PUGET SOUND 99ERS
PO BOX 6073
LYNNWOOD, WA 98036

N

EDMONTON USER'S GROUP
P.O. BOX 11983
EDMONTON, CANADA

T5J-3L1

FEBRUARY 85 Vol. 4 No. 2

OFFICERS

PRESIDENT: JOHN MUSSELMAN	365-4745	VICE PRESIDENT: CHUCK WYNNE	745-3249
SECRETARY: KEITH MIFFLIN	821-1047	TREASURER: RON STONE	481-1412
NEWSLETTER EDITOR: RALPH DEVIN	783-2807	LIBRARIAN: JOHN UELAND	778-5273

****

NEXT MEETING

DATE: THURSDAY, February 21, 1985 TIME: 7:00 PM
PLACE: Bellevue Public Library, 11501 Main St, Bellevue

PROGRAM:
* Update on BBS System
* Navarone Database module demo
* Hardware/Software tips - Chuck Wynne
* TI-Writer Module Eliminator demo
* SuperSketch(w/Print) demo
* General discussion session

****

MARCH MEETING

DATE: THURSDAY, March 21, 1985
TIME: 7:00 PM
PLACE: Snohomish County P.U.D. Building
21018 Pacific Hwy S., Lynnwood

APRIL MEETING

DATE: April 25, 1985
TIME: 7:00 PM
PLACE: Bellevue Public Library
11501 Main St., Bellevue

****

JANUARY MINUTES

Over 70 people showed up for the first meeting of the year. The new officers were introduced by Ralph Devin, past president, and the meeting was turned over to the new president, John Musselman. He had each officer make a brief statement of the status of their respective departments. An introduction of the TI dealers that were at the meeting was made, with the dealers telling the group a few of the new things that were available.

Getting into the demonstration part of the meeting, we had Jim Williams show us two utility programs, Disk Mapper and Memory Manipulator. These are two additions to the expanding group of fine utilities available for the TI-99/4A. Ralph Devin then began a demo of the new Data Base Management module from Navarone. He will

continue with the demo at the upcoming meeting.

The main program consisted of a demonstration of the bulletin board system that the club is going to instalT. We had two computers connected together, RS-232 to RS-232, via a null-modem. This allowed people to see what happened on both ends of the system. The meeting was then adjourned to general discussion, with anyone wanting to talk to the BBS having their chance.

****

BULLETIN BOARD

At the February officers' meeting it was decided to go ahead and set up the BBS on a trial basis. We are not sure of the final format we will have, but will start with the program we have now, just to see how it works. Those who call will be able to leave comments about the system in a special section. The system will be operational on February 22, 1985. The system hours will be evening only until the system is deemed fully operational. At that time longer hours of operation will be implemented. The telephone number will be 784-4142.

****

SOFTWARE REVIEW

TI-WRITER -- Ron Stone

This month I purchased a utility program which has been very useful to me. The utility allows you to operate your TI-WRITER programs using either the Extended Basic or the Editor Assembler module plugged in. There are four files included on the diskette, these are 1)CHARA1, 2)WRTR, 3)XBWRTR, and 4) LOAD/WRTR.

The utility does not include the copyrighted TI-WRITER files and is intended only as a convenience in loading without the normal module juggling that we are all used to.

The utility will allow you to perform all functions within TI-WRITER with one exception, which is Show Directory which is built into the module itself and not the software.

The use of the utility is simple with either the Extended Basic or the Editor/Assembler modules. To load with Editor/Assembler module simply use option 3 (Load and Run) and type DSK1.WRTR when prompted for a File Name. Then press <Enter> when again prompted for another File Name. Type DSK1.FORMAT or DSK1.EDITOR depending which TI-WRITER function desired when prompted for a program name. The utility will then load the TI-WRITER files and you will be ready to go. What can be easier? Well, after using the Editor/Assembler module all week I decided to use the Extended Basic version and found it to be a joy. If you change the LOAD/WRTR name to LOAD, then boot up Extended Basic with this disk in drive 1 your procedures are as follows:

After you press option 2 for Extended basic you will see the following screen; you simply make a single key stroke with your selection.

Press : 1 - TO LOAD EDITOR 2 - TO LOAD FORMATTER 3 - TO LOAD UTILITY

I find the utility to be very useful especially when I have downloaded text from a bulletin board and want an immediate printout, or just to save my module port from excessive wear and tear.

The diskette also came with some surprises including a "real time clock", a file of CALL LOAD statements, DISK INFORMATION, and some other items which will remain a surprise until the meeting.

Since I have found this utility to be useful to me, I have made arrangements with the author, Tom Knight, to

sell copies at our meetings. If anyone is interested contact Ron Stone, Treasurer.

****

CLUB LIBRARY

It has been a while since we reminded everyone of the large number of programs available through our club library. We have included the following list of program packages as a brief overview. In the coming months we will give detailed listings of each grouping. We are planning on providing every member with a detailed catalog of the entire library.

BASIC GAMES:	Package #1-3	6 Programs	
	Package #4	8 Programs	
	Package #5	9 Programs	
	Package #6-8	7 Programs	
EXTENDED BASIC GAMES:	Package #1,2,5	6 Programs	
	Package #3-4	7 Programs	
	Package #6-7	10 Programs	
BASIC UTILITIES:	Package #1	13 Programs	
EXT-BASIC UTILITIES:	Package #1	19 Programs	
BASIC EDUCATION:	Package #1	10 Programs	
EXT-BASIC EDUCATION:	Package #1	11 Programs	
BASIC MUSIC:	Package #1	7 Programs	
EXT-BASIC MUSIC:	Package #1	7 Programs	(Also labeled TI-PUBLIC #1)
	Package #2	5 Programs	
TI-PUBLIC:	Package #2	11 Programs	
	Package #3	7 Programs	
TI-FORTH:	Complete Forth System		
BASIC DEMOS:	Package #1	11 Programs	

This list represents over 200 programs. This is by no means the entire library, only the part that has been cataloged and tested to try to insure that the programs are bug-free and do what they should do. There has been a need for volunteers to help catalog the library so that more programs can be made available to the members.

Members are encouraged to contribute programs to the library. We have a policy that gives a person one of the above packages for every new program we receive that passes the inspection process.

We have just received a version of Forth that will load with Extended-Basic. This should make Forth programming available to those without the Editor/Assembler module.

****

CLUB DIRECTORY

There have been quite a few requests from members that the club publish a directory of its members for the use of the membership. We would like to do this as this would allow members to find other members who live in their neighborhood, or other members who share the same interests, etc. To this end, we would like to ask all members who DO NOT want their names in the directory to respond in writing by the end of February. If we do not receive a letter from you stating to the contrary your name will be included.

****

WANTED: LARGE TV

There have been a lot of complaints about the immensity of screen size represented by the monitors at the

monthly meetings. We would like anyone who has a 19-inch or larger(?) color TV that they could bring to the meetings, or loan to the club for use at the meetings, to contact Ralph Devin at 783-2897. A larger screen might just bring some of those demos to life!!

****

TALK TO A TI REP

At the upcoming meeting we will be visited by a representative from the TI Service Center. She will be available to answer specific questions regarding repair and exchange policies and prices.

****

WANTED

Anyone who has an RS-232 card, P-Code card, and/or Printer for sale, contact Ed Elliot by writing 2631 Soder Hill Rd, Lake Stevens, WA 98285 or call collect 334-7327.

****

FOR SALE

Completely expanded system (Disk, 32K, RS-232) with Speech Synthesizer, Editor/Assembler, Logo II, Extended Basic, Terminal Emulator II, all manuals and additional handbooks, and additional programs and material. Call Ron Leonard at 774-9722 if interested.

****

STOP

Check the prices at the retail stores then call

McDONALD COMPUTERS

Lets make a deal.

For Example: A Panasonic 1595 printer -- $238.

Call McDonald Computers at 525-1981.

****

PROGRAMMING by Tom Wynne

Extended BASIC Bug - The ACCEPT statement

1. If you list a program, enter ACCEPT A$, type one character and then hit 'ENTER'. Your computer screen will show different patterns and colors. Pretty, but you have to reset your computer and your program will be gone.

2. You can change the screen color and other things using the ACCEPT statement. First type in the commands then at the end of your command type ACCEPT A$. For example: type in this line, then press 'ENTER':

```
CALL SCREEN(4)::CALL SPRITE(#1,42,2,1,1,20,20)::ACCEPT A$
```

Now press FCTN 4 (CLEAR). You should have a green screen and an asterisk moving diagonally across the screen. It will continue until you get an error or you run the program. Now you can list or edit your program and the screen and sprite remain unchanged!

FORTH VS. BASIC - Tom Wynne

Just about every computer that s been created in the past years nas had the BASIC language either built in or supplied on a disk. BASIC is one of the easiest languages to learn and to understand. It has become one of the most popular languages for programming. You ask any programmer about BASIC and he will either have programmed in it or have an idea of what it is. That is why BASIC has been the most popular of most languages. Even though BASIC is a popular language, it has its drawbacks. These drawbacks may turn a programmer to another language. One may require that the language have speed and the ability to be tailored to one's needs and desires. These are qualities of "low-level" languages in that the command processes can be tailored. One wanting to convert to a lower level language will find difficulty in learning and understanding all of the aspects of that language. BASIC is a "high-level" language in that it is closer to the human's understanding. A "low-level" language is closer to the computer's understanding.

FORTH is a language that has been improperly classified as an intermediate language in that it has human words but "low-level" complexity. This is a reason one may choose to switch to FORTH. It may have the same commands as BASIC and the same applications but in a different format. An example can be the command "CLS" in BASIC. This command is used to clear the screen. In FORTH, that same "CLS" statement will exist and it will do the same function as in BASIC. In other "low-level" languages, there will not be a "CLS" command but a complex routine that the programmer must create to do the same application as in FORTH and BASIC. FORTH is a "human" language with "low-level" complexity.

FORTH differs from BASIC in that FORTH, one can directly keep track of numbers and manipulate them in more ways than BASIC. FORTH works with the stack in all of its operations. This requires that FORTH only uses zero operand instructions. When executing a command, FORTH automatically assumes that any parameters needed for that command are on the stack. In BASIC one must include any parameters in the command by either directly applying a number or a variable. In FORTH, manipulating numbers in formulas can be more difficult. Humans are used to Infix notation (2 + 3). This notation is used in BASIC for it is easier to understand. FORTH uses Reverse Polish Notation (2 3 +). This notation can be more confusing to some and to more of an advantage to others. This notation does not require the computer to search the formula and jump around to be computed. All it has to do is go from left to right through the formula and come out with an answer.

In some instances, one may wish to convert to a different number system for calculation. In BASIC this is not possible unless a subprogram is built to accomplish this. (In some computers BASIC will have a base conversion from decimal to hexidecimal.) In FORTH, one can convert to any number system very easily. One can convert from decimal to octal to binary with a few keystrokes.

FORTH uses an area in memory called the "parameter stack" for number storage. This type of stack is a last in, first out stack. If a number is encountered, it is pushed onto the top of the stack. A push to the stack will put the number in the memory location indicated by the stack pointer and then increment the stack pointer by one to the next location where a number will be placed.

BASIC uses variables to store numbers. It will take the variable name, compile it with an address and then write the value of the variable into that address. FORTH also can deal with variables as a storage of numbers and they work in the same way as BASIC variables do. In BASIC, to print the value of a variable "A" you would type: "PRINT A" and the value would be returned. In FORTH the same process would be achieved in typing: "A @ ." ("A" being the variable, "@" fetch the address, get the value, and push the value on the stack, and "." print the top of the stack) This would return the value of the variable A. Constants in FORTH work the same way - to hold values that will not change.

Arrays work very much the same way in FORTH as in BASIC. At the beginning of each program, one must define the number of cells to set aside in the array. In FORTH, arrays are just memory locations set aside for values. In BASIC, the programmer must not worry about the memory locations.

Strings are alot easier to manipulate in BASIC because they can be stored into a variable just like a number. (A$="HELLO") In FORTH strings are stored in a particular address. When they are stored in an address, the one must remember the address so one knows the memory location where the string is stored. The advantage of this is that pieces of the string may be directly accessed from memory.

FORTH is a language built on nothing but subroutines. Most of the routines are programmed in FORTH language to begin with so they can be changed to ones desire. The commands used in BASIC are permanent. They are subroutines which cannot be changed or linked together. FORTH subroutines can be changed and linked together to make a more powerful programming language. A BASIC program runs from top to bottom while FORTH is the reverse. A definition must be created (compiled) before it can be called. Once it is created, then it can be called by another subroutine and so on. All subroutines are called colon-definitions. A definition is not saved in printable source format except on a storage device.

The colon is a FORTH word that tells the computer that the word following is a new subroutine containing the code following until a simicolon is encountered. When a definition is created, it is placed in the Dictionary along with all other subroutines. In BASIC, a subroutine is called by a GOSUB statement and it terminates where a RETURN statement is encountered. In BASIC, this does not add to a dictionary. It is just referenced as a subroutine in the program only during execution. Once a definition is created in FORTH, it becomes part of the language. It can be called at any time whether or not the program is running.

If a program is saved in BASIC, it is given a file name and is automatically placed on the storage device. FORTH does not work with file names for programming. FORTH only works with 1024 byte screens and each screen has a number. Each screen is known as a block. In BASIC, if one wishes to execute a program, he would reference the file name to load, then run the program. In FORTH, one must know the starting screen number of his program, LOAD it, then type the definition name he has created to execute the program.

With FORTH, one may directly access the disk and obtain only pieces of data without referring to a record number. He would just reference the screen number and the number of bytes to retrieve. This is called a primitive disk read where no file name is referenced, just raw data is pulled off of the disk.

FORTH can also usp file names for sequential and random access files. These types of files work very much the same as BASIC.

BASIC works with nothing but line numbers as a reference to command lines. To edit a line, one references the line number and that line can be edited and compiled automatically in memory. In FORTH, only the screen to be edited and two other screens are in memory at the same time. The editor allows one to enter text and change the contents of the screen. The FLUSH command will dump the contents of all updated screens in memory to disk. In BASIC, the program is all in memory at one time, but in FORTH only parts are in memory until it is compiled. FORTH depends on the disk for programming source code. This allows more free memory space in the computer.

Using the FORTH editor has its disatvantages. On some systems, the editor is hard to operate and sometimes laborious to input data. Also when your program is on the disk in source format, it may take a long time for it to be compiled before it is able to be executed. One can save a program in compiled format on disk, but it would be almost impossible to read or change.

Many programmers who have programmed in BASIC have found that it is slow. FORTH is one of the fastest languages. Some people say that FORTH is the second fastest language next to ASSEMBLY language and even FORTH can even be faster than a poorly built program in ASSEMBLY.

FORTH has many advantages. For one, FORTH compiled code is very compact. FORTH applications require less memory than thier equivalent ASSEMBLY language programs. This enables FORTH to be run on small computers with a small amount of memory. Also FORTH is a transportable language. It has been used on just about every mini-computer in the industry. This is because of its little memory requirements. BASIC also is implimented on the same amount of computers, but BASIC requires more memory and usually takes up the majority of the available memory.

FORTH has some disadvantages. It has very compressed code so that once a program is compiled into memory, it cannot be read. If one is to make changes, he must change it on the disk and then compile it back into memory. BASIC supports readable text for one can list his program and change anything he wants without having to access the program on disk.

FORTH has been implimented for many uses. FORTH is used to operate robots for business and motion pictures. FORTH has also been used in medical applications. On a single PDP-11 at a major hospital, "FORTH simultaneously maintains a large patient database; manages thirty-two terminals and an optical reader; performs statistical analysis on the database to correlate physical types, diseases, treatments, and results; and analyzes blood samples and monitors heartbeats in real time." - Quoted from STARTING FORTH by Leo Brodie. FORTH was also used to control the graphics in the movie "TRON" from Walt Disney.

FORTH is a powerful language for many applications and for the programmer. If one is looking for a language that has speed, complexity, and expandability FORTH is the language to look into.

* Printed by the Text Columnizer Utility by Tom Wynne