8708

Notes from the Vice-President:

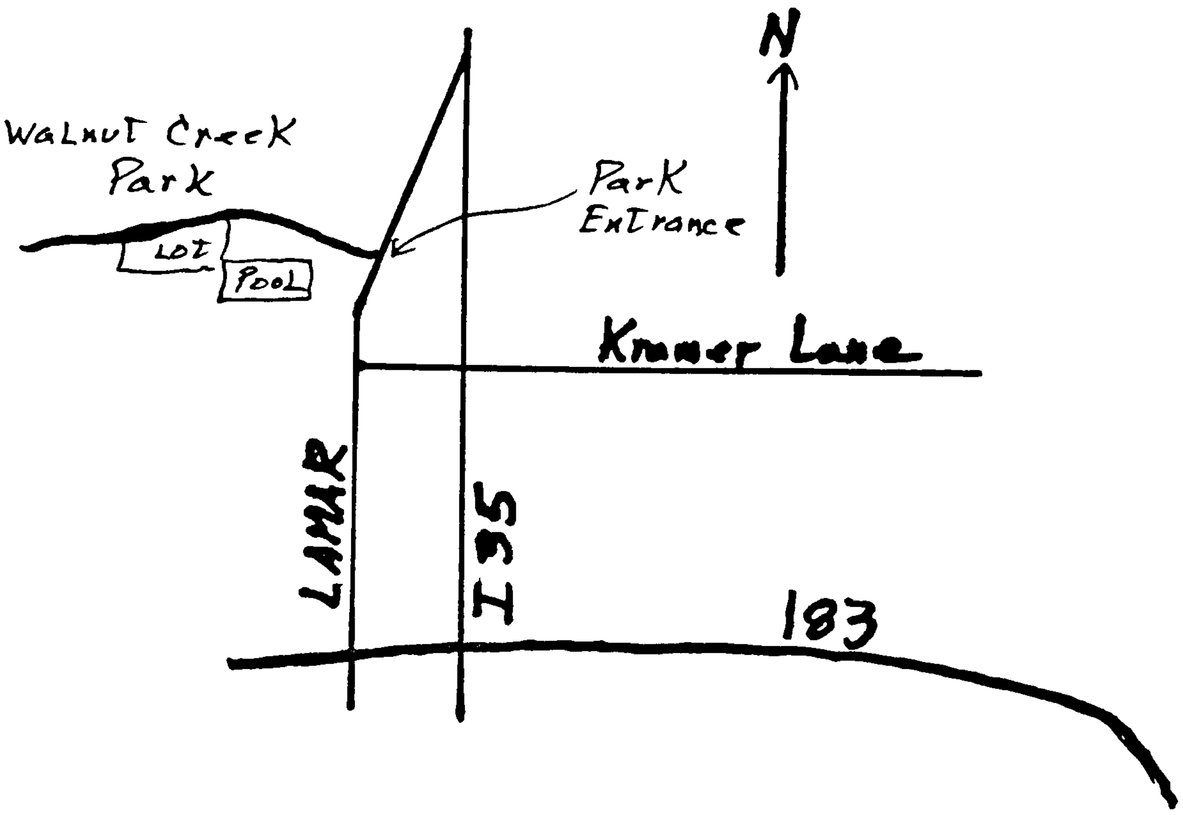
Apologies to all for canceling the June monthly meeting. What with no monthly newsletter, rain, several officers not being able to make it, a general lack of interest, witnesed by several months of poor attendance, etc. I took it upon myself after consulting the other officers to cancel the meeting. So go ahead IMPEACH me!!!
I did attempt to call all the regular attendee's to let them know.
I also trucked on down to post a notice. Four people showed up including myself. Two were among those who had indicated they couldn't make it.
OH WELL!!!

The monthly raffle of TI stuff (whatever attending members bring to donate) continues, and has been very succesful in reviving our treasury. Please check your closets for any surplus TI stuff which you might like to donate to our monthly raffle. Contact any of the officers if you are not able to bring the goodies to the meeting.

*******PICNIC/SWAPMEET*************

We will have a picnic/swapmeet on saturday OCT 3RD at walnut creek park in AUSTIN. We invite any and all who are interested.
The park is on north Lamar in north Austin, north of 183, located on all austin city maps. A rough map follows this paragraph.
Time will be from 10:00 till 1:00---maybe longer.
PLEASE SPREAD THE WORD TO ALL PRESENT AND PAST MEMBERS/NON-MEMBERS.
THE INTEREST OR LACK OF IT WILL DETERMINE THE FUTURE OF THIS USERS GROUP.
We will be located in the main parking lot next to the swimming pool.
Bring portable tables and chairs to display your goodies.
Bring munchies of all kinds. Bring your families.
More on what club might be able to provide will be in next newsleter after AUG meeting, discussion, voting etc.

