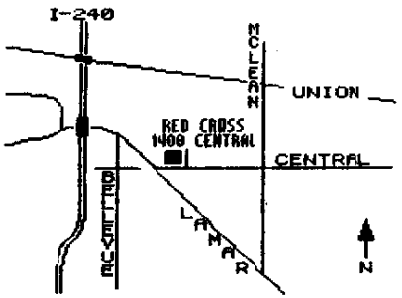
NOTICES

MEETING
7:00 P.M.
Thursday, AUG 15 th
Red Cross Building
1400 Central Ave.

WORKSHOP
To Be Announced



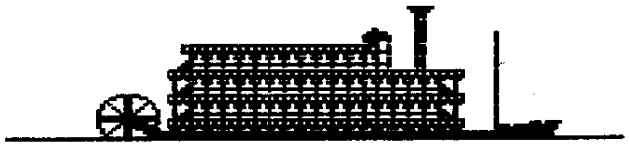
Mid-South 99 Users Group
P. O. Box 31522
Germantown, TN 38183-0522

UG 2/86
DALLAS TI USER GROUP
P.O. BOX 29863
DALLAS, TX 75229

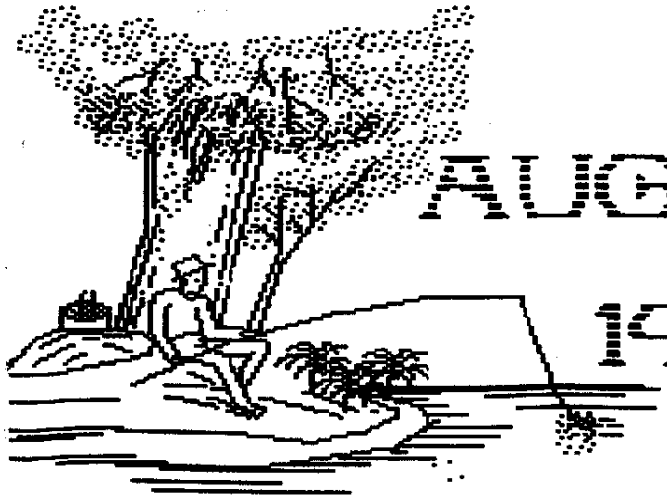
FIRST CLASS MAIL

TIDBITS

MID SOUTH 99 USERS GROUP



MEMPHIS TENNESSEE



AUGUST 1991

Newsletter of the MID-SOUTH 99 USERS GROUP-Vol 9, #8-AUG 1991

TIDBITS

OFFICERS

Gary Cox	PRESIDENT	901-358-0667
Richard Miller	VICE-PRESIDENT	901-794-9945
Beery Miller	SECRETARY	901-368-1169
Bob Jones	TREASURER	901-363-9213
Jim Saemanes	Technical Support	901-476-7011
Jim Saemanes	Disk Librarian	901-476-7011
Pierre Lamontagne	CO-Librarian	901-386-1513
Gary Cox	Program Chairman	901-358-0667
Mac Swope	Chairman - Equipment	901-363-3880
Marshal Ellis	Editor - TIDBITS Newsletter	901-327-2506
Marshal Ellis	Editor-Technical Interface	901-327-2506
Beery Miller	9640 NEWS BBS Sysop	901-368-0112

JULY 1991 INDEX

PRESIDENT'S BIT

----------By Gary W. COX

The Memphis computer fair is rapidly approaching September 28th at State Technical Institute. This fair will represent most of the local computer groups and our group will have a room to ourself as in previous fairs. More details elsewhere in this newsletter. Beery Miller of 9640 news shoud be present demonstrating his products... Any companies wishing to have their catalogs or flyers handed out at the fair (TI related items) please send them to my home address at 3174 Melbourne, Memphis, TN 38127.

Beery Miller will be starting an assembly language programming class at this months meeting. The class will be before the regular meetings and will start at 6:30pm in the cafeteria at the Red Cross. All interested persons in attending the assembly langage programming class should arrive by 6:30pm.

Volunteers are needed to host Saturday workshops. If you would like to have a workshop at your house please contact me or announce it at the meeting...

C ya at the meeting...

IN THE NEWS

---------- by Gary W. Cox

SUBJECT: COMPUTER FAIR AT STATE TECHNICAL INSTITUTE

CONTACT:
Gary Cox, vice-president HUG - (901) 358-0667 (after 5pm)
Ken Akins, president HUG - (901) 382-5541 (daytime)

Memphis, Tennessee, July 15, 1991 -- The Home Computer Users Group (HUG) in conjunction with State Technical Institute will be hosting a NON-PROFIT Computer Fair from 10am to 5pm Saturday September 28th, 1991 in the Fulton Building at State Technical Institute located at 5983 Macon Road in Memphis, Tennessee.

