

THE GUILFORD 99'ER NEWSLETTER

VOL. 5 NO. 5

MAY 1988

Janice Snyder, President
Mack Jones, Secretary/Treasurer
BBS: (919) 274-5760 (OPUS)

Bob Carmany, Vice President
Herman Geschwind, Program Library
ROS (919)-621-2623

+++++
The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are \$12.00 per family, per year. Send check to 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (George von Seth, Editor)
+++++

OUR NEXT MEETING

DATE: May 3, 1988. TIME: 7:30 PM PLACE: Herman Geschwind, 1714 Clarendon Drive, Greensboro, N. C. Either Market Street or Friendly Ave. to Holden Road, turn North on Holden to Hobbs Road, turn left on Hobbs to Clarendon (second street to right) to 1714 on the right.

Program for this meeting will be a demonstration of modems, terminal programs, and live access to one of the local BBS.

MINUTES

By: Mack Jones

The April 5th meeting of the Guilford 99'er Users' Group was held at the Glenwood Recreation Center on Chapman Street.

Meeting was called to order by President Janice Snider at 7:41 P.M. The March minutes were read and corrected.

OLD BUSINESS:

- 1. Bob Carmany has volunteered to take the office of Vice President that was vacated by Janice Snider.

NEW BUSINESS:

- 1. Herman Geschwind volunteered his home for the May meeting so members can have a first hand demo of the accessing of a BBS. Herman stated that if there was incimate weather such as a thunderstorm, no demo would take place.
- 2. Bill Woodruff donated the parts for a power supply for the new club drive that was donated by Herman. The building of the supply and case was given to George Von Seth and "Mac" Jones.

3. There was a discussion on the merging of the Winston Salem group with our group. It was decided at this time that we will stay in Greensboro, rather than trying to find a meeting place in Kernersville at the Library.

4. A motion was made and seconded that the President respond in writing to the Winston Club to the effect that we will welcome any of their members who want to visit and join our club.

5. The disk of the month was offered by Herman, and 5 were sold.

The meeting was then turned over to VP Carmany, who gave a demo on unlimited speech using the extended basic cart and software. Bob also included some musical renditions in his program.

Meeting was adjourned at 10:00 P.M. There were 9 members and 2 visitors attending.

Respectfully submitted,
L.F. "Mac" Jones, Sect./Tres.
Guilford 99er Users' Group

ATRAX TRACKS

By: Bob Carmany

In last month's newsletter, we looked at the word processing menu of F'WEB and this month we are going to look at the Editor/Assembler part of the program. When you choose the 'EDIT/ASSM' menu option, you are presented with yet another menu of options: 1) Editor, 2) Assembler, 3) Loaders, 4) c-Compiler, 5) Disk-Patch, 6) Utility, and 7) Reset.

The first option, 'EDITOR', loads the word processing editor with a few modifications to aid in entering your assembly language source code. First of all, the word wrap has been disabled so that 'Reformat' won't merge together 100 or so sectors of code into an unintelligible mass of characters. The tabs have been pre-set to the defaults necessary for your source code listings, and the second character set is loaded with its larger characters for easier reading. All of these modifications are very welcome if you do much assembly language programming.

The second option, 'Assembler' loads a modified version of the Assembler that came with the original E/A package. The file name is retained in the mailbox and it generates a suggested object code filename and the error handling is much improved. You can even access the disk directory with <FCTN 7> if you have forgotten the filename (file marking is not available, however). All in all, it is superior to the one TI supplied from the convenience standpoint.

The third option, 'LOADERS', is really an article in itself! When you choose it, you are presented with yet another menu of options to load A/L programs of just about any description. 'Textmode (TI/Wr)' will load any of the programs that you would normally load with option #3 of the TI-Writer cartridge. HYPHENATOR and SPELLCHECK are two that come immediately to mind if you haven't otherwise installed them at another point in the F'WEB system.

The second choice will load program image files that would normally be loaded from the GPL environment. That covers a wide range of program like ARCHIVER 2.4 and various others.