Hank Kennedy V.P.



Programming in C - Part 3

Things have settled down at work again
and I have spare energy to spend on
other things like this series of
articles. Nevertheless I want to
apologize for my lengthy absence.
Let's now get down to business!

In the last article I mentioned that I
was having problems with interactive
I/O, well I gotten a chance to work on
them and have written this program to
demonstrate two methods of reading
characters from the keyboard.

```
#asm
    REF PRINTF
#endasm
#include dsk1.stdio
/* In order for this program to run, the following files must
    be loaded:

    DSK1.CSUP
    DSK1.CCLASS3;O      This program's object
    DSK1.PRINTF
    DSK1.CFIO
*/
main()
{char a[80],
      c;
 int i, j;

 i = 0;
 do
   {c = getchar();
    a[i++] = c;
   }
  while (c != 10 & c != -1);

 a[--i] = '\0';

 puts(a);
 puts("\n");

 for (j=0; j<i; j++)
  printf("%c = %d\n", a[j], a[j]);

 fgets(a,80,stdin);

 i = 0;
 while (a[i])
   {printf("%c = %d\n", a[i], a[i]);
    i++;
   };
}
```

This program demostrates the difference between the C routines getchar and fgets.

The getchar routine simply waits on the user of the program to press one key and release it. It then returns the key to the program. Our program stores the key in an array and increments the count of the number for characters in the array. This program then loops back to get another character. It does this until the enter key is pressed (c == 10) or an End Of File is reached (c == -1). It then replaces the enter key, which was stored in the array with a null character (which terminates all C strings) and displays the data collected. The next loop dumps every character in the array including the ASCII value of the characters. Why this program does this I'll explain in a bit.

The next thing that this program does is calls fgets which reads characters from the keyboard until an enter key is pressed. On the surface this appears to be just what our program did in the preceeding "do while" loop. Our program once again dumps the contents of the array to the screen along with the ASCII values.

If you run this program typing the characters ABC and pressing enter, then ABC and pressing enter again, you'd get the same results both times the array is dumped A=41, B=42, C=43. However run the program again and this time type AB, press left arrow and then C both times. First you'll notice that the left arrow doesn't seem to do anything. But when the program writes the string using puts you'll see only AC on the screen so it appears that something happened. It is the dump to the screen that shows the array contains A=41 B=42 =8 C=43. In other words the left arrow key did not really remove the character B from array. Not only that but the left arrow itself became part of the data in the array (=8)! It was when the left arrow was displayed to the screen that the cursor backed up and then the C was written over the B.

When we perform the same experiment on the fgets function, we see that the array contains only A=41 C=43. This is because the fgets routine does not return to the program until the enter key is pressed and takes care of the left arrows for us. (I think however, with the inclusion of a single if statement in our "do while" loop we can do just as good.)

Clearly, the fgets is what we wish to use when we want to accept input from the user. It already takes care of the left arrows (sort of) and terminates the data with a null to turn it into a C string.

In what cases then would we use getchar? In those cases where fgets doesn't do what we want of course! For example, fgets uses the left arrow as what is known as a destructive backspace. That means that as you backup over characters you erase them. If you make a mistake on the first character, you must back up over all the characters you typed, correct the offending one and retype the rest of data again. With

getchar we could create our own input routine and have it a little more friendly. (This is a good little project to do on your own.)

I think that that's enough for this time. Next time I'll show you a little program that I'm working on that uses the locate function to position the cursor and, baring major problems, I'll show you how to implement a BASIC like ACCEPT routine in C.