A consortium of computer user groups in the Memphis area will be participating in the event demonstrating some of the latest in computer software and hardware. User Group representatives will be available to answer most computer related questions. Guest speakers are scheduled to give presentations on special topics such as desktop publishing, computers in business, computerized investing, word processing and computer produced music. On going presentations will be held throughout the day on a variety of topics.

Among the computer user groups scheduled to be represented are the Home Computer Users Group (HUG), The Memphis PC Users Group Inc., Apple Core of Memphis, Atari Systems Hobbyists, Participating Online Systems Serving the Mid-south (POSSM), Memphis Amiga Users Group, Memphis Commodore Users Group, Mid-South TI99/4a Users Group and the Memphis Color Computer Users Group.

The event is FREE and open to the public and a drawing for prizes will be held. This fair will be educational as well as entertaining. The above mentioned computer user groups as well as State Tech are both non-profit organizations.

For more information write:

Home Computer Users Group (HUG)
5541 Pleasant View #2
Memphis, TN 38134

The following edited from the June and July 1991 Micropendium:

The 9918Ø5 accelerator for the TI99/4a manufactured by Bud Mills Services was scheduled to be available to end users last month, according to Bud Mills. The device is also available from OPA.

The card, prices at $25Ø, dramatically increases the power of the TI. The standard TI uses a 9ØØØ chip with a clock speed of 3.3 Megahertz. The 9918Ø5 upgrade operates at a clock speed of 12 mhz. The upgrade is installed in the TI console and requires no soldering. According to its designer, Don O'Neil, the 9918Ø5 accelerator increases the speed of the TI by a factor of 5. The improvement increases to a 1Ø fold gain in speed when used with a 16 bit RAM card that fits in the Peripheral Expansion Box. According to O'Neil, the accelerator is invisible to the TI system. The PEB RAM interface card is expected to be available in late October or early November. This card is prices at $9Ø and will serve as a functional replacement for the TI PEB cable. It features battery backed static RAM DSRs (Device Service Routines), for easy upgrading, a smaller, 1 1/2 inch cable connection between the PEB and TI console, uses a 16 bit data bus, has 8 SIMM slots for up to 8 megabytes of RAM expansion, 32k static RAM with zero wait state operation built in (this replaces existing in console 32k RAM expansions) and 1 "processor direct slot" for future expansion.

The Ottawa TI99/4a Users Group is now offering Volume 4 of Lucie Dorais' Fast Extended BASIC Tutorials and Programs. This disk contains 1Ø new programs among which are the Balldrop game, a utility to rotate TI Artist Instanced, drawing programs t emulate abstract painters Mondrian and Vasarely, a study of visual perception and a French Christmas carol. The disk is available from the Librarian, Ottawa TI99/4a Users Group, 3489 Paul Anka Dr., Ottawa, Ontario, Canada K1V 9K6 for $2 plus

postage. Dorais Vols. 1-3 are also available for the same price.

Texaments has released the TI Artist Plus! Pak, a combination package of five programs for TI Artist and TI Artist Plus! users. As well, Texaments now buys, sells and trades used TI99/4a and Geneve 964Ø hardware, software, resource materials and accessories directly with end users, according to Steve Lamberti, president of Texaments.

The TI Artist Plus! Pak consists of five software packages, GuideLines, Display Master, Artoons, Designer Labels and Artist Companion of choice (Nos. 2 through 13) at $49.95. Purchased separately price would be $6Ø.

Texaments also offers a free, no obligation quotation service to persons wishing to sell all or part of their current TI99/4a or Geneve 964Ø system. To take advantage of this service individuals should mail in a listing of equipment that they wish to sell. Individuals who wish to obtain a free listing of used TI hardware, software and resource materials should send a self addressed, postage paid envelope to Texaments.

Also new from Texaments are Fonts, Frames and Fun, a 3 disk package of 11 fonts, 39 frame borders and 22 instances for use with TI Artist and TI Artist Plus, the price is $12.95. Also released is The Missing Link Font Pak consisting of 29 display fonts for use with The Milling Link. The price is $7.95.

Prices have been reduced on CSGD software as follows: CSGD I - $9.95, CSGD II (The Banner Maker) - $7.95, CSGD III - $12.95 and CSGD User Disk 1 -$2.95, CSGD User Disks 2 through 7 for $5.95 each, CSGD Cataloger -$3.95... For further information or to order contact Texaments at 53 Center Street, Patchogue, NY 11772 or call (516) 475- 384Ø or (516) 475-6463 (BBS). Mail orders should include an additional $3.25 for U.S. shipping.

A newsletter aimed at TI Base users has folded. TI BASE User is no longer being published, according to its publisher, Bill Gaskill. The newsletter ended its run with Volume 2 Number 4. Gaskill says that all subscribers have been refunded and that the newsletter is no longer available. Gaskill added that all products by PRK DataBasics and Junction Softworks are no longer available.