The third option loads programs that you would normally load from the E/A cartridge 'RUN Program File' option. Most of the program image files follow either this format or can be loaded with the previous loader --GPL.

The fourth selection of this sub-menu is the standard 'Load and Run' E/A option. It handles both compressed and uncompressed object code with equal facility. In fact, it is a complete replacement for the E/A loader of the same name.

The fifth selection is the Scriptloader. This is a unique loader through which you can load a series of A/L files by creating a D/V 80 (text file type) loader to load the whole "batch" in sequence. Hence, it is also called a "batch file" loader. The instructions for this little beauty are in the documentation and are easy to follow.

The next selection (#6) is a loader to load files into the low 8K memory block. This is handy for relocatable code that you may want to "force load" somewhere other than where it was designed to go. You may have to make some modifications in the code, however. It works particularly well with some of the E/A package programs.

The last option is a Load and Run All-Memory" option. It is exactly what its name implies and it is fully documented in the F'WEB doc file.

This selection of various loaders will load virtually any program that can be loaded on the TI. There are, however a few exceptions but it is the odd program indeed that cannot be loaded with one of the 'LOADERS' options of F'WEB.

The next option on the 'Edit/Assm' menu is 'c-Compiler'. It loads a modified version of c99 (all the files aren't included in the package). Vn 3.0 of the c99 compiler loads very nicely with it and it re-enters F'WEB very elegantly --- it is a simple way to assemble and compile your c99 programs from a single software package!

'DISK-PATCH' loads a modified version of DISKPATCH or DISKO --- a disk sector editor that was released by TI prior to leaving the market. Tony has re-worked the program to include a byte counter and a disk directory can be displayed if the DD file is present. DISKO is easy to use and it is ideal for small editing jobs that don't require something 'heavy-duty' like John Birdwell's program.

The next option 'UTILITY' loads a file with the filename T1. You can alter UTILITY with CONFIG so that it will load any program image file with the name T1. Then, all you have to do is to rename the file of that favorite program for it to load out of this option.

'RESET' is the last option in the series. It is the preferred method to exit from F'WEB if you are not going to turn off the computer. It clears the filename from the 'mailbox' and resets all of the pointers. When it is chosen, it changes to 'QUIT' and selecting it again causes the program to exit to the title screen.

So, there you have it, an overview of Tony McGovern's Funnelweb package. It was the inspiration, I suspect, for John Johnson's Horizon menus and many of the other programs that have come since it was first introduced a couple of years ago. Anyone who doesn't have a copy of this fine program is simply living in the past. With an Extended Basic cartridge, you now have no need for either TI-Writer or the E/A cartridge. Anything that they can do can be done (usually easier and better) with F'WEB. I would encourage all of you who are using the program to send Tony McGovern a software donation if you haven't already done so. A kind letter would also be nice. I can tell you from personal experience (and two years' correspondence) that he is one of the true gentlemen of the TI world!

RAMBLING BYTES

BY: Mack Jones

For those of you who have not attended the meetings lately for one reason or another, I would like to say we now have a mini 32K memory expansion kit in the console. It was a lot of fun building this kit, and when looking at the 4 piggy backed chips on their socket, you wouldn't think it would ever fit inside that console!! I did have one trouble to be honest, I "kneled" one pin when I was plugging in the first chip, so naturally, none of the other chips would work either. I was looking for something hard such as a wiring mistake (yep, I can make 'um too!) so I really didn't check the pins for awhile. Next time that will be the first thing I check!! We used it for the April meeting and it performed just as if we had one in the PEB.

The new terminal emulator program "Telco" is making a big splash in the TI community. There is one thing you must remember...you must have your RS232 card available when using or configuring Telco, as it looks for dsr in the card and will lock-up without it. Bob and I both experienced this lock-up when we tried to format the program without using the RS232, and at the time, thought a bug was in the program. Also in the docs, it says that only the TI memory card will be recognized at this time. I love the idea of being able to use macros in the program. Also, there is a conference enhancement that as yet I haven't tried, but Bob is checking it out.

