	38	070	8	
ADD HL,SP	057	39	071	9	
ADD A,nn	058	3A	072	:	
DEC SP	059	3B	073	;	
INC A	060	3C	074	<	
DEC A	061	3D	075	=	
LD A,n	062	3E	076	>	
CCF	063	3F	077	?	

Z80 OPCODE	DEC	HEX	OCT	ASCII	COMMENT
LD B,B	064	40	100	@	
LD B,C	065	41	101	A	
LD B,D	066	42	102	B	
LD B,E	067	43	103	C	
LD B,H	068	44	104	D	
LD B,L	069	45	105	E	
LD B,(HL)	070	46	106	F	
LD B,A	071	47	107	G	
LD C,B	072	48	110	H	
LD C,C	073	49	111	I	
LD C,D	074	4A	112	J	
LD C,E	075	4B	113	K	
LD C,H	076	4C	114	L	
LD C,L	077	4D	115	M	
LD C,(HL)	078	4E	116	N	
LD C,A	079	4F	117	O	
LD D,B	080	50	120	P	
LD D,C	081	51	121	Q	
LD D,D	082	52	122	R	
LD D,E	083	53	123	S	
LD D,H	084	54	124	T	
LD D,L	085	55	125	U	
LD D,(HL)	086	56	126	V	
LD D,A	087	57	127	W	
LD E,B	088	58	130	X	
LD E,C	089	59	131	Y	
LD E,D	090	5A	132	Z	
LD E,E	091	5B	133	up arrow	
LD E,H	092	5C	134	down arrow	
LD E,L	093	5D	135	left arrow	
LD E,(HL)	094	5E	136	right arrow	
LD E,A	095	5F	137	_	cursor
LD H,B	096	60	140	shift @	
LD H,C	097	61	141	a	
LD H,D	098	62	142	b	
LD H,E	099	63	143	c	
LD H,H	100	64	144	d	
LD H,L	101	65	145	e	
LD H,(HL)	102	66	146	f	
LD H,A	103	67	147	g	
LD L,B	104	68	150	h	
LD L,C	105	69	151	i	
LD L,D	106	6A	152	j	
LD L,E	107	6B	153	k	
LD L,H	108	6C	154	l	
LD L,L	109	6D	155	m	
LD L,(HL)	110	6E	156	n	
LD L,A	111	6F	157	o	
LD (HL),B	112	70	160	p	
LD (HL),C	113	71	161	q	
LD (HL),D	114	72	162	r	
LD (HL),E	115	73	163	s	
LD (HL),H	116	74	164	t	
LD (HL),L	117	75	165	u	
HALT	118	76	166	v	
LD (HL),A	119	77	167	w	
LD A,B	120	78	170	x	
LD A,C	121	79	171	y	
LD A,D	122	7A	172	z	
LD A,E	123	7B	173	{	
LD A,H	124	7C	174	|	
LD A,L	125	7D	175	}	
LD A,(HL)	126	7E	176	~	
LD A,A	127	7F	177	DEL	delete