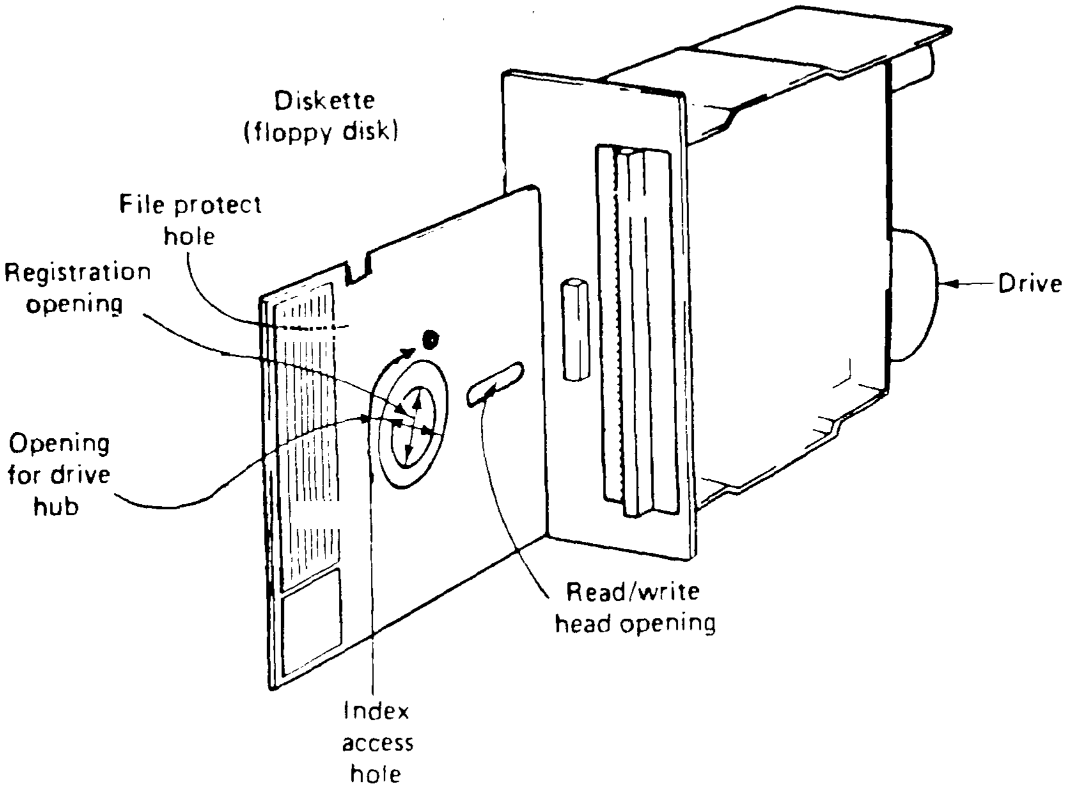
Mike Schmitz

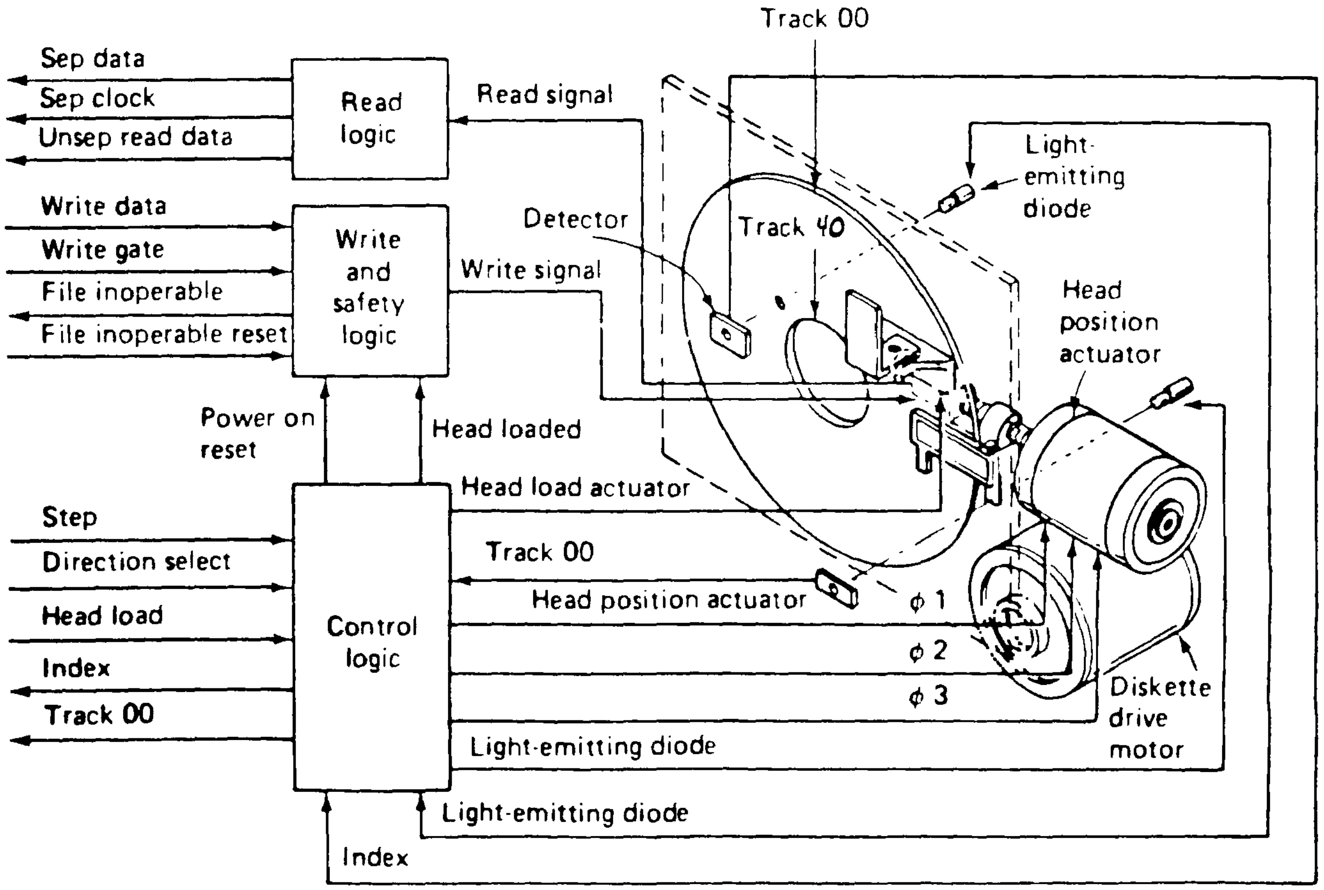
If you have been wanting a two-column printer utility program, that is, a program that will print side-by-side two columns of text, without messing around with the paper in the printer, then this program is a real starting point for you. Format your text as usual, setting up TI Writer to justify right margins. Run the program, access your completed file, and this program will load up the whole page and it will print out in two columns down the page. For more informatin on the line-by-line operation of this program, send a self-addressed stamped envelope to the editors, Fred and Amy, and we will send you a more comprehensive tutorial on this Extended Basic program.

```
100 REM **SIMPLIST**
110 REM * PRINTER LIST *
120 REM *FROM DISK*
130 REM FUNNELWEB FAM
140 OPTION BASE 1 :: DIM PRLN$(66,2)
150 REM *DEFAULT VALUES*
160 CALL TITLES :: SFIL$="DSK1.LIST" :: PDEV$="RS232.BA=4800"
170 CALL KEYCON
180 REM *NEW FILE ENTRY*
190 CALL OPTIONS(SFIL$,PDEV$):: ENDFILE=0 :: LINPUT #1:NEW$
200 REM *NEW PAGE ENTRY*
210 CALL PAGEBUFFER(PRLN$(,),ENDFILE)
220 CALL PRINTPAGE(PRLN$(,),PDEV$):: IF ENDFILE=0 THEN 210
230 REM *END OR NEXT*
240 CLOSE #1 :: CLOSE #2 :: CALL MORE(NM):: IF NM THEN CALL
SPEAK("GOODBYE"):: G OTO 250 ELSE 190
250 STOP
260 SUB TITLES
270 CALL CLEAR :: CALL SCREEN(11):: DISPLAY
AT(12,6)BEEP:"PRINTER LISTING"
280 SUBEND