The TI Fest West 92 will be held February 15-16 in Phoenix, Arizona. The sponsoring group is the Valley of the Sun TI users Group (VAST). The site will be at the Days Inn Phoenix/Camelback, 1-8ØØ-688-2Ø21. The fair will be going on from 9am to 5pm Saturday and 9am to 3pm Sunday. Among the door prizes to be presented is a color printer. Tickets to the event are $5 for both days. For more informaiton contact the VAST BBS at (6Ø2) 233-8Ø79 or call K. Knight at (6Ø2) 938-5446 or R. Rees at (6Ø2) 869-8145.

A new Finnelweb is in the works. Tony McGovern is working on V4.4 of the program. The current version is 4.32. Reportedly, McGovern is rewriting the editor from scratch for V4.4. Also, the Disk Review segment of the program may be able to view GIF pictures as well a being expanded into a complete disk manager. The editor may also incorporate the formatter, instead of being handled as a separate function as is currently the case. A print preview function may also be added.

Want to buy used software and hardware but don't know where to find it? Want to sell your unused computer stuff but don't want to spend a lot of money doing it? If you answered yes to both of those questions then the National Hardware Club may be for you. Buy used computer hardware and software to sell your unneeded computer hardware and software (Apple, Atari, Commodore, Macintosh, PC, TI etc). "We will ace as middlemen between buyer/seller and insure honest transactions. We guarantee that your buys and sells will be done to your satisfaction." Annual membership (includes newsletter) is $15 a year. Use Visa/MC and call 800-777-6632 to expedite your membership (call 9-5 central time). Or send $10 self addressed stamped envelope to National Used Software/Hardware Club (NUSHC, P.O. Box 1343, Round Rock, TX 78680 for FREE no obligation pack. (U.S. orders only)

That's the news for this month...

THE ALTMAN FAIRWARE LIST

FROM THE PAGES OF THE SOUTHWEST NINETY-NINERS/JULY,'91

The Altman Fairware List is being maintained and updated by the SouthWest 99ers. The latest list includes over 400 titles, and indexes for author, and subject. It is available for $2 (posteage inc.). You can choose the recieve it, either pre-printed or on disk. The SSSD unarchives to 591 sectors, and prints out to over 30 pages . Send a check or maney order to SouthWest Ninety-Niners, P.O. Box 17831, Tucson, Az. 85730.

Announcements of new fiarwre may be sent to the above address. Please include a short description of the program(s) and the author's address. The information provided will be added to The Altman Fairware List and made available to other TI/9640 users.

TELECOMMUNICATIONS

----------by Richard Lumpkin
Houston User Group
August, 1990

The following article is compiled from source: 3. Fred and Amy Mackey, Pittsburg U/G, Feb 87, series.
Note: Editors comments/additions/changes are in [.....]

Fred and Amy Macky :

GETTING ON LINE: AN INTRODUCTION TO TELECOMMUNICATIONS:

There are more things you need to know before you can transfer data between computers - the softwre or telecommunications program designed especially for your computer. Theis program directs your computer on jow to use the modem and how to transfer information between the two computers. For the TI-99/4A, most telecom programs will require 32k memory and wither the E/A or X-B modules.

Any two computers using compatible programs c an communicate. Compatability occurs by setting the "configuration parameters" of your telecom program to match that of the computer you are calling, or the "host" computer. When you run your ... program, it wioll ask you to set most or all of the following. (Some programs will automatically set them for you [or] you will be given the options to change them according to your needs:

1. BAUD RATE - 110, 300, 600, 1200, 2400 -- This is the number of bits per second (not bytes) that can be sent or recieved. The setting for this parameter depends of the modem and BBS compatability, usually 300, 1200, 2400.

2. DATA BITS - 7 or 8 -- This is the number of data bits you are going ot send for each character (byte) being sent. (TE-II uses 7 bits/byte, X-Modem uses 8 bits).

3. PARITY - Odd, Even, None -- PArity os a simple means of detecting errors which might occur during data character transmission. It is only in effect during 7-bit operations -- 'None' is the setting for 8-bit operations.

4. SERIAL PORT OF MODEM - 1 or 2 -- This number specifies which "port" or plug of the RS-232 the modem is commected to.

5. PRINTER DEVICE NAME - PIO or RS-232 -- DEvice name of the printer you are using. You do not necessarily need a printer to run a telecom package.

6. SCREEN WIDTH - 40 or 80 -- [columns in display]. The screen width used with the TI is 40. Some oackages will l;et you use a lesser number to compensate for monitor or TV screen differences in display size and clarity.

7. DUPLEX - Full or HAlf -- This controls the source of the characters which appear on your screen. half duplex assumes 'one way' transmission (no echo of characters sent) and therefore local display of characters being sent is used. Full duplex assumes constant 'two way' transmission and 'echo' of all characters back to the sending terminal , and it is this 'echoed' character which is displayed, thus constantly indicating the quality of both directions in the link.