Z-80 INSTRUCTIONS

Z80 OPCODE	DEC	HEX	OCT	BASIC TOKEN
ADD A,B	128	80	200	END
ADD A,C	129	81	201	FOR
ADD A,D	130	82	202	RESET
ADD A,E	131	83	203	SET
ADD A,H	132	84	204	CLS
ADD A,L	133	85	205	CMD
ADD A,(HL)	134	86	206	RANDOM
ADD A,A	135	87	207	NEXT
ADC A,B	136	88	210	DATA
ADC A,C	137	89	211	INPUT
ADC A,D	138	8A	212	DIM
ADC A,E	139	8B	213	READ
ADC A,H	140	8C	214	LET
ADC A,L	141	8D	215	GOTO
ADC A,(HL)	142	8E	216	RUN
ADC A,A	143	8F	217	IF
SUB B	144	90	220	RESTORE
SUB C	145	91	221	GOSUB
SUB D	146	92	222	RETURN
SUB E	147	93	223	REM
SUB H	148	94	224	STOP
SUB L	149	95	225	ELSE
SUB (HL)	150	96	226	TRON
SUB A	151	97	227	TROFF
SBC A,B	152	98	230	DEFSTR
SBC A,C	153	99	231	DEFINT
SBC A,D	154	9A	232	DEFSNG
SBC A,E	155	9B	233	DEFDBL
SBC A,H	156	9C	234	LINE
SBC A,L	157	9D	235	EDIT
SBC A,(HL)	158	9E	236	ERROR
SBC A,A	159	9F	237	RESUME
AND B	160	A0	240	OUT
AND C	161	A1	241	ON
AND D	162	A2	242	OPEN
AND E	163	A3	243	FIELD
AND H	164	A4	244	GET
AND L	165	A5	245	PUT
AND (HL)	166	A6	246	CLOSE
AND A	167	A7	247	LOAD
XOR B	168	A8	250	MERGE
XOR C	169	A9	251	NAME
XOR D	170	AA	252	KILL
XOR E	171	AB	253	LSET
XOR H	172	AC	254	RSET
XOR L	173	AD	255	SAVE
XOR (HL)	174	AE	256	SYSTEM
XOR A	175	AF	257	LPRINT
OR B	176	B0	260	DEF
OR C	177	B1	261	POKE
OR D	178	B2	262	PRINT
OR E	179	B3	263	CONT
OR H	180	B4	264	LIST
OR L	181	B5	265	LLIST
OR (HL)	182	B6	266	DELETE
OR A	183	B7	267	AUTO
CP B	184	B8	270	CLEAR
CP C	185	B9	271	CLOAD
CP D	186	BA	272	CSAVE
CP E	187	BB	273	NEW
CP H	188	BC	274	TAB(
CP L	189	BD	275	TO
CP (HL)	190	BE	276	FN
CP A	191	BF	277	USING

Z80 OPCODE	DEC	HEX	OCT	BASIC TOKEN
RET NZ	192	C0	300	VARPTR
POP BC	193	C1	301	USR
JP NZ	194	C2	302	ERL
JP nn	195	C3	303	ERR
CALL NZ,nn	196	C4	304	STRING$
PUSH BC	197	C5	305	INSTR
ADD A,N	198	C6	306	POINT
RST 00H	199	C7	307	TIME$
RET Z	200	C8	310	MEM
RET	201	C9	311	INKEY$
JP Z,nn	202	CA	312	THEN (,)
2-byte ins	203	CB	313	NOT
CALL Z,nn	204	CC	314	STEP
CALL nn	205	CD	315	+
ADC A,n	206	CE	316	-
RST 08H	207	CF	317	*
RET NC	208	D0	320	/
POP DE	209	D1	321	] (powers)
JP NC,nn	210	D2	322	AND
OUT (n),A	211	D3	323	OR
CALL NC,nn	212	D4	324	>
PUSH DE	213	D5	325	=
SUB n	214	D6	326	<
RST 10H	215	D7	327	SGN
RET C	216	D8	330	INT
EXX	217	D9	331	ABS
JP C,nn	218	DA	332	FRE
IN A,(n)	219	DB	333	INP
CALL Z,nn	220	DC	334	POS
2-byte ins	221	DD	335	SQR
SBC A,n	222	DE	336	RND
RST 18H	223	DF	337	LOG
RST PO	224	E0	340	EXP
POP HL	225	E1	341	COS
JP PO,nn	226	E2	342	SIN
EX (SP),HL	227	E3	343	TAN
CALL PO,nn	228	E4	344	ATN
PUSH HL	229	E5	345	PEEK
AND n	230	E6	346	CVI
RST 20H	231	E7	347	CVS
RET PE	232	E8	350	CVD
JP (HL)	233	E9	351	EOF
JP PE,nn	234	EA	352	LOC
EX DE,HL	235	EB	353	LOF
CALL PE,nn	236	EC	354	MKI$
2-byte ins	237	ED	355	MKS$
XOR	238	EE	356	MKD$
RST 28H	239	EF	357	CINT
RET P	240	F0	360	CSNG
POP AF	241	F1	361	CDBL
JP P,nn	242	F2	362	FIX
DI	243	F3	363	LEN
CALL P,nn	244	F4	364	STR$
PUSH AF	245	F5	365	VAL
OR n	246	F6	366	ASC
RST 30H	247	F7	367	CHR$
RET M	248	F8	370	LEFT$
LD SP,HL	249	F9	371	RIGHT$
JP M,nn	250	FA	372	MID$
EI	251	FB	373	' (REM)
CALL M,nn	252	FC	374	
2-byte ins	253	FD	375	. (dec. pt.)
CP n	254	FE	376	!
RST 38H	255	FF	377	ISA