290 SUB OPTIONS(S$,P$):: DISPLAY ERASE ALL :: CALL
TXTCOL(16,5)
300 CALL FILENAME(1,2,"EDIT AS NEEDED AND ENTER","N?")
310 CALL FILENAME(4,4,"SOURCE FILE FOR LISTING",S$)
320 CALL FILENAME(8,4,"PRINTER DEVICENAME",P$)
330 CALL YN("CHANGE MIND?","N",22,5,I):: IF NOT(I)THEN CALL
HCHAR(22,1,32,64):: GOTO 300
340 DISPLAY ERASE ALL :: IF S$="" OR P$="" THEN DISPLAY
AT(1,2)BEEP:"NO INPUT/OU TPUT POSSIBLE" :: CALL DELAY(500)::
GOTO 300
350 OPEN #1:S$,DISPLAY ,INPUT ,VARIABLE 80 :: OPEN
#2:P$,DISPLAY ,OUTPUT,VARIABL E 80
360 SUBEND
370 SUB PAGEBUFFER(PRLN$(,),EFL)
380 REM *NEW COL ENTRY*
390 PLN=6 :: COL=COL+1 :: IF COL>2 THEN COL=0 :: SUBEXIT ELSE
PRINT "":"**READIN G COLUMN #";COL:"":""
400 REM *NEW PARA INPUT*
410 IF EFL THEN PRINT "":" *":"***END OF FILE***":" *":"" ::
SUBEXIT ELSE CALL B ASICLINE(NEW$,EFL):: PRINT NEW$:""
420 CALL WRITECOL(PLN,COL,PRLN$(,),NEW$)
430 IF NEW$="END OF COL" THEN 390
440 SUBEND
450 SUB BASICLINE(N$,E)
460 N$="" :: IF NX$="" THEN LINPUT #1:NX$
470 N$=N$&NX$ :: IF LEN(NX$)<80 OR EOF(1)THEN NX$="" ::
E=EOF(1):: SUBEXIT ELSE LINPUT #1:NX$
480 PX=POS(NX$," ",1):: IF PX<2 OR PX>6 THEN 470
490 P=POS(N$," ",1):: IF PX<P THEN 470
500 NR=-1 :: FOR I=1 TO PX-1 :: C=ASC(SEG$(NX$,I,1)):: NR=NR
AND C>47 AND C<58 : : NEXT I :: IF NOT(NR)THEN 470
510 IF SEG$(N$,LEN(N$),1)="" THEN 470
520 IF VAL(SEG$(NX$,1,PX-1))<VAL(SEG$(N$,1,P-1))THEN 470
530 REM **CHECK QUOTES**
540 NQ,I=0
550 I=POS(N$,CHR$(34),I+1):: IF I THEN NQ=NQ+1 :: GOTO 550
ELSE IF NQ<>2*INT(NQ/ 2)THEN 470
560 SUBEND
570 SUB WRITECOL(P,C,P$(,),N$):: IF NC THEN P=6 :: NC=0
580 IF P>=57 THEN N$="END OF COL" :: NC=-1 :: SUBEXIT
590 CALL WRITEPAR(P,C,P$(,),N$)
600 SUBEND
610 SUB WRITEPAR(P,C,P$(,),N$)
620 P=P+1 :: IF LEN(N$)>28 THEN P$(P,C)=SEG$(N$,1,28)::
N$=SEG$(N$,29,LEN(N$)-28 ):: GOTO 620 ELSE P$(P,C)=N$ :: N$=""
630 SUBEND
640 SUB PRINTPAGE(P$(,),D$):: PRINT "":"**PAGE PRINT STARTED"
650 PRINT "":"**ASSEMBLING PRINTLINES":" AND PRINTING TO" ::
PRINT "":" ";D$
660 FOR I=1 TO 66 :: PRINT #2:TAB(9);P$(I,1);TAB(45);P$(I,2)::
P$(I,1),P$(I,2)=" " :: NEXT I
670 SUBEND
680 SUB YN(A$,B$,R,C,X)
690 DISPLAY AT(R,C)BEEP:A$ "(Y/N) "&B$ :: ACCEPT
AT(R,C+LEN(A$)+7)VALIDATE("YN") SIZE(-1)BEEP:A$ :: X=A$=B$ ::
R=R+2 :: SUBEND
700 SUB KEYCON :: DISPLAY AT(24,6)BEEP:"ANY KEY TO PROCEED"
710 CALL KEY(3,I,ST):: IF ST=0 THEN 710 ELSE DISPLAY ERASE ALL
720 SUBEND
730 SUB FILENAME(R,C,M$,D$)
740 DISPLAY AT(R+1,C):RPT$("-",LEN(M$)):: DISPLAY AT(R,C):M$
:: IF D$<>"N?" THEN DISPLAY AT(R+2,C):D$ ELSE SUBEXIT
750 ACCEPT AT(R+2,C)SIZE(-15)BEEP:D$ :: SUBEND
760 SUB MORE(NM):: DISPLAY ERASE ALL :: CALL TXTCOL(3,12)::
CALL YN("MORE LISTIN GS","N",16,2,NM):: SUBEND
770 SUB DELAY(A):: FOR A=1 TO A :: NEXT A :: SUBEND
780 SUB TXTCOL(A,B):: CALL SCREEN(B):: FOR I=0 TO 12 :: CALL
COLOR(I,A,B):: NEXT I :: SUBEND
790 SUB SPEAK(A$):: CALL PEEK(-28672,SP):: IF SP=96 THEN CALL
SAY(A$)ELSE CALL D ELAY(5*LEN(A$))
```

EXERPTS FROM THE PITTSBURGH USERS GROUP.

PITTSBURGH USER GROUP
P.O. BOX 18124
PITTSBURGH,PA.15236


Diskette
(floppy disk)
File protect hole
Registration opening
Opening for drive hub
Drive
Read/write head opening
Index access hole


Track 00
Sep data
Sep clock
Unsep read data
Read logic
Read signal
Light-emitting diode
Write data
Write gate
File inoperable
File inoperable reset
Write and safety logic
Detector
Track 40
Write signal
Head position actuator
Power on reset
Head loaded
Head load actuator
Step
Direction select
Head load
Index
Track 00
Control logic
Track 00
Head position actuator
ϕ 1
ϕ 2
ϕ 3
Diskette drive motor
Light-emitting diode
Light-emitting diode
Index

DISK DRIVE WOES
By, R.M. Bies

As our disk drives age, they become subject to problems which someone with the proper equipment can correct. (You have an indication of this problem when the drive will read and write reliably on a disk recently formatted on it, but the data so written cannot be read in another drive.) Also, with many of the half-height double sided drives, the heads are fragile, and if caught on the disk envelope, can be pulled out of position. Generally, the cost of replacement heads and the labor of replacement and alignment is greater than the cost of a new drive.

There are also less obvious sources of trouble not so exotic to remedy. I will deal with three which I have encountered: insecure mounting of the 12V regulator on the power-supply board, faulty power connections, and gummy head rails.

If it looks like the drive or drives are bogging down (paticularly in twin half-height instalations), and the drives ultimately stop and crash, it may be worthwhile to check the mounting of the 12V regulator. This is a TO-3 on the power-supply board, held down with screws. The board is its heat sink. I have found that tightening a loose mounting screw can alleviate this problem. (Tighten with care, of course.) You know the power-supply is being overloaded when the measured voltage on the 12V line on the drive drops off well below 12V as the drive slows.

If that wasn't it, and particularly when moving the drives around sometimes seems to fix it for a while, check the connections in the four power lines to the board on the disk drive, again, particularly in a twin drive installation. These connections come in two slightly different sizes and for some reason, the male pins seem to always be of the smaller size, the female of the larger size. It may help to spread the male pins slightly, or to close the female connectors slightly. They are not gold-plated so are subject to oxidation--contact cleaner may also help.

Now for the most obscure. One of my drives regularly would not read certain tracks on a disk, but the disk worked fine in the other drive. That the drive woud read most tracks suggested that the electronics was OK, that the other drive would read the whole disk suggested that it was not an ordinary alignment problem. It looked like a mechanical problem. Nothing seemed out of place with the drive removed, no foreign objects present. Yet, the heads seemed to offer slightly more resistance toward the end of their travel. A drop or two of lighter fluid on each rail with the assembly exercized by hand back and forth seem to yield a smoother action. Indeed it worked. The rails were gummy enough at the extreme of travel to defeat the electronic positioning procedure.

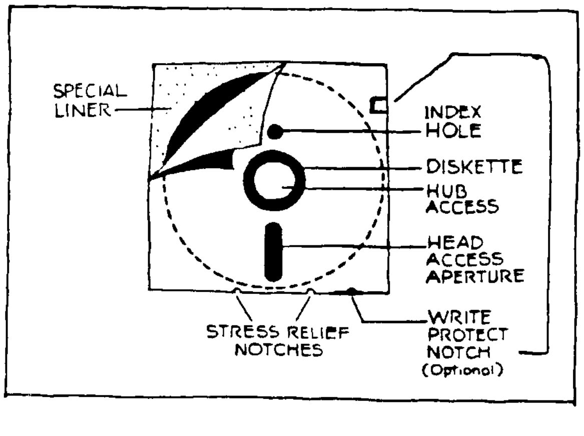
\\\

SIMPLY PUT...
How A Disk Drive Works
By, Fred and Amy Mackey

The disk is placed into the drive, the door is closed, and a spindle hub inside the diskette hole spins it around very quickly, at about 300 RPM. A magnetic read/write head moves towards the hub or out to the edge. The combination of spinning and head movements allows data to be placed on any part of the disk. Data is written on or read from the disk as it spins around inside the disk drive. The characters are stored as a series of magnetic pulses treated as zeroes or ones, called bits. Eight bits is a byte and is one unit of data.

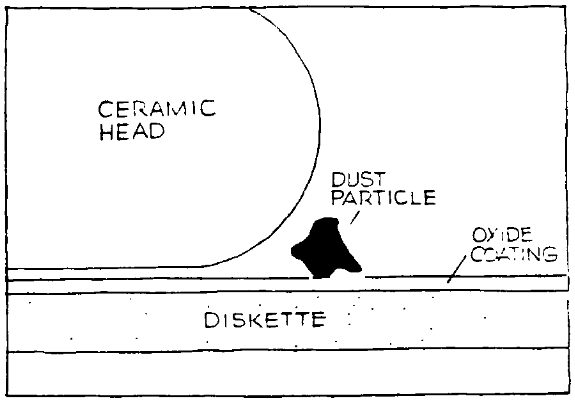
A double sided disk drive has two of everything and can read and write to both sides of the disk without flipping the disk over. A double density disk drive can hold twice the usual number of bytes.

The data is read/written on concentric bands called tracks. The drive sees the correct track through the index hole on the disk. Both sides of the disk jacket and the disk itself have this index hole. When the three are lined up, a beam of light passes through them and strikes a photo recepter which tells the drive it is the start of the track. Each track on a disk is broken into equal areas called sectors.



MAGNETIC MEDIA DEFECTS

By, Fred and Amy Mackey



As we are certain you are well aware, all computers using external storage systems rely on magnetically created electrical impulses for their memory. Whether using cassette tapes, cartridges, floppy disks, hard disks, or other storage systems, these impulses are what make computers function. The physical devices that accept and hold the magnetic code are called the media. And all media require periodic attention.

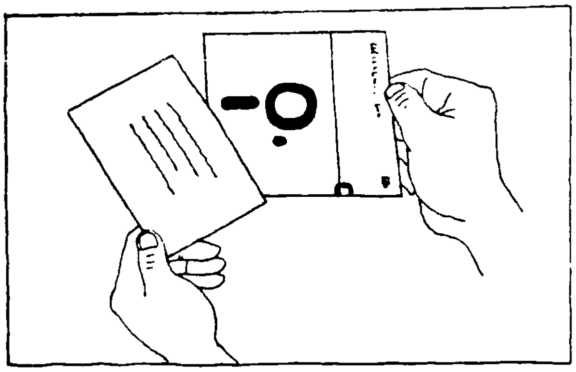
If you've ever tried playing a "wrinkled" cassette tape on a cassette recorder, you know there is a problem. Besides producing a skip at the wrinkle, the tape will be weakened in that spot and will eventually break.

The same holds true with magnetic media used for computer storage. Both the media and the programs and data they contain may be damaged or destroyed by such things as heat, static, magnetism, polluted air, chemicals, dirty drive heads, grease or oil from fingertips, excessive humidity, excessive dryness, etc. Even brand new media is not exempt from these problems.

Older media is subject to more problems, such as the base material holding the oxide may turn brittle and become useless with age. The adhesive holding the oxide to the base may deteriorate. Or possibly the software may just wear out with use.

The magnetic media can easily be damaged through abuse or negligence. Therefore,it is important that you be as careful as possible in your handling of it. This means running regular maintenance checks to keep your media in shape and doind such things as cleaning the heads of the disk drive and keeping your computer work area as clean as possible. And most important, get in the habit of making back-up disks for all of your software.

The following are some rules that you should follow to give your disks and cassette tapes the longest life span as possible.



1. Always store your media in dustproof, non-metallic containers.
2. Avoid touching the magnetic surfaces of the disks or tapes.
3. When labeling a disk, write on the label before putting it on the disk. If you must write on the label after it is on the disk, use a soft-tipped marker, and make sure the ink is dry before putting it into the drive.
4. Store all media away from sources of heat, as well as from humidity. (This means that the basement is not a good home for your computer!)
5. If you just walked across the room dragging your feet on the carpet, first discharge the static electricity by touching a grounded metallic object before you pick up a disk.

\\\

NEWS YOU CAN USE

Newsletter Spotlight

This month the Newsletter Spotlight is again on the Mid South User's Group December Newsletter.

<u>What Is A Ramdisk?</u>

By, Gerald Smith

A ramdisk is a certain portion of RAM memory that has been partitioned and set up to imitate or "emulate" a disk drive. For all practical purposes, the Ramdisk IS a disk drive. It just stores the files in RAM rather than on a floppy disk. The main advantage of this is <u>speed</u>. If you have ever seen a hard disk drive in operation, then you have a good idea of how fast a Ramdisk oprates. Or think back to when you first upgraded your TI from cassette to disk operation. You were probably thrilled with loading in a program in 10 seconds instead of 2.5 minutes. Now that time has been cut down to 1.5 seconds. Of course, the time differential will increase the larger the program. Once you get used to using a RAMdisk, you'll want to store large files into it that you will be accessing a lot in one session at your computer. MULTIFLAN would be a good candidate for that. In comparison, a program which takes 6 seconds to load from a disk drive will take 2 seconds to load from a Ramdisk; A program that takes 16 seconds to load from a disk drive will take only 4.25 seconds to load from a RAMdisk.

MULTIPLAN CHANGES...

I'm not sure where this file originated to give credit, but it looks to have some interesting information.

Here are some modifications you might want to make to your multiplan disk... This is a description of how (with Disk Fixer - or something like it) the MPINTR file of Multiplan can be modified to permanently change: Default Disk Drive, OVERLAY file name, MPHLP filename and Default Printer assignment.

The 2nd segment of the MPINTR file contains the default entries for:

- Disk drive for Transfer commands
- OVERLAY file name
- MPHLP file name
- Default printer assignment

Each entry starts with a 1-byte length field (non-inclusive) and (except for the printer assignment field) unused bytes are filled with HEX zeroes (the printer assignment field is space- filled). The location of the fields are:

Location	Entry
>0055 - >005C	Default Data Disk Drive
>005D - >006E	OVERLAY file name
>006F - >008A	MPHLP file name
>00BD - >00E5	Default Printer assignment

For example, to change the printer assignment to 'PIO' the entry in >00BD should be: 03 50 49 4F. Note that the length value (03) does not include itself. The rest of the field (through location >00E5) should be filled with spaces. EDITOR: To further explain this, the '03' is the legnth of the word 'PIO', if you wanted to use RS232, the the legnth would be 05. The other values, '50 49 4F' is the hexidemcila equivalents of the characters 'P I O'. After you enter the legnth of the string, I would suggest that you edit the rest of the sector using ASCII instead of HEX as you may then enter the characters as 'PIO' and the program will convert to HEX for you.

The easiest method of getting to the right segment of the file is to initialize a new disk and copy MPINTR to it. Then, unless disk errors during the initialization messed things up, the needed segment will be at sector address >0023. Use Disk Fixer to change the data and write it back to disk. Then (after making sure you have a back-up copy of the original MPINTR file!!) copy the new MPINTR file to the TIMP disk.

The Multiplan cartridge requires that the 4 program files be loaded from a disk titled TIMP, but once they are loaded they are no longer used. The ability to change the disk volume name for the OVERLAY and MPHLP files means that you can put them on separate disk that you switch to after the program files are loaded. This will save you 117 segments of disk space (275 if you don't need the MPHLP file!) that could be put to better use.

One last note. The time it takes to initialize Multiplan and the response time when it is working with the OVERLAY file is effected by the location of the files on disk. You can load the files in a desired order by copying them one at a time to a newly initialized disk. The best order appears to be: OVERLAY, MPHLP, MPCHAR, MPDATA, MPINTR, and then MPBASE. The last 4 files are only accessed at start-up of Multiplan and in that order. The OVERLAY file should be loaded first because the opened at each use, thereby causing disk access to the directory area at the beginning of the disk. (The further the files are from the disk directory a the beginning of the disk, the longer the seek times.)

PROBLEMS AND SOLUTIONS

ANOTHER FUNCTION KEY?

By John Parkins

Did you say another Function Key? Or, does it sound good to you? Ok, so you are a professional typist and would have absolutely no use for another key on the board. But, they did put two shift keys and made them available to both sides of the keyboard didn't they? If you are that professional, then you may as well just go ahead and skip the rest of this article and save yourself some time. This is mainly focused at the non-professional typist like me, who has short fingers and has a very difficult time trying to do a function 2 and 3. The 3 is ok, but it really streches out for the function 2. Need I say more? It is impossible for me to do a function 1 with only one hand since I end up hitting five keys at the same time, which includes the function, space bar, the 1, the 2, and the Q. Some of my friends seem to have no trouble at all as a one handed typist doing a function 1. They ought to be good at basketball too!

At the TI-Faire in Chicago, 1985, there was a table setup where they were displaying a TI-99/4A keyboard with another function key placed beside and to the left of the A key. I am sorry but I do not remember the name of the firm that had the display or I would give them a great big plug. It was a valuable idea, I thought, but they were asking $35.00 for the keyboard and I remember thinking that was a little too high in price. At that time Radio-Shack was selling their 4A replaceable keyboards for only $3.95.

When I went back to the Faire again in Nov.,1986, I looked in the same area for that keyboard, but the firm was not there and neither were the keyboards. I remember all year as I sat in front of my computer, if I had only gotten one of them then! "Too little too late," as the old saying goes.

I got agrevated with myself and decided to do something about it. I got one of those Radio-Shack keyboards and cut a key off of it, and mounted it on my spare beige console with 5 minute epoxy. Then came the job of soldering it to the ribbon cable of the existing keyboard with a couple of short jumper wires. I used the wires #5 and #9 to make the connections. I cut out the little square to the left of the A key in the top of the console, (the lid) to make room for the extra key to protrude evenly with the rest of the keys making it look like it was really supposed to have been there all along from the start.

It works great! Now I can do a function 1, 2 or up to a function 6, even use the function and the arrow keys with only one hand. Let me tell you, I have been so justly rewarded with the ease of use that I probably never will learn how to type correctly now. That's beside the point though. If I have sparked your imagination and you may find it useful, then I rest my case.