If your communications attempts get only garbage, or partial garbage, the parameters are porobably incorrect. (Ed. NOTE: One further note: the"originating" modem does not originate the carrier tone. The "answering" modem answers with a carrier tone, which the "originating" modem then responds to with a different carrier tone. 3Ø0 baud modems use tones which remain steady when the link is not transmitting, while 12 bps and higher modems sound like scratchy or noisy tones, with no distinguishable difference when active or idle.)

COMPUTING - THE FUTURE?

----------19??

This article was written by Gary Christensen, and first appeared in the November issue of Bug-Bytes, the newsletter of the Brisbane Users Group.

No discussion about home and personal computers would be possible without introducing the Big Blue, IBM. When TI was pressing the standard of the home computer upward, IBM decided to get in on the small computer with one that they called a personal computer. I personally do not know a lot about these machines and to be honest, I'm not that interested but it really came down to a battle between the Home Coputer and the Personal Computer. This battle was fought, not on the ability of the computer but in the marketplaces. It perhaps is history now who won that battle.

The majority of the home computers did not make it. The only remaining ones are the Macintosh and the Amiga. Once the marketing had thinned the market out a little the companies could start concentrating on who had the best machine and makng sure that it was theirs. The drive for computing power and performance was on.

It took only a couple of years for IBM to progress from the 8Ø88 processor to the faster 8Ø286, then another short peroid to the release of the 8Ø386 and recently the 8Ø486. Each are faster than its predecessor and have greater capabilities. Today it has reached the stage that a laptop computer with a 8Ø486 processor has more memory as is faster than the big mainframes of a decade ago. It also raises the point - who needs that much power in their lap?

A PC with an 8Ø286 will does everything that the normal person or business will ever need. Let'sface it, the TI will do just about everything that the average householder will ever need. Take an 8Ø386 to link the 8Ø286 computers together and you have a network that will run a large business, even one that is spread throughout Australia. The 8Ø486 is great if you are working with enormous amounts of data, say in graphics manipulation, or where high speed data processing is necessary.

To my way of thinking we have approached the end of the track. Computer response times have greatly exceeded the human need. Who cares if you have to wait 1/1ØØth of a second for a computer to find the data, assess the users requirements and respond accordingly? I'll save the money and use the old one that takes 1/25th of a second. Could you tell the difference?

Sure there are needs for better systems but there is no longer the bulk market. Specialized needs and big dollars - would you spend $15,ØØØ on an 8Ø486 based computer when a $3ØØØ

8Ø286 will do the job? I believe that the areas of development will be in memory, data storage and periherals.

Memory first. The amount of data that can be stored on one chip is increasing all the time. Will we soon see a chip that can hold a 1 megabyte? How about 4 or 16?I think that it is only a matter of time. The other area of development is non-volitile memory. Chips that do not lose the contents when the powerís disconnected are available, but the systems are not completely satisfactory. Eproms have to be burnt and Eprams need special circuitry. Commercial development of non-volitile memory is not far away.

How about data storage? Floppy drives give up to 1.4meg storage. Higher densities may become available but there is a limit inherent in the media used to store the data. Magnetic imprints can only be so close together without affecting each other. Laser seems to be the way. CD players already cost less than ahard drive. CD-ROMs are already used to store bulk infor- mation. Soon the process of writing as well as reading will be perfected and these units will become available. On a dollars- per-megabyte basis the hard drives will not be able to compete. Some manufacturers are already talking in gigabytes (1 gigabytes = 1ØØØ megabytes = 3ØØØ DSDD disks). The time to find data on the laser disk will also be very small, no heads to move - just a beam of light!

The other area of improvment is n printers. It's great to be able to put the information into the computer but it still has to be retrieved. There are a couple of areas of work in printers. The dot matrix printer has improved with the 24 pin head. This gives high quality without much cost. The next step up is the ink jet printer. This printer shoots a small droplet of ink onto the page instead of using a pin and ribbon. They have up to 48 nozzels on the head. The final type is the laser printer. These use a laser beam to create an image on what is essentially a photocopier. These are fast and have high resolution, although the ink jet printers can give them a good run for their money in the resolution stakes.

In all these types of printers we will see the price continue to fall. I don't know where it will stop, but I believe that a laser printer will be within reach of most computer buyers in the near future. An added feature will be colour. The dot matrix printer wil lead the way here because of the ease of manufacturing colour ribbons. In fact, there are already quite a number of good colour printers available. The colour ink jet printers will come soon and may have the ability to create more colours than the dot matrix variety. The dot matrix are presently limited to 7 coours, although this may improve in the future. By mixing coloured ink, the ink jet printers may be able to generate an enourmous variety of colours. The top range will be the colour laser printers. Anyone who has seen a colour copier in action will understand what is possible, but it will be quite a while before the price comes down to to point where the individual can afford it.

Where does all this leave the TI? The bit about printers is obvious. As each new improvement is made, the software will be upgaded to support it. TI-Artist Plus already supports colour printers. As the hard disk technology improves, the cost will fall. Hard drives will shortly be available to more and mote TI'ers. The only problem here is the cost of the

conroller, but perhaps someone will do something about that too.