RETRIEVING KILLED NEWDOS/80 FILES

by Tim Porreca
Reprinted from TBUG

This technique has been upgraded by Jack Decker and Dave McGlumphy to add additional information and make it more universally applicable to the many different disk formats that NEWDOS/80 is capable of producing. The revised technique appeared in Northern Bytes, Volume 5, Number 2.

One good thing about computers is that everything is so well documented. The problem is, can you understand the documentation? I have always been plagued with questions from computer owners who have the information they need but can't understand it. A good example of this is the NEWDOS/80 manual. The wording in this manual can be very difficult and technical for a beginner. I found one book, TRS-80 ASSEMBLY-LANGUAGE PROGRAMMING, to be geared in the direction that the person reading it was already a master assembly-language programmer.

In this article I will try to explain an easy way that one can recover a killed file using Superzap on NEWDOS/80 version 2.0. When one first enters Superzap one must first use the DNTH command (Display Name/Type Hashcode). You willthen enter the filename and the type of fiel (i.e. for LMOFFSET/CMD, LMOFFSET is the filename and CMD is the file type). Make note of the hex hashcode that is displayed. Now type DFS (Display File's Sector). Answer the FILESPEC? question with DIR/SYS:n (replace n with the drive number that contains the disk with the killed file), then answer the RELATIVE-SECTOR-WITHIN-FILE #? question with "1". This screen is the HIT sector for the directory. It can be recognized by the FRS 1 on the bottom left. Now press the ; sign on the keyboard once and the drive will start and the screen will change, this should be remembered as number 1. If you do not see the filename of the file you want to recover along the right (ASCII) side of the screen, press the ; sign again and count 1 more for each screen until the program you want to retrieve is seen on the right side of the screen. When you reach this screen, write down your current count (the number of times you had to press the ; sign to get to this screen). Now, type the letters MOD00. A large cursor should appear at the upper left of the screen. Use the arrow keys to move the cursor to the beginning of the line that your file is on. For any /CMD, /BAS, /TXT or other "user" file type the numbers 1 and 0 followed by <ENTER>. For a system (SYSnn/SYS) file type 5F and for BASIC/CMD type 1E. After <ENTER> is pressed, press "Y" to modify the disk sector. Now make note of the horizontal line on which this correction was made (i.e. E0, 20, 60 etc. - this is shown in the vertical column of two-digit hex numbers near the left-hand side of the screen). Return now to the HIT sector (type "K" and answer the RELATIVE-SECTOR- WITHIN-FILE #? question with "1").

Type MODØØ and move the cursor to the leftmost side of the screen and then to the line on which the correction was just made in the sector that contained the filename (check the two-digit hex number near the left-hand side of the screen to make sure). Now recall the count of the number of times you had to press the ; sign to get to the screen containing your filename, subtract one, multiply by two, and press the right arrow key that many times. After this is completed type in the hashcode that was previously recorded. Now hit <ENTER> and respond "Y". Type EXIT and your program should be there.