I am really looking forward to meeting at Herman's house for the May meet. Any of you who have not as yet seen an actual session on a BBS should try to be there. Herman will access some local boards and we might even pressure him to access CompuServe!

Speaking of BBS', I think Herman and Bob have uploaded just about anything you could ever want as far as files, games, and utilities on the RDS, which is our favorite TI board. Danny Post tried a different software for the board, but it failed, so we are back up again on the old software which is, in my opinion, just as good as any. Any of you with modems should be able to download from this board. Just dial (919) 621-2623 and if you are a new user, fill in the questionair that will get you access to the board. From then on it is there for you 24 hours a day. (And free)!!

I would like to wish a very HAPPY BIRTHDAY to our friends in Great Britain that are celebrating their first year as a group. Congrats to Donald and Jo Ann Copeland and the members of the EAST ANGLIA REGION 99'ERS. They have a very fine newsletter which you may check out if you will let me know when you want them.

Sure seems nice to be able to get to a meeting with the sun still up once again! Even tho Boone and some others are still having snow, we are sopping up some warm sunshine in the valleys. Come on out to the May meet at Hermans' and enjoy some warm fellowship with your fellow members. 'Till then happy hacking.

MACHINE CODE MASTERY

By: Tony McGovern

The ultimate way to get at the real potential of the TI-99/4a is to write or run machine code programs. The next best thing is TI-Forth, but that's grist for another mill. Originally TI did not intend that users would ever write their own machine language programs and provided no hooks at all in console Basic to link to machine language routines, or to allow direct access to machine functions. The same sort of corporate marketing contempt for the customer led them to put calculator keys on the original TI-99/4. They weren't and aren't alone in that attitude of contempt for the user look at the IBM PCjr years later, or the Apple Macintosh.

And when they did bring out the expansion system, it still did not realize the potential of the TMS-9900 processor because of the fractured memory map and conversion of the 16 bit data path to successive 8 bit slices for all but a small part of CPU memory space, adding that insult to use of external memory with wait states. However when TI finally made machine code available to users they did it in style, adapting their mini-computer software for the purpose. Some of the programs supplied still contain traces of their origin, such as memory mapper instructions relevant only to the larger 990 minis.

First let's look at how machine code programs are recorded as disk or cassette files, and then survey the modules which allow these files to be loaded and executed. These files come in two forms. The most direct form is as memory image files, in which the actual contents of a block or blocks of memory are stored, with control information appended. These are known as PROGRAM files in TI-99/4a language (and correspond to .COM or .CMD files in other systems). TI Basic and XB programs are also stored in this format, which can be saved to and loaded from cassette as well as disk. The other kind, usable only with a disk system, is the assembly tagged OBJECT file. In normal usage of the TI-99/4a, object files are created to be relocatable in memory by the loader, and the programmer does not have to know explicitly where the loader has placed them, and calls their entry points up by name. None of the primitive CSR or suchlike business. There are 4 (maybe 5) modules available which can load and run machine language programs. They all have different capabilities and limitations.

Editor/Assembler

~~~~~

The primary one is EDITOR/ASSEMBLER which is necessary for creation of relocatable object files (... well you could write one with a word processor but that would be masochism of a high order, exceeding even direct POKEing of machine code bytes). E/A will load any of the machine code file types mentioned above, from its menu screen. The LOAD AND RUN option handles both uncompressed and compressed tagged object files, and will resolve REFERENCES by name from one object file to another, or to standard system names. Uncompressed object files represent bytes in Hex notation, and take about twice as much disk space as compressed object files. Invocation of LOAD AND RUN re-initializes the memory pointers completely while loading the system utility routines such as VMBW from GROM storage, so if a sequence of file loads is interrupted by an error, it must be started all over again.

E/A adds CALL LOAD and CALL LINK to console Basic to allow these same object files to be loaded and accessed from Basic. The standard utilities such as NUMREF for communicating with the Basic program must be loaded as a separate file BSCSUP.

E/A will also load and run from the RUN PROGRAM FILE option, program files of machine code, prepared according to a specific recipe as SAVE files. The details of these will be a subject for HV99 News articles in the future. It will load them from cassette as well but I can't see anyone doing that in preference to using disks, unless perhaps they have installed the TIUP internal memory expansion mod in the spare console that gets taken away on holidays. No provision exists in Basic to load SAVED program files from Basic as they could overwrite part of the Basic program in the VDP on the way in.

#### Extended Basic