We can already put memory expansion on a single chip so, as the memory becomes cheaper, we may see true 512K memory expansion for the TI. This will of course include the development of the software to make use of the extra memory so we could expect some fantastic programs. All of TI-A+ could be loaded at once and the images could be much larger than the screen - perhaps a full page of high resoltion picture.

The extra memory also opens the way for other options. How about video image digitizers? Hook that to a fax board for sending and recieving faxes.

Better memory technology means better chips of other types as well. Perhaps new video chips. The 9958 seems promising but not really applicable to the TI computer. There is a lot of room for work here.

Well, there are a few of my ideas. Some may be right, most probably wrong, but I know that I am looking forward to the future with my TI-9640 and one day I will be able to read this to see how far out I was.

What are your thoughts? What do you think will happen in the future? What do you hope will happen? Why not let us all know and if enough speak out, we may even get what we want!

REAL PROGRAMMING

---------- author unknown

October 1990

part 2

Sydney Home Computer User Group

What kind of programming tools does a real programmer use? In theory, a real programmer could run his programs by keying them into the front panel of the computer. Back in the days when computers had front panels, this was actually done occassionally. Your typical real programmer knew the entire bootstraping loader by memory in hex, and toggled it in when ever his program destroyed the bootstrap. Back then, memory was memory - it did not go away when the power went off. Today memory either forgets things when you do not want it to, or remembers things long after they are best forgotten. Legend has it that Seymour Cray (who invented the Cray-1 supercomputer, and most of Control Data's computers) actualy toggled the first operating system systen for the CDC-7600 in on the front panel from memory. Seymour, needless to say, is a real programmer.

One of my favorite real programmers was a system programmer at Texas Instruments. One day, he got a long-distance call from a user whose system had crashed in the middle of saving some important work. Jim was able to repair the damage over the telephone, getting the user to toggle in disk I/O instructions at the front panel, repairing system tables in hex, reading register contents back over the telephone. The moral of the story : while a real programamer usually includes a keypunch and lineprinter in his toolkit, he can get along with just a front panel and a telephone in emergencies.

In some companies, text editing no longer consists of ten engineers standing in line to use a 029 keypunch. In fact, the building I work in does not contain a single keypunch. The real programmer in this situation has to work with a "text editor" program. Most systems supply several text editors to select from, and the real programmer must be careful to pick one that reflects his personal style. Many people believe that the best text editors in the world were written at Xerox Palo Alto research center for use on their Alto and Durado computers (3). Unfortunately, no real programmer would use a computer whose operating system is called SMALLTALK, and would certanly nevr talk to a computer with a mouse. Some of the concepts in the Xerox editors have been incorporated into editors running on more reasonable operating systems - EMACS and VI being the two. The problem with these editors is that real programmers consider "what you see is what you get" as just as bad a concept in text editing as it is in women. No, the real programmer wants a "you asked for it, you got it" text editor - complicated , cryptic, powerful, unforgiving and dangerous. TECO, to be precise. It has been observed that a TECO command sequence more closely resembles transmission line noise than readable text (4). One of the more entertaining games to play with TECO is to type your name in as a command line and try to guess what it does. Just about any possible typing error while talking with TECO will probably destroy your program , or even worse, introduce subtle and mysterious bugs in a once working subroutine. For this reason, real programmers are reluctant to actually edit a program that is close to working. They find it much easier instead to just patch the binary object code directly, using a wonderful program called Superzap (or its equivalent on non-IBM machines). This works so well that many working programs on IBM systens bear no relation to the original FORTRAN code. In many cases, the original source code is no longer available. When it comes to fix a program like this, no manager would ever think of sending anyone less than a real programmer to do the job - no quiche eating structured programmer would even know where to start. This is called "job security".

Here are some programming tools that real programmers do not use:

* FORTRAN preprocessors like MORTRAN and RATFOR. These are cuisinarts of programming - great for making quiche. See the comments above on structured programming.

* Source language debuggers. Real programmers can read core dumps.

* Compilers with arrays bounds checking. They stifle creativity, destroy most of the interesting use of the equivalence statement, and make it impossible to modify operating systems with negative subscripts. Worst of all, bounds checking is inefficient.

*Source code maintenance systems. A real programmer keeps the code locked up in a card file, because it implies that the owner cannot leave important programs unguarded (5).

THE REAL PROGRAMMER AT WORK

Where does the typical real programmer work? What kind of programs are worthy of the efforts of so talented an individual? You can be sure that no real programmer would be caught dead writing accounts-recievable programs in COBOL, or sorting mailing lists for People magazine. A real programmer wants tasks of earth shaking importance (literally!).

* Real programmers work for Los Alamos National Laboratory, writing atomic bomb simulations to run in Cray-1 computers.
* Real programmers work for the National Security Agency, decodeing Russian transmissions.
*It was largely due to the efforts of thousands of real programmers working for NASA that our boys got to the moon and back before the Russkies.
* Real programmers programmed the computers on the space shuttle.
* Real programmers are at work for Boeing, designing the operating systems for cruise missles.

Some of the most awsome real programmers work at the Jet Propulsion Laboratory in California. Many of them know the entire operating systems of the Pioneer and Voyager spacecraft by heart. With a combination of large ground-based Fortran programs and small space craft based asembly language programs, they are able to do incredible feats of navigation and imporvisation - hitting ten-kilometer wide windows at Saturn after six years in space, repairing or bypassing damaged sensor platforms, radios and batteries. Allegedly, one real programmer managed to tuck a pattern- matching program into a few hundred bytes of unused memory in a voyager spacecraft that searched for, located and photographed a new moon of Jupiter.

The current plan for the Galileo spacecraft is to use gravity- assisted trajectory past Mars on the way to the surface of Jupiter. This trajectory passes 80 +- 3 kilometers of the surface of Mars. Nobody is going to trust a Pascal program (or a Pascal programmer for that matter) for navigation of such tolerances. As you can tell, many of the real programmers work for the U.S. government - mainly defense department. This is as it should be. Recently, however, a black cloud has formed on the real programmers' horizon. It seems that some highly placed quiche eaters at the defense department decided that all defense programs should be written in some grand language called ADA ((c) DoD). For a while, it seemed that Ada was destined to become a language with structure, a language with data types, strong typing and semicolons. In short, a language designed to cripple the creativity of the typical real programmer.

Fourtunately, the language which the DoD adopted has enough interesting features to make it approachable - it is incredibly complex, includes methods for messing with the operating system and rearranging memory, and Edsger Dijkstra does not like it (6). Dijkstra, as I am sure you know, was the author of "the GO TO considered harmful" - a landmark work in programming methodology, applauded by Pascal programmers and quiche eaters alike. Besides, the determined real programmers

can write FORTRAN programs in any language. Real programmers might compromise their principles and work on something slightly more trivial than the destruction of life as we know it, providing there is money enough in it. There are several real programmers writing video games at Atari, for example (but not playing them - a real programmer knows how to beat the machine every time - no challenge in that). Everybody at Lucasfilmm is a real programmer (it would be crazy to turn down the money of fifty million Star Trek fans). The porportion of real programmers in computer graphics is somewhat lower than the norm, mainly because nobody has found a use for computer graphics yet. On the other hand, all computer graphics programming is done in FORTRAN, so there are a fair number of people doing graphics in order to avoid having to write COBAL programs.

REAL PROGRAMMERS AT PLAY

Generally, the real programmer plays the same way as he works - with computers. The real programmer is constantly amazed that his employer actually pays him for what he would be doing anyway (although he is careful not to express this opinion out loud). Ocassionally, a real programmer does step out of the office for a breath of fresh air and a beer or two. Some tips on recognizing real programmers away from the computer rooms:

* At a party, the real programmers are the ones in the corner talking operating system security and how to get around it.
* At a football game, the real programmer is the one comparing the plays against a simulation printed on 11 by 14 fanfold paper.
* At the beach, the real programmer is the one drawing flowcharts in the sand.
* At a funeral, the real programmer is the one saying " Poor George. And he almost had the sorting routine working before the coronary".
* In a grocery store, the real programmer is the one who insists on running the cans past the laser checkout scanner himself, because he never could trust keypunch operators to get it right the first time.

THE REAL PROGRAMMER'S NATURAL HABITAT

What sort of environment does the the real programmer function best in? This is an important question for the managers of real programmers. Considering the amount of money it costs to keep real programmers on the staff, it is best to put him or her in an environment where they actually get the work done. The typical real programmer lives in front of a computer terminal. Surrounding this terminal are:

* Listings of all the programs them real programmer has ever worked on, piled in roughly chronological order on every flat surface in the office.

* Some half-dozen or so partly filled cups of cold coffee. Occassionally there will be cigarette butts floating in the coffee. In some cases, the cups contain orange crush.

* Unless the real programmer is very good, there will be copies of the OS JCL manual and the principles of operation open at some particularly interesting pages.

* Taped to the wall is a line-printer Snoopy calendar for the year 1969.

* Strewn about the floor are several wrappers for peanut butter filled cheese bars - of the type that are made pre-stale at the bakery so that they cannot get any worse while waiting in the vending machine.

* Hiding in the top left-hand drawer of the desk is a stash of double-stuff oreos ..... (I could not make out what this was supposed to be. ED) ..... for special occasions.

* Underneath the oreos is a flow-charting template, left there by the previous occupant of the office. Real programmers write programs, not documentation - leave that to the maintenance people.

The real programmer is capable of working thirty, fourty even fifty hours at a stretch, under intense pressure. In fact, the real programmer prefers it that way. Bad response time does not bother the real programmer - it provides the chance to catch a little sleep between compiles. If there is not enough schedule pressure on the real programmer, he tends to make things more challenging by working on some small but interesting part of the problem for the first nine weeks, then finishing the rest in the last week, in two or three fifty-hour marathons. This not only impresses the hell out of the real programmers manager, who was despairing of ever getting the product done on time, but also creates a convenient excuse for not doing the documentation. In general:

* No real programmer works nine to five (unless it is the ones at night).

* A real programmer might or might not know the name of their spouse. The real parogrammer does, however, know the entire EBCDIC (or ASCII) code table.

* Real programmers do not know how to cook. Grocery stores are not open at three o'clock in the morning. Real programmers survive on twinkies and cofee.

THE FUTURE

What of the future? It is a matter of some concern to real programmers that the latest generation of computer programmers are not being brought up with the same outlook on life as their elders. Many of them have never seen a computer with a front panel. Hardly anyone graduating from school these days can do hex arithmatec without a calculator. College graduates these days are soft - protected from realities of programming by source level debuggers, text editors that count parentheses and "user friendly" operating systems. Worst of all, some of these alleged "computer scientists" manage to get degrees without ever learning FORTRAN! Are we destined to become an industry of Unix hackers and Pascal programmers? From my experience, I can

only report that the future is bright for real programmers everywhere. Neither OS/370 nor FORTRAN show any signs of dying out, despite all the efforts of Pascal programmers the world over. Even more subtle tricks, like adding structured programming structures to FORTRAN, have failed. Ohhh sure, some computers vendors have come out with FORTRAN-77 compilers, but every one of them has a way of converting it back to a FORTRAN-66 compiler at the drop of an option card - to compile DO loops the way God intended. Even Unix might not be as hard on real programmers as it once was. The latest release of Unix has the potential of an operating system worthy of any real programmer - two different and subtly incompatible user interfaces, an arcane and complicated teletype driver and virtual memoyy. If you ignore the fact that it is structured, even C programming can be appreciated by real programmers. After all, there is no type checking, variable names are seven (ten? eight?) characters long and the added bonus of the pointer data type is thrown in - like having the best parts of FORTRAN and assembly language in one place (not even talking about #define).

No, the future is not all that bad. Why, in the past few years, the popular press has even commented on the bright new crop of computered nerds and hackers ((7) and (8)) leaving places like Stanford and MIT for the real world. From all evidence, the spirits of real programming lives in these young men and women. As long as there are ill-defined goals, bizarre bugs and unrealistic schedules, there will be real programmers willing to jump in and solve the problem, saving the documentation for later. Long live FORTRAN!

REFERENCES TO REAL PROGRAMMERS

(1) FEIRSTEIN, B. REAL MEN DON'T EAT QUICHE, NEW YORK, POCKET BOOKS 1982

(2) WIRTH, N. ALGORITHMS + DATA STRUCTURES = PROGRAMS, PRENTICE HALL, 1976

(3) XEROX PARC EDITORS .....

(4) FINSETH, C. THEORY AND PRACTICE OF TEXT EDITORS - OR - A COOCKBOOK FOR AN EMMAC, B.X. THESIS, MIT/LCS, TM-165, MASSACHUSETS INSTITUTE OF TECHNOLOGY, MAY 1980

(5)WEINBERG, G. THE PSYCHOLOGY OF COMPUTER PROGRAMMING, NEW YORK, VAN NOSTRAND REINHOLD, 1971, PAGE 110

(6)DIJKSTRA, E. ON THE GREEN LANGUAGE SUBMITTED TO THE DoD, SIGPLAN NOTICES, VOLUME 3, NUMBER 10, OCTOBER 1970.

(7)ROSE, FRANK, JOY OF HACKING, SCIENCE 82, VOLUME 3, NUMBER 9, NOVEMBER 1982, PAGES 58-66.

(8)THE HACKER PAPERS, PSYCHOLOGY TODAY, AUGUST 1980.

MEMORY SAVERS

---------- from TI-D-BITS, by Jim Swedlow
reprinted by : K.C. 99'er CONNECTION, August 1990

This is another look on how our TI stores stuff in memory. We will look at ways to save this valuable memory space and how program speed is affected by how variables are chosen. A variable name takes only its length in memory. (A) takes 1 byte while (master device) takes 13 bytes and (A$) takes 2 bytes. A number however, takes the number of characters plus 2 bytes. For ex., the number 2.13 would take six bytes for memory and the number 1 wold take 3 bytes. Strings also take the number of characters in the string plus 2. "He won" takes 8 bytes.

A FEW SUGGESTIONS

-Use the shortest possible variable and subprogram names you can.
-If you use a given number often, assign it to a variable and use that instead. If A=2.13 takes 8 bytes, each time you use A instead of 2.13 you save 5 bytes. Therefore, after two substitutions you are conserving memory.
-Look for places where you can replace numbers with variables. This line: 10 C=0 :: INPUT A(0) :: PRINT A(0) would take 4 bytes less if done this way:
10 C=0 :: INPUT A(C) :: PRINT A(C)

Be sure, however, that you only do this when the variable must be the number you intend it to be.

SPEED

Longer variable names slow program execution. I ran this program with progressively longer variable names substituted for (C): 10 C=(0) :: FOR I=1 TO 1000 :: A=C :: NEXT I

I ran each 3 times and averaged the results. Here is what I found.

LENGTH OF (C)	AVERAGE RUN (seconds)	LENGTH OF (C)	AVERAGE RUN (seconds)
1	7.72	9	8.46
2	7.74	10	8.56
3	7.76	11	8.62
4	7.90	12	8.71
5	8.04	13	8.83
6	8.12	14	8.94
7	8.23	15	9.02
8	8.34		

I also ran this one: 10 C=0::FOR I=1 TO 1000::A=0::NEXT I

The average run time was 7.06 seconds. There is a cost when substituting variables for numbers. (TINSTAAFL)
<There is no such thing as a free lunch>

Editor's Bit

---------- by Marshal Ellis
Mid-South 99er User Group, August, 1991

* "A Brief History of the World: Certifiably Genuine Student Bloopers" written on essay questions and collected by teachers throughout the United States, from eighth grade through college level, 1987.

==========

George Washington married Martha Curtis and in due time became the Father of Our Country. The Constitution of the United States was adopted to secure domestic hostility. Under the Consititution the people enjoyed the right to keep bare arms.

Abraham Lincoln became America's greatest Precedent. Lincoln's mother died in infancy, and he was born in a log cabin which he built with his own hands. When Lincoln was President, he wore only a tall silk hat. He said, "In onion there is strength." Abraham Lincoln wrote the Gettysburg Address while travelling from Washington to Gettysburg on the back of an envelope. He also freed the slaves by signing the Emasculation Proclamation, and the Fourteenth Ammendment gave the ex-Negroes citizenship. But the Clue Clux Clan would torcer and lynch the ex-Negroes and other innocent victims. It claimed it represented law and oder. On the night of Aril 14, 1865, Lincoln went to the theater and got shot in his seat by one of the actors in a moving picture show. The believed assinator was John Wilkes Booth, a supposingly insane actor. This ruined Booth's career.

Meanwhile in Europe, the enlightenment was a reasonable time. Voltaire invented electricity and also wrote a book called *Candy*. Gravity was invented by Issac Walton. It is chiefly noticeable in the Autumn, when the aples are falling off the trees.

----- * ----- * ----- * ----- * ----- * ----- * -----

PROGRAM BIT

JULY 1991

THIS MONTH : FREE popcorn & soft drinks.

6:30 pm - Assembly Language programming class, meet in Cafeteria.

7:00 pm - Main meeting begins, general discussion. library open.

7:15 pm - Demonstrations begin, demonstrations by Jim Saemanes.

9:30 pm - Meeting ends, late dinner at location to be announced.

NOTICE

Information contained in **Tidbits** is accurate and true to the best of our knowledge. Viewpoints and opinions expressed in **Tidbits** are not necessarily that of the Mid-South 99'ers. We welcome any opinions/corrections from our readers. Articles may be reprinted elsewhere as long as credit is given to the author and newsletter.

GROUP INFO

Visitors and potential members may receive 3 free issues of **Tidbits** while they decide if they wish to join (no obligation) On the top of your label is a code. A Y means you are a member, N means 3 free list, UG means user group and S means a business. Beside the Y is a date, one year from that date your dues are due. A dollar sign ($) on the label will indicate that your dues are due. The library is open only to members. Library list is $1. Mail order disk library access is $2 for the first disk and $1 for each additional disk - max of 5 disks per month. Order by disk number only. At meetings, library access is FREE if you exchange your disk for ours or $1 per disk for our disks. Send all mail order library requests to librarian's address! Send dues and correspondence to group address.

CALENDAR

MEETINGS: AUG 15, SEP 19, (3rd Thursday!)
WORKSHOPS: TO BE ANNOUNCED

24HR TI BULLETIN BOARD

The 9640 NEWS BBS 300/1200/2400 Hayes. 901-368-0112

The Full Moon BBS 300/1200/2400/9600/14400 HST 901-386-1760

GROUP MAILING ADDRESS

Mid-South 99 Users Group
P.O. Box 38522
Germantown, Tn. 38183-0522

LIBRARY ADDRESS

Jim Saemanes
46 Higgins Road
Brighton, Tn., 38011

MEMBERSHIP APPLICATION

NAME _____ |_| $15.00 FAMILY
ADDRESS_____ |_| $10.00 JUNIOR (under 15)
CITY_____ST____ZIP_____
PHONE(____)___-_____:INTERESTS_____

EQUIPMENT,ETC._____

Detach and [illegible] with check payable to: Mid-South 99 Users Group,
P.O. Box Germantown, Tn, 38183-0522