Your program may appear to be restored at this point, but there is still one problem - the GAT sector hasn't been fixed and writing another file to that diskette may overlay the data sectors. You can prove that by using a blank diskette, creating a file, KILLing it, fixing it by using the above method, writing another file to the same diskette and running DIRCHECK. The simplest solution is to copy the "unKILLed" file to another diskette, KILL the "unKILLed" file, and copy the copied file back again. Whew!

In order for this to work one must understand a few basic things. First, when a program is killed it is not deleted from the disk, but a code is removed so the directory will skip over it. With these instructions one can rebuild the code. Second, if another program has been saved on the disk since the file was killed, then there is a very good possibility that the program wrote over the previous one, in which case it will be impossible to recover the original program. One should try this process on a spare disk before using it on the real thing, for if something is done wrong then it may destroy some data if not the entire disk. I would like to thank Ray and Glenn Kreisel for their help in doing this process.

$$$

President's Page

March 15, 1984

BEWARE APRIL FOOLS JOKES

Date : Tuesday, April 3, 1984

Time : 7:30 PM

Place : New Canaan Country School
Frogtown Rd
New Canaan, CT

At the last meeting with were blessed with a number of demonstrations. Peter Mead demonstrated Pascal - 80 and we had a lively discussion of different languages. (This appears to be a hot topic.) Aron Bisberg demonstrated his bicycle selection program and described the Model 4 Basic and problems getting used to the machine. Hope Dunham showed us the new Model 4P and talked about her experience with the Model 4. And last of all I gave a demonstration of MID$.

At the next meeting, we will have a tutorial on Z - 80 machine language by Alan Abrahamson for the beginner. I will give a demonstration of the disk technique called Linked - Lists. This is a method of handling the quick sorting and searching of random access files.

Don't forget to pay up your dues before the June meeting and save $4.00. That means if you pay before June you pay only $20.00 as opposed to $24.00.

(Note: I heard recently that the TRS-80 Model 16 is a hot item now that it has Xenix working and can provide multiuser service.)

Voice of the '80
Index for April

Macintosh Revealed! .. Page 1
TIMEDAT.BAS ..... Page 2
MERGER/BAS ..... Page 4
Visicalc Formulas? ... Page 7
ROM CHECK Revisited .. Page 10
RX 80 Ref Card Part 1 Page 12
Ask Alan ..... Page 14
Z-80 OPCODES Part 1 .. Page 16
Retrieving Killed File Page 18
Model 100 Disk Drives Page 2
Directory Utilities .. Page 4
DIRPRT/BAS ..... Page 6
PRINTER LOWER CASE MOD Page 9
ROM TABLES--ROMCHK/BAS Page 11
RX 80 Ref Card Part 2 Page 13
WHITEOUT/SRC ..... Page 15
Z-80 OPCODES Part 2 .. Page 17
The President's Page . Page 20

The Voice of the '80 is a publication of the Fairfield County TRS-80 Users Group. Permission to reprint is granted for non-profit and non-commercial uses. Credit is appreciated. Submissions should be sent to the Editor at the address below:

PRESIDENT: Robert Jackson
VICE PRESIDENT: Herman Parks
SECRETARY: Pat Brinson
TREASURER : John Roy
EDITOR : Alan Abrahamson
ASSOC. EDITOR : I. M. Anonymous

Ask Alan HOTLINE: Evenings (6PM-10PM) 203/866-7883

This Newsletter printed by Sir Speedy Printing Center, Wilton CT

{{{{{{{{ This months cover creation by Alan Abrahamson }}}}}}}}

Licking, sticking, folding & stapling by: Herman and Jeff Parks

FAIRFIELD COUNTY TRS-80 USERS GROUP
% Alan Abrahamson
10 Richlee Road
Norwalk, CT 06851

FCUG

First Class

TO: