

=====

Volume 2, Issue 3

August 1st., 1985

TI-LINES is the monthly newsletter of OXON TI USERS, and the form of presentation of any material remains the copyright of each individual author. OXON TI USERS reserves the right to submit any material to other newsletters for publication, with due credit to the author(s) concerned. Submissions are accepted on this understanding.

It is the responsibility of each author to ensure that no copyright infringement will occur by the publication of their articles in TI-LINES, and OXON TI USERS cannot be held responsible for any such infringements. Every effort is made to ensure that any information given in TI-LINES is correct.

TI-LINES is produced and published by:

```
*****  
*                               *  
*   Peter G. Q. Brooks         *  
*                               *  
*   96 Banbury Road           *  
*   OXFORD                     *  
*   OX2 6JT                    *  
*                               *  
*   Tel: Oxford 50822         *  
*                               *  
*****
```

Oxon residents are offered a subsidised subscription at £1.56 p.a., payable either as 12 second class postage stamps or by cheque.

THIS OFFER APPLIES UNTIL MAY 1986 ONLY

TI-LINES is available on Associate subscription to Users resident outside Oxfordshire, for £10 p.a. Overseas subscriptions are by arrangement. Back issues are £2 including post and packing. New subscriptions begin with Issue 1 of the current volume, up to and including the current issue, regardless of the number of issues elapsed.

Contributions should be submitted either on diskette in TI-Writer compatible files, or in a form which is as legible as possible. Art work should fit within an A4 area and should not contain colour. Very high contrast line drawings are preferred, and these may be produced by arrangement with the publisher.

I n d e x   t o   a r t i c l e sP a g e

EDITORIAL.....	3
Drained Brain. No-Win. Enhanced BASIC. Warning. The Largest Known Number. Repeat FS In TI-Writer. Errata. Bob Mack On TI-Writer Colours. Disk Explorer. Fillers. A Little Learning. Rolling Pin Time.	
PUBLIC DOMAIN SOFTWARE LIST.....	10
An Update From STEPHEN SHAW, Including Stainless Clearance	
GETTING DOWN TO BASICS.....	12
The Ups And Downs Of A TI Owner, By JOHN BAIRD	
CONTACTS.....	12
Only One This Time	
BULLETIN BOARD.....	13
Sales And Wants From Both OTIUsers And Non-members	
RICHARD OWEN.....	14
Young RICHARD OWEN, Taking On The Task Of Debugging His Peers	
SORTING AND SEARCHING.....	15
The Start Of A Series Which Should Sort Some Of You Out	
DIS/VAR WARS.....	17
Page 0 Of A Filler Which Threatens To Take Over The Universe (Further Segments On Pages 25 And 26)	
ENHANCED BASIC.....	18
As Promised Long Ago, Some Applications Examples From PAUL KARIS And FRANC GROOTJEN	
DISK FILE DIRECTORY ENTRIES.....	19
From BRUCE CARON of OTTAWA TI-99/4 Users Group, Who Examines The Directory Sectors >2 To >21	
FURTHER EXPLANATION.....	23
An Addendum To The Previous Article, Expanding On Some Details	
CLOSE FILE.....	27
Jug-heads To Egg-heads ? The Late News. OTIU Re-growth	

-----  
E D I T O R I A L  
-----

BRAIN DRAINED  
-----

The other day OTIUsers RICHARD SIERAKOWSKI and GORDON PITT and I linked up to give and receive bits of hardware and information, and we played around with Gordon's latest acquisition, a MYARC 128K card (reviewed by HOWARD GREENBERG last issue).

Later on that day, I was attempting to show Richard a piece of software which was designed to load into the MINIMEMORY module from cassette. I had duly connected the cassette recorder with the TI leads to my console, installed the MiniMemory module, and prior to loading the program with EasyBug I was attempting to initialise the module. The first initialisation went OK. The second (which clears everything completely from the unit) did not. Every time the screen presented me with HIT PROC'D TO CONFIRM and I touched the FCTN key, the menu was re-presented as if I had touched another key. I never even got the chance to hit PROC'D - just touching FCTN caused the response. I began to get more than a little concerned, and tried switching the console off and then on again, extracted and re-inserted the MiniMemory module, and wondered out loud if the battery had finally failed and if the module's chips had somehow been damaged.

Richard leaned forward and took the cassette leads out of the joystick port...  
~~~~~

A NO-WIN SITUATION  
-----

Right, I thought. I have a set of extremely nimble fingers in the form of JENNY KEANE typing up large swathes of text for me while I type and sometimes think at the same time; I have professional printer and OTIUser JOHN MATTHEWS making an excellent job of photocopying TI-LINES; now maybe I can get the blasted thing out on time.

I was on holiday (yep, I finally found out what one is - I looked it up in a dictionary) for two weeks from 22nd July (it's actually only the 17th today but producing TI-LINES lets me indulge in a little Time Travel) which meant that I couldn't put this August issue out on time (so what else is new?).

Despite my best endeavours I only managed to get the July issue to John on the 16th., and even working at top speed I couldn't finish this issue in time, so my efforts have once more come to naught. I think I could make a career of it...  
~~~~~

ENHANCED BREATHER  
-----

The alert reader will have realised that there has been no article in the much-dreaded Enhanced BASIC series for the last two issues. This has been largely due to space considerations, and possibly also because I forgot where I had carefully laid the manuscripts from PAUL KARIS when I moved recently. The first article appears in this issue with a couple of others to follow. Some of the material duplicates what has already been presented, which should enable those of you who didn't follow a word that I wrote in the previous articles to have an opportunity not to understand a word of the material which is to come.  
~~~~~

WARNING

MARTIN ROSS, a shortly-to-become OTIUser, has been reading TI-LINES and has noted a source of possible error in an article which appeared in V1.13. It's on page 47 and concerns an item from TIC TALK, newsletter of ROCKY MOUNTAIN 99ERS which describes how to disassemble the Peripheral Expansion Box (PEB). The possible error is the description of the upgrading of the power supply, mentioned in the second paragraph. The author, JOHN COLSON, says that a FAIRCHILD ua79H12KC voltage regulator can be used. Martin tells me that there are two voltage regulators with similar numbers: a 78H... which is +12 volts, and a 79H..., which is -12 volts.

It is VERY IMPORTANT that you check to see which one your PEB should have.

We know that there are a number of different PEBs knocking about, some of which have muscular power supplies, and others which don't. In some cases, the type of ON/OFF switch at the front of the PEB can indicate which is which: the ones with "rocker" switches have stronger power supply units than those with just a "push-button" switch, or so it seems.

-----  
THE LARGEST NUMBER IN THE KNOWN UNIVERSE

I mentioned last issue that ERNST NOWAK of the PHILADELPHIA AREA TI-99/4A USERS GROUP had sent me some disks with downloads from COMPUSERVE. I have had a chance to dump them out to my printer and examine them in detail. Although one segment was missing, there was still plenty to get on with, and I came across some interesting items which I was able to follow down through the message list.

The sheer volume of data which Ernst has passed to me suggests that his phone bill must have been nothing less than astronomical: there were no less than 1097 sectors of Dis/Var 80 files (downloaded presumably at something like 300 or 1200 baud). Some of the files were so large, I could not load them into my TI-Writer (running under the Extended BASIC loader until I can obtain a backup copy of the TI-Writer module), but had to do it chunk by chunk.

Some of the items were sufficiently interesting for me to consider publishing them in TI-LINES - for example, the TI FORUM, which interested me on two fronts: firstly because of its content, and secondly because I had at one time been considering a second publication for the more advanced experimenters which would contain details of the research and findings with respect to things like GPL, the direct access to the TI disk controller chip, the possibility of replacing all of the console's BASIC chips with some RAM in order to vastly increase the flexibility of the machine, and so on. Lack of time knocked that one on the head for me, but it is possible that CompuServe's Forum could have a similar content. CRAIG MILLER of MILLER'S GRAPHICS has recently joined the CompuServe TI crew, which leads me into a little aside: recently CLIVE SCALLY's TI\*MES mentioned that Miller's SMART PROGRAMMER had not been sent to the UK for several months because the UK Distributor had not paid his bills. I know the distributor in question, and although I have been asked to soft-pedal the issue, my strong sense of injustice forces me to at least state that the distributor DID pay his bill, AND the cash was banked in the USA, and there is now a legal battle because Craig claims that he has not received any cash at all. I hope that this goes a little way to setting things straight. I have a great deal of respect for the distributor and it goes against the grain when I hear no-one speaking up for him.

Back to CompuServe.

Thanks to Ernst, who also sent me a stack of his newsletters, I am now aware of something which I don't believe has cropped up before: there are TWO types of TERMINAL EMULATOR II knocking about, with DIFFERENT file formats. One is called TE2, the other TE11, and there may also be other identifying factors. The difference in the file format is that one uses DISPLAY VARIABLE 80, while the other uses DISPLAY FIXED 80. I am not aware of the exact problem which arises over this difference, but I do know that problems have arisen, sufficiently often for folks to start providing details on their TEs to a co-ordinator.

I would like OTIU to play its part in this, so I would be grateful if any TE owners could provide the following information to me and I will pass it on:

- 1) Does the label say TE2 or TE11
- 2) The copyright date on the module
- 3) The two alphanumerical sequences stamped on the back
- 4) Does it output text files in D/V 80 or D/F 80 format

RALPH and DOLORES LANDRUM are collating the information, so I will pass whatever comes my way to Ernst and he will presumably know of a way to communicate it to them. The details above were requested around late May this year.

There are also items on Forth, on number conversions (a smart-looking Extended BASIC subprogram was provided by one User which copes with all of the major conversions likely to be necessary - in just 6 lines!), and one or two other things which I found interesting: the view of HACKING for example, and the discussion of the concept of FREEWARE, all of which will receive an airing in TI-LINES over the coming issues. I believe that it should be alright to print selected items from Ernst's download as long as I give credit wherever it is due (authors, originators, CompuServe itself, etc.).

I also want to publish a list of the Bulletin Boards which was given in a recent issue of ROCKY MOUNTAIN 99ERS' Tic Talk.

As one of the writers of an item in CompuServe's BB says, we are drowning in a sea of Information, and dying for lack of Knowledge. This has only really been brought home to me within the last few weeks as I realise that if I continue to try and present information in a processed form (i.e. Knowledge) I will probably spend the rest of my life on the TI alone. I haven't even begun to scratch the surface yet.

~~~~~  
TI-WRITER COMMENT  
-----

While glancing through TI\*MES, the newsletter of TI EXCHANGE, the other day, I noticed an item from a subscriber in which the lack of a repeat function on the Find String command was bemoaned. It set me thinking, and I believe I have the answer: use Replace String to find wanted items instead. If you want to play really safe, tell RS that you want to replace the sought-after string with the same thing (e.g. RS /wellies/wellies/) just in case you hit the wrong key.

First get the machine out of Word Wrap mode with CTRL O so that you have an "open" cursor. This will avoid the hazard of having the bloody thing reformat all your text if you DO press the wrong key.

Next, get up to the top of the file with Show 1, and home the cursor with CTRL V (beginning of line). You could use CTRL H, but if you have any controls in your text you could be in for a lot of keypressing.

Then set up Replace String, including any column delimiters. For safety's sake do as suggested above, or live dangerously and use a null string (/wellies//).

Then set it in motion. If RS finds an occurrence which you aren't looking for, press N for No (don't replace). Don't press Y or A. When and if you finally find the occurrence for which you ARE looking, press S to stop.

That way, you get the Find String facility with the Replace String repetition.

A little while ago one OTIUser suggested that it would be helpful to have a series of tutorials on several TI-related items, like TI-Writer, MultiPlan, Extended BASIC, etc., etc. The aim would be to offer an instruction course parallel to the one in the relevant manual, but in Bringlish (British English) rather than Ameringlish (wossat ?) and perhaps pitched at a simpler level. In addition, subtleties could be highlighted and shortcuts described, as well as useful tips on how to avoid the depilatory effect that the 99s have on us.

Let me know your thoughts on the matter.

~~~~~  
ERRATA  
-----

As a result of doing other things (as usual) I spotted an error in an earlier TI-LINES, and as a consequence learned a little more about the TI. In this issue I am publishing another description of part of the TI disk layout which duplicates some of the information published earlier.

The reason for this is that the particular topics involved are complex, and I have found great difficulty in understanding part of the subject matter covered.

This led to me delving a little deeper and coming up with some answers, so I will expand on a small part of the article by BRUCE CARON. I owe a certain debt of gratitude to RICHARD BLANDEN for this information, which he imparted to me earlier this year but which did not register fully until now.

Back to the error.

In Volume 1 Issue 14, May 1985, page 24, the last section gives details of the use of bytes in sectors 2 to 33. There are two errors: the first is in the Dec column, which converts )1D to decimal 27 and )1E to 28. They should of course be decimal 29 and 30, which of course you all noted but decided not to make a big thing of it and spoil my day.

The second is the description of the manipulation of the bytes in )1C to )1E inclusive, and in subsequent bytes. It is also the subject of the expansion after Bruce Caron's article, so rather than duplicate things I would refer you to that expansion.

~~~~~

## A BETTER WAY TO FIND THOSE TI-WRITER COLOUR BYTES

---

As promised last issue, a presentation courtesy of BOB MACK from across the Big Pond:

"The defaults for screen colour choices are in the EDITA1 program. If you copy EDITA1 to a clear, freshly-initialised disk, the first sector of the program will be at sector >22. It will be headed by the following bytes:

FFFF 2000 2000 0420 3674

Using Disk Fixer, DISKO, or whatever, if you locate address >00F4 you should see a series of words starting with >87xx. The xx represents what you need to change.

My EDITA1 has:

87F4 - white on blue  
8713 - black on green  
8717 - black on cyan  
87F1 - white on black

The third digit refers to the foreground, the fourth to the background. Enter the hexadecimal code for whichever of the 16 colours trips your trigger. Then save the edited sector back to the disk. Copy the changed EDITA1 back to your TI-Writer disk, and you're in business."

If you don't possess a disk utility which gives you sector access; I will shortly be presenting an article based on information from RICHARD BLANDEN which should enable you to make use of the Editor/Assembler's Debugger to gain that access. It is also possible that in the fullness of time (i.e. somewhere in the next century!) I will write a utility which can be accessed through the Utility option (3) of TI-Writer and which will confer the same facility.

---

## INTREPID EXPLORER AND WHAT'S IN A (HIDDEN) NAME...

---

Talking of Disk Fixers (don't argue) it can sometimes be fun to explore beyond the horizon. If you've been sent an ostensibly single-sided disk, with a Sector 0 which says that there are only >0168 sectors, and with unused directories running full of >E5s all the way up to >21, then throw caution to the winds and try reading sectors beyond the limits - all the way up to >02D0 if you are good.

What can happen is that someone can take a DOUBLE-sided disk and re-initialise it to single-sided. They then record a few sectors or programs/data/whatever, but lurking beyond >0167 are those previously used sectors, waiting for you to rediscover them and then spend half your life trying to make out what they could possibly be. Have fun...?

Following on from the discussion of TI-Writer's sectors, because I handed my own TI-Writer over to JENNY KEANE for her to TI-Write loads of useful items for me (and ultimately you) I bought the Public Domain (well, Freeware) loader which runs under Extended BASIC from STEPHEN SHAW, so that I could still carry on while I waited for an opportunity to purchase another TI-Writer module (I really miss that Show Directory command!).

In the process of whiling away a few hours by making use of the M command in my Disk Fixer (Navarone's badly-spelled module) to examine cpu RAM outside the usual range for sector analysis, I came across some interesting names. Apart from the usual EDITA1, EDITA2, FORMA1, FORMA2, CHARA1, and UTIL1, there were also CHARA0 and UTIL2. Now these only appear to be loaded by the Freeware loader (although they may exist somewhere in the TI-Writer module as well) and are present within H131A (the loader) in coded form: the hexadecimal equivalent of their ASCII codes, flanked a pair at a time by the usual Bs. If you allow the loader to do its job, and then zap it with a Reset, and then use Disk Fixer, or whatever is useful, to examine cpu RAM from )F800 onwards, you will see the names in all their glory.

Now the problem is: what are CHARA0 and UTIL2 doing there ? There are no files present with those names and I have never seen them mentioned before. Has any bright spark got any ideas ?

By the way, the Freeware loader (no, not free-loader) was written by TOM KNIGHT. Freeware is a relatively new concept from the States, where an author can write a program and put it out as Freeware, which means that he has placed the item in what amounts to the Public Domain, but at the same time requests a contribution from whoever makes use of the item (relying on the User's honesty, so they say). That way he/she doesn't run the risks involved in mass-producing and marketing the item, but still stands to make something out of it. I'm not sure whether it is a valid or indeed useful concept, but that's not to say that it shouldn't be tried. However, it is based on the same philosophy which appears in one copier I have heard about: the User is asked if he/she has the permission of the author to copy an item; if the response is Yes, the copier goes ahead and does its job. If the answer is No, the copier NEWS itself.

Now in all honesty, I ask you: who is going to answer No ?

As to whether I have sent the requested \$7.50 for TK-Writer, as the loader is called, well, I'm thinking seriously about it. I'll let you know what I decide (honestly).

~~~~~  
FILLERS  
-----

I am currently wrist-deep in a conversion of DEC WARS (see V2.1) which has a smidgeon of HITCHHIKER'S GUIDE to the GALAXY thrown in - well, an HGG emulation, to be exact. Because there is little justification for such an item in a publication such as TI-LINES where space is more or less at a premium (or at least should be given over to useful information rather than light-hearted banter...sounds like a criticism of this Editorial), and because I often find myself with odd half pages or so blank, I will "serialise" DIS/VAR WARS and slot it in the blanks. I will also drop in "paragraph hints" developed for HOME COMPUTING WEEKLY and never used by them (the same applies to the series of 12 articles for the not-so-experienced which begins this issue). I hope that you will like it. Ars-two-three-two says he hopes you'll like it too. Peeppeiyoh won't commit himself one way or the other, but then that's his problem. Look out for the baddy, TI 'Vader, and the love interest in the shape of Princess Salmon and Prince Tronic (bit of a shocking red herring there). Luckily for all of us, the gallant Spook TI-Writer will sort things out for us along with OB1, Con Solo, and his trusty DISK+AID and comp\_nion, Disk\_Backer.

May the 4s be with you...



Now,

) READ ON

\* STRING-NUMBER MISMATCH

) READ ON\$

~~~~~  
A LITTLE LEARNING IS AN ARDUOUS THING...  
-----

I spent some time during my "holiday" strengthening my knowledge and understanding of TMS9900 Assembly Language by re-reading, and attempting to inwardly digest, the Editor/Assembler (E/A) manual. I picked up some errors which didn't appear on the Errata sheet and which threw me for quite a while.

It took me the best part of two days this time around (which should indicate to the alert reader that I have tried before) to fully understand the operation of the eight instructions on page 336, which are provided by TI to help you correctly determine which byte in the PATTERN DESCRIPTOR TABLE (also confusingly called the PATTERN GENERATOR TABLE) to change when accessing a pixel on the screen in BIT MAP mode.

Distrusting TI's code - well, so many small errors had led me to be wary of placing too much reliance on the manual's examples - I began from FIRST PRINCIPLES (i.e. started from scratch) and worked out my own method, beginning in territory I know reasonably well - TI BASIC - and translating little by little into TMS9900 Assembly Language. I knew the status of every bit in every byte of every word of the resulting instructions.

Armed with some unwieldy code, I bowed to my fetish for compactness (gained from my days spent programming calculators) and set about reducing it to as few instructions as possible.

At this point my pattern recognition skills, honed through years of trying to get TI BASIC to do what other BASICs find coming naturally, began to spot strong similarities between my code and TI's example.

Finally full comprehension dawned, and I had to admit a sneaking admiration for the programmer who had polished the routine to near perfection. (That's just the scientist in me allowing for the possibility of someone - possibly me - finding an even more compact method!). Perhaps TI were also chuffed at the concise use of the code: what better way to trumpet the skills of your staff than by sticking examples of their better work in a manual?

If anyone wants a detailed - and I mean detailed - explanation of how the routine on page 336 of the E/A manual does its stuff, bung me a request and I'll devote an article to it in a future TI-LINES (perhaps even as an integral part of the planned series on TMS9900 Assembly Language For Beginners).

Of such small personal victories the stuff of Life is made (Pompous Quote Of The Year At 3.30 am On The Second Day!!).

~~~~~

WHAT TIME DO YOU CALL THIS ?!

-----

It is amazing how quickly time can fly when you're deeply involved with your 4A. The other evening a few OTIUsers converged on the Brooks Residence for an evening of swapping information and Public Domain items. By eight o' clock everyone (4) had arrived and we began babbling. At midnight one couple left, and the other two lads and I carried on. At 1.30 we tramped out for some late night nosh, and at 5.30 one of the lads left to travel the 40 miles home.

At 6.00 his distraught wife rang me. He had said that he would be back by midnight, and at 3.00 she had awoken to find a distinct lack of husband. It was at this point that I realised that I had not once thought to remind her husband to ring home and tell his better half the score, hence the reason for writing this item.

So lads, if you're away from home and you've left the little woman behind, spare a thought for her feelings when you don't turn up as planned.

As for the fourth member of the group, well, he left at 7.00 (no wife to go home to), and I'm still tapping away on the keyboard.

~~~~~  
-----  
FROM STEPHEN SHAW  
-----  
~~~~~

Some further news from Stephen regarding his list of PUBLIC DOMAIN SOFTWARE

**NEWS:**

NOW IN: EX BAS LOADER FOR TI WRITER Vn 5.

The revised TI WRITER is supplied with Vn 5 of the XB loader. (Listed earlier)

This much improved XB loader is sold with the revised TI Writer files! (Two entries in list, same disk!!!).

This XB loader is far superior to any other: it fully supports SHOW DIRECTORY, and by amending lines 250 and 260 in the LOAD program you may enter default printer names for both Editor and Formatter. Also, by amending line 250 you may amend colour choices (CTRL 3) with the Editor.

This Loader is a very fast loader, and the new Show Directory is faster than that in the module. It is page orientated rather than scrolling, and any fractured disk files (inefficient) are marked with an asterisk. They can be repaired by using DISK MANAGER to copy the disk.

NB: If you wish to use the TI Writer revision also with the TI Writer module, please specify a default printer name for the FORMATTER option.

Using Option 3, you may load and run any program in machine code PROGRAM format.

M/C UTILITIES 1: A selection of mc utilities you may wish to use in your ExBas programs via CALL LINK. There is a simple demo for each one. GOTHIC character set for your printer, text justification, Save & Recall of VDP Screen info including colours and sprite movement detail, VDP access, and a DSRLNK file for m/c programmers to use, also a fast sorter.

CLEARANCE SALE: STAINLESS SOFTWARE

To clear stocks, the following prices will apply  
WHILE PRESENT STOCKS LAST.

In some cases only one cassette is held!

To order from this list, please either send separate cheques/ p.o.'s for each item or send a cheque with the amount not filled in.

If stock items are 'out of stock' please indicate if you are prepared to pay the full catalogue price.

These prices apply only while stocks last and in any case this offer will be withdrawn on October 5th 1985.

Please see Stainless Software catalogue for descriptions.

**£1.00 each:**

CRAZY CLIFF. Extended Basic  
SPRING HEELED JACK. Extended Basic.  
TROGMAN. TI Basic.

**£2.00 each:**

PLANET DESTROYER. TI Basic  
NOTEWORTHY. Extended Basic.  
NIGHT FLIGHT. TI Basic.  
BREAKPOINT. TI Basic  
ROD. Extended Basic

**£3.00 each:**

GOBLINS REVENGE. TI Basic  
Robin Hood. 16k Extended Basic.  
HAPPY MATH. TI Basic  
THE WALL. Extended Basic.  
GOLF. TI Basic.

**£4.00 each:**

STARPROBE 99. TI Basic  
WALLABY. Extended Basic  
HANG GLIDER PILOT. TI BASIC

NO DISCOUNTS ON THESE PRICES. Stocks are limited!!!

Stephen Shaw  
10 Alstone Road  
STOCKPORT  
Cheshire  
SK4 5AH

Best wishes

Stephen



-----  
GETTING DOWN TO BASICS  
-----

They say life is hard at the top, but I can assure you that it is even harder at the bottom, especially when it is difficult to distinguish the difference between ambition and ability. One Christmas, many moons ago, I tactfully let it be known that I would like a home computer, believing that I was more likely to get the usual pair of nylon socks. It came as something more than a little surprise when Santa brought me my TI 99/4A, the machine I had seen described as a Mercedes in a field of Minis, and had let this be known to my dear wife.

In an initial burst of enthusiasm I sat down and wrote a few programs, which after a bit of difficulty, I managed to get to work. I could get the contraption to print little messages on the screen or convert pounds to kilograms. I felt great. Ambition then took over; I decided to use the TI 99/4A for my main interest in life, football administration. I plodded away at compiling a league program without very much success, and finished up losing heart and interest.

My Christmas toy was discarded, and lay untouched for months, until I chanced to come across Stephen Shaw's book "Getting Started with the Texas TI 99/4A". I have become a enthusiast again. This time, however, I realise the pitfalls and will try to take things in easy stages. It will be hard, for fools like me will always rush in where angels fear to tread, and wise and experienced TI users avoid.

It may only be a personal failing, but I feel that it may be common-place, for I find it hard to sit down and read instructions right through. This has led to many of my problems. Before I am half-way through any information on a subject I want to be up and at it, usually resulting in frustration and a bout of blaming everybody but myself. This I will try to avoid. Another difficulty which I have is that of understanding the computer jargon. Initially I don't understand what "initialisation" is (or should it be spelt with a Z?). Perhaps there should be some sort of course on jargon before we start calling at computer shops, because I for one usually walk out in a daze, not knowing if I am stupid or if someone was trying to con me out of my hard earned filthy lucre.

At present I have ambitions of getting a printer, to help with my programming, and to print out league fixtures and tables, but I am lost in a maze of serial and parallel interfaces; and what on earth are centronics? A simple and full article on this subject, and where to get the bits (non-jargon type) would be helpful to a simpleton like me.

John M. Baird.

-----  
C O N T A C T S  
-----

PETER CROSS

8 Broom Mead, BEXLEYHEATH, Kent, DA6 7NY  
Tel: 0322 522809

-----  
 BULLETIN BOARD  
 -----

FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

KEVIN SMITH is not an OTIUser but I have given him the opportunity of advertising his items for sale in TI-LINES. He would prefer to sell the lot at one go, so if members want to club together to make a combined purchase that would be very well received. Contact Kevin after 7.00 pm on HUNTINGDON 74782:

- PERIPHERAL EXPANSION BOX }
  - 32K RAM EXPANSION CARD }
  - SS SD 40T DISK DRIVE }
  - DRIVE CONTROLLER CARD }
  - RS232 INTERFACE CARD }
  - EPSON MX80 PRINTER }
- } The lot for £400  
 } (which, if you work it out, is very reasonable)

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

I have a copy of THE BEST OF 99ER Volume 1 in good condition, going for £15 inclusive of post and packing. 350 large pages of programs and articles for the 99/4A to keep you going when you get sick of TI-LINES...

I also have a single-sided "internal" disk drive for sale, no guarantees but if it DOES go wrong I'll see what can be (cheaply/freely) done, for £50 including post and packing. (Both of these items are being sold on behalf of someone else). My details are on the front page of this issue (just in case you were wondering how to get in touch!).

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

You may recall a little while ago that I asked for anyone with a DIAGNOSTIC MODULE for sale, or knowledge of the whereabouts of one, to get in touch. Well, no-one has yet, but in the meantime I have received another request for one. I am therefore looking for TWO such modules on behalf of OTIUsers. I'd be very grateful for any information.

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

OTIUser Dr JOHN RICE (of the Clinic at the Brighton Show earlier this year) has asked me if anyone has an NTSC TV/MONITOR for sale. That's the same as the TV which was sold with the old 99/4 when it first came out. Yes, I know that I have one, but I'm using it. Anybody have one that they care to sell ?

~~~~~

-----  
R I C H A R D O W E N  
-----

First of all, this is my first piece of going solo, and writing for any magazine. So, any feedback on this subject (meaning what is written ... not how it was written!!!) will be welcome.

This page is for those awful bugs in programming. These may be syntax, or even bad programming, or even mis-using subprogram utilities.

The first problem was one of my own... It was in the Mailing List... It was that my printer ran on PIO or AXIOM type codes... And the Mailing List refused to allow me to use those codes.

My solution...

```
1141 IF SEG$(DEV$,1,3)="PIO" THEN 1170
1142 IF SEG$(DEV$,1,5)="AXIOM" THEN 1170
```

The Mailing List gives options for TP (Thermal Printer), and also RS232. PIO and AXIOM were not around at the time of writing the Mailing List.

The way line 1141 (and for that matter line 1142) is ...

```
SEG$ find a SEGment of the string
DEV$ this is the DEVICE name of the printer
SEG$(DEV$,1,3) this finds the segment of the string starting at
                character 1, and going on 3 characters
If this is PIO, or AXIOM, it transfers the execution to line 1170
```

Hopefully this will be of some help to you... I have another two routines, one is for use with CALL KEY, and branching to different routines... The other is also using CALL KEY... Or rather not... It is a routine for using a Joystick instead of keys.

After these two, I will be rather hard pushed to do any more for this type of page. So... if you have any problems needing to be solved, which you think is of help to any OTIUsers, send it to me, and I'll try and sort it out.

My address is:

```
Richard Owen,
17, Highfield Avenue,
Litchard,
Bridgend,
Mid-Glam.,
S. Wales. CF31 1QR
```

Thanks. And any feedback is appreciated.

~~~~~

-----  
S O R T I N G   A N D   S E A R C H I N G  
-----

P e t e r   B r o o k s

A u g u s t   1 9 8 5

The two processes of SORTING and SEARCHING, often strongly linked, appear more frequently in "Business" and "Scientific" than in "Games" programs, but they do crop up and it is useful to know more than just a little about them.

We'll deal with Sorting first, as in general it is not practical to Search before you have Sorted.

The process of Sorting is applied to one or more LISTS of items. Essentially it involves arranging the lists of items in a specific order: alphabetically, numerically, or whatever is required.

For example, compare the two lists of names below. They have both been arranged in alphabetical order (strictly, in "ascending" order). The first has been sorted according to FORENAME, the second by SURNAME.

```
+-----+
| FORENAME | SURNAME |
+-----+
| ALBERT | HIGGINS |
| ARTHUR | DENT |
| ARTHUR | MULLARD |
| BRIAN | BLOGGS |
| CAROLE | BLOGGS |
| JULIE | MULLARD |
| LESLEY | DENT |
| SONIA | HIGGINS |
+-----+
```

List 1

```
+-----+
| FORENAME | SURNAME |
+-----+
| BRIAN | BLOGGS |
| CAROLE | BLOGGS |
| ARTHUR | DENT |
| LESLEY | DENT |
| ALBERT | HIGGINS |
| SONIA | HIGGINS |
| ARTHUR | MULLARD |
| JULIE | MULLARD |
+-----+
```

List 2

The lists could even have been sorted according to the last letter of the FORENAME, if that was what was required.

Note two things:

In the first list, where two forenames are identical, the decision as to which is placed first is taken by referring to the SURNAME. Thus ARTHUR DENT appears before ARTHUR MULLARD, since in ascending alphabetical order D appears before M.

Similarly, in the second list, where two surnames are identical, the decision as to which is placed first is made by referring to the FORENAME. Thus BRIAN BLOGGS appears before CAROLE BLOGGS since B appears before C.

(And in case anyone notices, it is entirely coincidental that the forenames in the second list alternate between male and female).

Note also that it may not always be important, or even possible, to arrange the lists in absolute alphabetical order. The computer distinguishes between the Upper case letters and their Lower case counterparts (e.g. A and a), and in some cases the segment which is being used for sorting may not be directly linked to the rest of the list(s), and therefore be unable to be used to resolve the difference between items which are ostensibly identical.

The item, or part of the item, which is used for sorting is usually called the KEY. In List 1, the KEY is the forename; in List 2, it is the surname.

In each case, in both lists, the items are "linked" together directly; e.g.:

```
F$(1) = "ALBERT HIGGINS"  
F$(2) = "ARTHUR DENT"
```

and so on, rather than being linked "indirectly":

```
F$(1) = "ALBERT"      S$(1) = "HIGGINS"  
F$(2) = "ARTHUR"     S$(2) = "DENT"
```

and so on.

In this second example, there are two "linked lists": each element in the F\$( ) array has a corresponding element in the S\$( ) array. Thus whatever forename is in F\$(n), the corresponding surname is in S\$(n).

Sorting the first example is straightforward, as we shall see. Sorting the second example is not quite so straightforward, but as long as whatever changes are made to one array are also made to the other, everything should be alright.

The form of sorting in which one array is sorted, and a parallel array is re-arranged in tandem with it, is known as a TAB SORT, and can be done on many lists at the same time.

There are many different sorting techniques, and many variations on each, but generally the simpler the Sort, the slower it is.

A fast Sort can be several hundred times faster than a slow one, and we will examine and compare a few of the more common types of Sort.

The speed of execution of any one type of Sort depends on two things:

- 1) Just how "sorted" the list of items is already
- 2) How much data is to be manipulated each time, and how many times it will have to be manipulated (related to (1))

Other factors include the language in which the Sort is written: generally a machine code sort will complete the task much faster than one written in TI BASIC, although it will be much more difficult and demanding to write.

What exactly do we "sort", and how ?

We sort LISTS, and I have deliberately used the terms LIST and ARRAY above interchangeably, to (hopefully) indicate that the two are synonymous, at least for our purposes.

The simplest List is the simplest ARRAY - a "single dimensional array variable". (Smart alics, who know all about List Theory, can stop muttering about single items constituting a List!)

Therefore, in TI BASIC,

```
100 DIM Z$(5)
```

sets up such an array or list, and specifies that it will have 6 (yes, 6) items, entries, or elements, numbered 0 to 5 inclusive, and that they will be "alpha-numeric".



This description is a little inaccurate, since "alphanumeric" technically limits us to using the digits 0 to 9, and the letters A to Z. In fact, we will also be able to use the lower case letters a to z and various punctuation marks, as well as the "space" and even the User-definable character set (and more besides).

For the duration of this explanation, the term "alphanumeric" will encompass all the characters possible on the 99/4A, which means all characters with codes 0 to 255 (i.e., CHR\$(0) to CHR\$(255)).

For this reason, when the computer sorts a list of items into order, it may appear to do things a little peculiarly. This is because it will decide which character appears first by examining the code of that character. Thus an entry beginning with a "space" (code 32) will appear in a sorted list before the upper case letter A (code 65), and a \$ symbol (code 36) will appear before a "period" or full stop (.) (code 46).

Find a printout of the full character set available (Users Reference Guide) to see in which order characters are likely to appear.

We are not limited to sorting alphanumeric lists (i.e. STRING arrays, as in the example), but can also sort lists of numbers as NUMERIC arrays.

At the conclusion of a game, for example, you might want to provide a list of the ten highest scores achieved since the program was loaded, together with the names of the scorers. You would need to sort a list of NUMBERS (numeric array) constituting the score, which would be the KEY, and each time the score's position changed you would need to similarly change the position of the correspondingly appropriate scorer's name (the STRING array), thus performing a TAG SORT.

That begins to cover WHAT you would sort.

The next article will discuss HOW you sort.

-----  
D I S / V A R   W A R S  
-----

A tale of two CTs, adapted from two original stories in such a way as to make them both totally unrecognisable.

PAGE 0  
-----

In a small country village a wedding had taken place. It had been a short service, conducted in a friendly, informal manner by the local incumbent, and the happy couple were now esconced at the top table within a moderately large (but not too ostentatious) marquee in the gardens of the local Squire.

The toasts, speeches, and reading of cards and telegrams was yet to come; the guests were busy arranging themselves around the tables according to age, size, and number of tentacles. Representatives from all over the less fashionable end of the spiral galaxy had warped, hyped, or transported themselves to Earth in order to attend the wedding, itself a symbolic conjunction of two of the races most active in the resistance to the might of the tyrannical REMpire.

(Continued at the bottom of page 25)

## ENHANCED BASIC for efficient loading/saving of arrays

by Franc Grootjen and Paul Karis  
Blauwgras 2  
3902 AA VEENENDAAL  
The Netherlands

This article will only make sense to those who have familiarised themselves with ENHANCED BASIC the term Pete Brooks of the British users group invented for the extra's provided by CALL A/ CALL D/ CALL P/ CALL L/ CALL S/ CALL G/ CALL H in combination with a Personal Record Keeping or Statistics Module.

Imagine having completed 150 observations with values between 0-100 and that you want to save these values on tape. You don't want to use the PRK or STAT mode as the standard possibilities of these modules are too restricted for your intentions. TI BASIC allows us to OPEN and PRINT # n and so writing data on tape, but this method has several drawbacks:

- saving and loading consume a lot of time and tape. Frequently the pilot tone alone, marking the start of each record, takes longer than the actual time required for data transmission. Waiting time is wasted time.
- the compacting of data that sometimes partly solves the first problem in itself is complicated. (E.g. the string "2/2.3/5/6.7/3.4/1.3/3/34/123/4.5" can be written as one variable but used to transmit 10 numerical values)
- there is no built-in data check possibility like we have for program saving.
- when an error occurs the program BREAKS and cannot be CONTINUED, and the data loaded cannot be used and must be loaded again. The same holds true if a small alteration needs to be made in the program after the data are loaded.

ENHANCED BASIC does away with all these problems.

Back to our example: Let us presume that the 150 observations are held in the array O(1). At this point we would like to remind you that the use of ENHANCED BASIC requires you to prepare your computer with CALL P before typing in or loading the following subroutines or programmes containing these subroutines.

```
10000 REM SAVE SUBR                10100 REM LOAD SUBR
10010 CALL H(0,5,1,1)              10110 CALL L("CS1",Y)
10020 CALL H(0,10,1,2)             10120 FOR I=1 TO 150
10030 CALL H(0,11,1,2)             10130 CALL G(1,1,1,MIS,0(1))
10040 FOR I=1 TO 150               10140 NEXT I
10050 CALL G(0,1,1,0(1))           10150 REM END OF SUBR
10060 NEXT I                       10160 RETURN
10070 CALL S("CS1",Y)
10080 REM END OF SUBR
10090 RETURN
```

When your main program is at a point that you want to save the values of array O it only takes a line GOSUB 10000 to do so. In the same way GOSUB 10100 will fill the array with the values played from the tape.

Detailed explanation:

```
10010 tells the computer that one item will be used per page.
10020 determines this item to be of type 2 (=integer)
10030 reveals that you do not intend to have values beyond 99. Width is 2.
10040 to 10060 stores all 150 values of the elements of array O in MEM2.
    In PRK language: item 1 on page 1 gets a value of 0(1)
10070 saves the array as a PRK file from MEM2. CHECK TAPE (Y/N) is asked.
10110 loads PRK file from tape in MEM2
10120 to 10140 is a loop that fills array O with the values in MEM2
```

We recommend you study this example closely and you will discover other possibilities that are meaningful to you. Change lines 10020 and 10030 to suit your data.

Use nested loops for multi-dimensional arrays. Use your head, as in computing:

The sky is the limit.

Murphy's Law No.516 Part 5 "Directory Entry"

---

by Bruce Caron

---

In this last part of the series we will be looking at the file directory entries. Each and every file or program that you have stored on a diskette must have a directory entry. The entry is a 256 byte block of data, that is normally stored on sectors h02 through h21 (2-33). To find out where all the entries are just look in the Directory Link Map sector (1). If you have more than 31 files on a diskette the entries may be located on any sector except 0 and 1.

A directory entry contains everything you ever wanted to know about a file, but didn't know what to ask. Rebuilding a directory entry from scratch is quite a job, but it is by no means impossible. Here's what's where, and what's what on a directory entry.

| ADDR | SIZE | DESCRIPTION                                                      |
|------|------|------------------------------------------------------------------|
| ---- | ---- | -----                                                            |
| 0000 | 10   | Filename                                                         |
| 000A | 2    | Unknown (always set to h0000)                                    |
| 000C | 1    | File Flags                                                       |
| 000D | 1    | Number of records per disk sector                                |
| 000E | 2    | Total number of sectors in the file                              |
| 0010 | 1    | End of File offset                                               |
| 0011 | 1    | Record size                                                      |
| 0012 | 2    | Number of records in file, with the order of the bytes reversed. |
| 0014 | 8    | Unknown (always set to h0000)                                    |
| 001C | 3    | Block Link - 3 byte clusters for each fraction of a file.        |

---

The directory entry specifies the type and location of files currently on the disk. The two areas that I have marked unknown in the directory entry have been filled with h00 on every disk that I have ever examined, I really don't think they are used for anything.

h0000 FILENAME: 10 characters (no spaces or periods), if someone wants to be tricky they can put a DEL character here (Fctn V) and it won't be displayed on the screen when you catalogue a disk.

h000C FILE FLAGS: The file flags are actually bits in the byte, you have to take the byte apart to find out what they all mean. The bits are numbered

0 to 7, and they are laid out from left to right.

Bit 0 : Type of record. 0=Fixed length records  
          1=Variable length records

Bit 1 : Not used

Bit 2 : Not used

Bit 3 : Not used

Bit 4 : Protect Flag 0=Not delete/write protected  
          1=delete/write protected

Bit 5 : Not used

Bit 6 : Basic File information 0=Display format  
          1=internal format

Bit 7 : Type of file 0=Data file  
          1=Program Image

h000D NUMBER OF RECORDS PER SECTOR : This byte is supposedly used only for fixed length records, however, I have seen cases where this byte is used with variable length records.

h000E TOTAL NUMBER OF SECTORS IN THE FILE : When you catalogue a diskette this is the word (2 bytes) that tells you what size the program or file is. The value that you see on a catalog (Size), is always 1 more than this value because you have to add 1 to account for the directory entry.

h0010 END OF FILE OFFSET : This byte is not used with fixed length files. For variable length files it points to the end of file marker (hFF) for the last sector in the file. For program image files it points to the byte that follows the last data byte in the file.

h0011 RECORD SIZE : For fixed length records it contains the actual record length. For variable length records it contains the maximum record size allowed. Not used for program image files.

h0012 NUMBER OF RECORDS IN FILE : The total number of records in the file with the order of the bytes reversed. For example if a file contains 423 records (h01A7) the entry would be hA701. Not used for program image files.

h001C BLOCK LINK : The block link is a three byte entry that gives you the exact location of the program or file on the diskette. When you store a file on a freshly initialized disk the controller puts the information down on a sector and continues to write the info on each consecutive sector until finished. If you remove a file from a disk that is partially full you will end up with holes where a file is no longer present. The controller stores the info on a first come first serve basis, so it will fill in these holes then it will look for free space elsewhere on the diskette. When a program is stored in this fashion it is said to be a FRACTURED FILE. Interpreting these fractured files is probably the most difficult thing to grasp.

To interpret a block link requires a little bit of manipulation and an open mind. Whoever invented this format was probably high on something at the time. Here goes..

Unlike the bit manipulation, this requires a little nybble manipulation.

What is a nybble? Good question. A byte is made of 8 bits, the left 4 bits is called a nybble, and the right 4 bits is also a nybble. Got that! OK.

hFF = 1 1 1 1      1 1 1 1  
           nybble      nybble

The nybble order in the 3 byte Block Link has to be changed around somewhat to make any sense out of it.

Nybble order : SS2 SS1 - R01 SS3 - R03 R02  
 Change to : SS3 SS2 SS1 - R03 R02 R01

These 3 byte clusters point to the Start Sector (SS) and the highest Record Offset (RO) in the cluster. For example suppose we have a block link with the values h26 h90 h02. Convert them to the order that we can read.