~~~~~

The next module which can load assembly code is our old friend EXTENDED BASIC. This is much more limited than E/A in what it will handle. Firstly it recognises only the 8K low memory area from >2000 to >4000 for loading relocatable object code. Absolute code can be loaded, or RAM buffer areas used in the lower part of high memory once it is known how far down this is filled by the XBasic program. The loader does not handle external REFERENCES, and the utilities loaded by CALL INIT in XB are missing the most useful one - DSRLNK, and GPLLNK as well. The Basic support utilities are loaded by CALL INIT from GROM. The assembly source code has to locate them with EQUates. A minor difference from E/A is that CALL LINK always hands over control in the GPL workspace. Programs written to LINK to E/A Basic will almost always need at least minor modifications

to LINK to XB successfully.

The operational hangup with XB is that the loader is written in GPL and is painfully slow. A long assembly routine, such as Text to Speech, may take several minutes to load (shades of the Commodore 64's disk system). The usual way round this is to load an assembly language loader which in turn does a faster load of the longer program. The great virtue of the XB module that sets it apart from the others is that it supports auto-RUNning from disk, as soon as the module is selected, of an XB program DSK1.LOAD which can then load further programs. The other reason for preferring XB is that it is a vastly more powerful language than the mildly enhanced console Basic offered by E/A. Unlike E/A it can never load machine code programs without Basic as an intermediary.

Mini Memory ~~~~~

This module has its own particular charm as the only one which allows access to machine code without the 32K memory expansion and using cassette storage. In this mode a LINE by LINE (or immediate input) Assembler allows standard TMS-9900 mnemonic assembly code to be entered in a restricted format. This is a descendant of TI's board level 990 evaluation systems. Only 700 odd bytes are available in the module's 4K of CMOS RAM after loading the assembler but I can't imagine anyone wanting to do programs longer than that with L by L. Still it's enough to do a pretty fair Game of Life program. MM also contains a full set of system utilities and Basic support routines in ROM and EASYBUG in GROM, a monitor program that is useful but much less powerful than the E/A DEBUG.

MM is even more useful in a fully expanded system. It does not provide the Editor and Assembler features of E/A, but offers more scope for loading and running programs. Firstly there is 5-6K more RAM available, 4K of CMOS RAM in the cartridge and the saving of space in RAM because of the utilities in cartridge ROM. Its principal deficiency as compared to E/A is the lack of a PROGRAM file loader, but this can be easily remedied by writing your own to reside in MM cartridge RAM. Even the L by L Assembler, as well as EASYBUG, remains useful for occasional little purposes anywhere in RAM, and I have prepared a disk based version for convenience.

E/A object code, even compressed, is loaded successfully as long as REFs are used for system utilities and Basic support routines. EQUates as used for XB code will only get it right for one module. The loader has one more space, cartridge RAM, to place relocateable object code as a last resort. I have not yet experimented to see whether the loader will link object files with external references as the E/A loader does. The MM manual, never a fount of information, is silent on this point.

MM does not erase its DEF table unless it is explicitly done by one of several means. The table survives a return to the title screen, and even switch-off if the internal battery is still alive. This is different from E/A's workings, and must be taken into account for better or for worse. Code in memory expansion does not survive switch-off even if its name lives on.

TI-WRITER ~~~~~

Now we leave the modules which can load files under any name for one which loads program files with particular names. TI-WRITER tries to load an E/A type SAVE file from DSK1 under the name EDITA1 (and successor filenames) when the EDITOR option is selected from the menu screen. If you have followed the E/A manual's advice on using the SAVE utility with TOMBSTONE CITY as the victim, take the file so generated and place it on a fresh disk under the name EDITA1 and place in DSK1. Then fire up TI-WRITER, choose the Editor option, and see what happens. Extension to Formatter and Utility options are obvious. It may provide light relief to heavy TI-WRITER sessions. More seriously, short of writing a PROGRAM file loader to be loaded by XB using DSK1.LOAD, it is the nearest that the TI-99/4a comes to an auto-loader for machine code program files.

FORTH TUTORIALS

By: Lutz Winkler

FORTH TO YOU TOO! - SESSION 3

As I mentioned earlier, there is an elegant way to boot whatever you want your system disk to be ready for, but before we can proceed with the installation of an autoboot we need to consider the following :

Since FORTH is a disk-based system (none of it comes in a command module or in the console itself) it occupies memory which otherwise would be available for programming. That - in my opinion - is the reason TI provided many of the utilities as LOAD OPTIONS. Look at the menu and also Appendix F. Some of the options, i. e. the editor, are essential, others are rarely needed. For instance, if you are not programming in Code there is no need to clutter up the memory with -CODE and -ASSEMBLER. Similarly, if you aren't going to operate with graphics then there is no need for -VDPMODES etc. It is not very likely that you will run out of memory while still in the learning process but why boot unnecessary stuff? I consider only -DUMP, -COPY and -PRINT along with the editor as essential for now. To see how fast memory is occupied even with your extra 32K, do this (assuming you are in FORTH):

```
: FREE SP@ HERE - . ;  
(colon FREE SP@ HERE minus dot semicolon)
```

Now enter FREE. You should get an answer of about 14140 (9790 if you opted for the 64-column editor). If you want to see how fast memory shrinks with each LOAD OPTION boot a few more, but enter FREE in between them. (If you are convinced enter COLD.)

Here is my suggestion for a beginner's boot:

1. -PRINT, to be able to list or dump screens
2. -COPY, to copy and move screens
3. -DUMP, to look at the parameter stack
4. -BSAVE, this makes the binary image save (and load) possible

If you want to use a printer there is one more item to check. Look at SCREEN 72 in the manual or - for practice - call it up from your disk. Look at where it says " RS232.BA=9600". This routine is written for a serial printer operating at 9600 BAUD. If you have a PIO printer you must modify #72 as shown below and FLUSH it to your system disk.

```
SCR# 72  
Line 0 - no change  
1 - no change  
2 0 0 0 FILE >PIO BASE->R HEX  
3 : SWCH >PIO PASS @ 10 + DUP PAB-ADDR ! 1- PAB-VBUF !  
4 SET-PAB OUTPT F-D" PIO" OPN 3  
5 PAB-ADDR @ VSBW 1 PAB-ADDR @ 5 + VSBW PAB-ADDR @ ALTOUT ! ;  
6 to 15 - no change
```

To make sure that everything is ok with your new version of #72, enter -PRINT, let the system boot it, turn on your printer and when you see the 'ok' type:

```
SWCH ." THIS IS A PRINT TEST" CR UNSWCH
```

Make sure there is a space between ." (DOT-QUOTE) and THIS. If your printer responds with THIS IS A TEST, pat yourself on the back and play with SWCH ." xxxxxx" UNSWCH some more. If not, you will have to start over again, and this time pay close attention, particularly to spaces!

Before we proceed with the actual set-up for your autoboot take a quick look at any SCREEN between 8 and 19, no not in the manual, on your display (remember nn EDIT). Not much there that's legible, but believe it or not on those few SCREENS resides every FORTH word that is identified in the Glossary as a RESIDENT word, only they are saved in a binary form. We will do the same with the LOAD OPTIONS you decide upon by the use of BSAVE. So let's go. First, start off with COLD, then boot your options by entering the appropriate words (-PRINT, etc) and as the final one -BSAVE. Find the apostrophe key (FUNCT 0 - that's 0 not ZERO!). This is also a FORTH word pronounced TICK (page 3, Glossary). Now enter:

```
TASK 22 BSAVE . ( tick TASK 22 BSAVE dot )
```

Here is what's happening: We are saving in binary form all that has been added to the dictionary (by booting the LOAD OPTIONS) starting at screen 22. We can afford to wipe out 22 and some of the following screens because they contain the 64 column editor which you have either booted already (so it's in the autoboot dictionary) or you aren't going to use it. The

final dot will print on your display the first screen available after the BSAVE. All other LOAD OPTIONS are still intact and can be booted when needed. Now for the finishing touches enter

EMPTY-BUFFERS 3 EDIT

and carefully erase all but lines 0,1,2,13 & 15. On line 2 change 20 LOAD to read 22 BLOAD. You might want to replace the word BOOTING on line 0 with some other phrase which would let you know that you are using your new system-disk. On line 5 put: 0 DISK_LD ! and depending on how many drives you have and whether they are single or double sided enter ONE of the following:

(if you have only one single-sided drive you can skip this)
90 DISK_SIZE ! 180 DISK_HI ! (for 2 single sided drives)
180 DISK_SIZE ! 180 DISK_HI ! (for 1 double sided drive)
180 DISK_SIZE ! 360 DISK_HI ! (for 2 double sided drives)

Note that these words use the underline, not the hyphen. (If you have double DENSITY drives it is not quite that simple, you have to make several modifications to screens 33 and 40 as well the disk header before you can utilize DSDD to its fullest extent.) After you have FLUSHed your edit to the disk, go COLD and - with any luck - you will have a working, auto-booting system disk.

As I mentioned before, you should be learning HOW Forth works and get Brodie's STARTING FORTH (or PROGRAMMING FORTH by Chirlian.) These tutorials will continue, but are not intended as a substitute for a textbook. It's time to start learning and to do that read Chapters 1 and 2 of STARTING FORTH. I went through the book first and annotated each page with the corresponding remarks from the manual's Appendix C (Notes on STARTING FORTH). This helped to lessen my initial confusion (at least somewhat).

*** END SESSION 3 ***

COPYCATS

By: Jack Sughrue

It is 3:30 in the morning. You can't sleep. You try counting spreadsheets, but nothing seems to work. One way to cure insomnia is to get a pile of, say, 15 disks full of games and utilities and tutorials and generic stuff. Then get a pile of blank disks (Keep all these things on hand for those nights of insomnia.) and copy the full pile onto the blank file. You'll finally have those backups you've been wanting in case something Dreadful happens to Guess My Number and Loan Amortization and other impossible-to-live-without files.

Anyway, shove in your Disk Manager and start initializing all those 15 disks. This, alone, may put you to sleep, out if that fails, copying the disks - file by file - is sure to do it. Great for insomniacs; not great for making backups.

So, if sleep's your reason for making copies, stay with Disk Manager.

If getting the stuff copied is your reason for making copies, try Floppy Copy (\$24.95 plus \$2 S&H from Softspot, P.O. Box 8786, Silver Spring, MD; or from mail-order houses for maybe less.

Floppy Copy copies quickly. With 2 drives it is capable of copying some disks completely in 24 seconds. It will copy ALL disks within 3 passes at 138 sectors a sweep.

FC can be loaded with any of the following: Mini-Memory, Extended BASIC, or Editor/Assembler modules. (M-M & E/A load 24 seconds; XB takes 50 seconds.)

If all FC could do is make backup copies super fast, it would be worth the price. But that is not all it can do.

It catalogs: displaying the complete status of the disk, including disk name and file index.

It validates the duplicating process. Error messages will appear if the disk is not transferring data.

It initializes. And it does that faster than Disk Manager, too.

It has REDD functions (great for initializing or duplicating a number of disks).

It has an immediate menu. Ease of back-and-forthing among the functions is just right.

Within the larger functions, there are other menus: you can take just files for a quicker copy, or you can copy the entire disk (including the blank sectors), for an example.

You can copy SSSD or any other configuration established by the DM or by CorComp.

So far I've found that FC will copy everything but itself.

FC comes with a single page of documentation. But you can choose some rather lengthy on-screen directions if you so choose. Once you've gone through the four sets, you really don't need to bother again. FC operates smoothly and with great ease.

Even I got it working within a minute or two.

One night I initialized 22 blank disks for a club swap. I stopped for a beer and a telephone call and returned to the task of copying the club's 22 disks. I did it with dread, in spite of the fact I really wanted to give Floppy a real test.

I did all 22 disks before I went to bed. A piece of cake, thanks to Floppy Copy.

The next night I did all kinds of things to check out its default system. I put in initialized disks. Floppy told me and gave its name.

I put in damaged disks. Floppy gave me an error message. I put in disks with only 5 sectors used. Floppy read in 1 second, wrote in 2.

FC always behaved and always protected me (and my disks) from myself.

I'd highly recommend this fast, versatile tool to anyone with a disk system.

Then a week or so passed and some disks I ordered (and some unexpected birthday disks) arrived at our post office on the same day.

The disks I had bought contained DM1000 and MASSCOPY. The gift contained ULTRACOPY and DMIII.

I tried ULTRACOPY and loved it. It was about as fast as Floppy but lacked some of Floppy's features. MASSCOPY was about as fast as Floppy and had a pile more features (getting closer to the DM cart).

Then I loaded DMIII (which came from an Apple-owner friend who felt sorry because I had such an inferior machine). Pitiful creature, this friend. But, still, I appreciated the wonderful gift. And I thought it WAS wonderful! "Imagine, a resident DM, just like the big computers," my friend remarked.

It was, too. A resident DM to draw from whenever my heart desired. With a few simple CALLS I could now Catalog or Copy or Rename or Printout or many of the other activities permitted only through the cart.

For the next week I played with DMIII and MASSCOPY and ULTRACOPY and with a lot of the other programs I bought and the other (mostly utility) disks I got as gifts.

I noticed, but I didn't "have time" (read inclination) to download DM1000 docs. I'd get to them and the program when I could to see what I had purchased, but it sure couldn't beat Floppy and ULTRA and MASS and, particularly, DMIII.

How wrong I was!

DM1000 (now up to version 3.3 and distributed Fairwarely and as part of FUNLWRITER and freely through SENIAL TRAVELER and FUNLPLUS! and, hopefully, your local user group) is stupendous! As a disk-manager environment all by itself it has no equal among any home computers. It simply makes the TI an immense tool.

Developed by Bruce Caron for the Ottawa Users Group, DM1000 lets you move, copy, rename, protect, unprotect files; lets you rename, copy, catalog, print contents, initialize disks; lets you read ALL DIS/80 files on screen; lets you initialize in box format (but also automatically initialize before copying disks); lets you change screen colors; lets you...

But why don't you get DM1000 from your group, unload the docs, read them over WHILE running the program, and amaze yourself and flabergast your Apple-owning friends. I did just that and it was great fun. (And don't forget to send a check to the Ottawa Users' Group!)

A few months later I got FUNLWRITER (which has an update DM1000 built in) and couldn't wait to show that same friend what my "inferior machine" could do. An even bigger flabergast resulted.

Elaine, my wife, says I am a truly evil person to be so heartless to my Apple friend. My four kids concur. So do I.

Ain't LIFE grand!?!

SWAP SHOP

TI ACOUSTICAL MODEM AND CABLES.....\$35.00

ASSORTED CARTRIDGES--WILL NEGOTIATE!!!

See George Von Seth, 292-2035

Wanted: Standard Parallel Printer Cable

Bob Carmany 855-1538

This space will be available in each issue of Guilford 99'er Newsletter. If you have anything to swap, sell, or need anything for your TI, contact George Von Seth (phone number above).