SS2 SS1 - R01 SS3 - R03 R02 = h26 h90 h02  
 SS3 SS2 SS1 - R03 R02 R01 = h026 h029

We end up with the values h026 and h029. The Start sector address is h026 and it continues for h029 consecutive sectors. Thats all there is to it.

On the next page are examples of 4 different directory entries. I have only included the first 128 bytes of each entry so I could fit the 4 of them on the same page. Examine them and you should have no problems identifying the various bytes and their functions.

| FILENAME  | FLAG | REC/SECT | SIZE | EOF | REC/SIZE | REC/FILE | SS  |
|-----------|------|----------|------|-----|----------|----------|-----|
| HOROSCOPE | 09   | N/A      | 0023 | 6F  | N/A      | N/A      | 1BE |
| DRAW/O    | 08   | 03       | 0025 | N/A | 50       | 006E     | 0E1 |
| PRGLIST   | 82   | 04       | 0017 | A9  | 3C       | 0017     | 022 |
| TYPING    | 01   | N/A      | 0019 | 01  | N/A      | N/A      | (1) |

| (1) | FRACTURED FILE | START SECTOR | OFFSET | LENGTH |
|-----|----------------|--------------|--------|--------|
|     |                | 051          | 006    | 007    |
|     |                | 081          | 00F    | 009    |
|     |                | 138          | 010    | 001    |
|     |                | 007          | 011    | 001    |
|     |                | 00D          | 012    | 001    |
|     |                | 00F          | 018    | 006    |

I hope this series of articles has shed a little light on what used to be a big secret by TI. I enjoyed writing this series, and I'll probably continue to write a few more.

SAMPLE DIRECTORY ENTRIES

| ADDR | 0 | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | A | B | C | D | E | F | INTERPRETED     |             |
|------|---|------|------|------|------|------|------|------|------|------|---|---|---|---|---|---|-----------------|-------------|
| 0000 | = | 484F | 524F | 5343 | 4F50 | 4520 | 0000 | 0900 | 0023 |      |   |   |   |   |   |   | HOROSCOPE ***** |             |
| 0010 | = | 6F00 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | BE21 | 0200 |   |   |   |   |   |   |                 | o*****>!*** |
| 0020 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |                 | *****       |
| 0030 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |                 | *****       |
| 0040 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |                 | *****       |
| 0050 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |                 | *****       |
| 0060 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |                 | *****       |
| 0070 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |                 | *****       |

PROGRAM IMAGE FILE

| ADDR | = | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9 | A | B | C | D | E | F            | INTERPRETED   |
|------|---|------|------|------|------|------|------|------|------|------|---|---|---|---|---|---|--------------|---------------|
| 0000 | = | 4452 | 4157 | 2F4F | 2020 | 2020 | 0700 | 0803 | 0025 |      |   |   |   |   |   |   | DRAW/O ***** |               |
| 0010 | = | 0050 | 6E00 | 0000 | 0000 | 0000 | 0000 | 0000 | E140 | 0200 |   |   |   |   |   |   |              | *Pn*****a@*** |
| 0020 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |              | *****         |
| 0030 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |              | *****         |
| 0040 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |              | *****         |
| 0050 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |              | *****         |
| 0060 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |              | *****         |
| 0070 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |              | *****         |

DIS/FIX 80 FILE

| ADDR | = | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9 | A | B | C | D | E | F             | INTERPRETED  |
|------|---|------|------|------|------|------|------|------|------|------|---|---|---|---|---|---|---------------|--------------|
| 0000 | = | 5052 | 474C | 4953 | 5420 | 2020 | 0000 | 8204 | 0017 |      |   |   |   |   |   |   | PRGLIST ***** |              |
| 0010 | = | A93C | 1700 | 0000 | 0000 | 0000 | 0000 | 0000 | 2260 | 0100 |   |   |   |   |   |   |               | )<*****" `** |
| 0020 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |               | *****        |
| 0030 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |               | *****        |
| 0040 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |               | *****        |
| 0050 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |               | *****        |
| 0060 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |               | *****        |
| 0070 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |               | *****        |

INT/VAR 60 FILE

| ADDR | = | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8 | 9 | A | B | C | D | E | F              | INTERPRETED |
|------|---|------|------|------|------|------|------|------|------|---|---|---|---|---|---|---|----------------|-------------|
| 0000 | = | 5459 | 5049 | 4E47 | 2020 | 2020 | 0000 | 0100 | 0019 |   |   |   |   |   |   |   | TYPING *****   |             |
| 0010 | = | 0100 | 0000 | 0000 | 0000 | 0000 | 0000 | 5160 | 0081 |   |   |   |   |   |   |   | *****Q`**      |             |
| 0020 | = | F000 | 3801 | 0107 | 1001 | 0D20 | 010F | 8001 | 0000 |   |   |   |   |   |   |   | p*B***** ***** |             |
| 0030 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |   | *****          |             |
| 0040 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |   | *****          |             |
| 0050 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |   | *****          |             |
| 0060 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |   | *****          |             |
| 0070 | = | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |   |   |   |   |   |   |   | *****          |             |

FRACTURED FILE BLOCK LINK

-----  
FURTHER EXPLANATION  
-----

Addendum to BRUCE CARON's article

A couple of points need expanding to fully explain them:

- 1) How the fractured file "clusters" are decoded
- 2) How the length of each "fraction" is calculated from the "offset" - and even what an "offset" is.

Taking Bruce's example of a fractured file - TYPING - you should be able to see a string of hexadecimal digits beginning 5160 0081 etc., from byte 1C onwards. Where is 1C ? At the far left of the TYPING printout there are rows of numbers - 0000, 0010, 0020, etc. We want 1C or 001C, which lies somewhere between 0010 and 0020. Look along the 0010 line. At the top of the printout there are further column headings: 0, 1, 2, ... F. These are examples of another use of "offsets". Look for C and trace down to where its column intercepts the 0010 line. 0010 + C = 001C. The term "offset" describes a value which is added to a "base" (which is itself a value) to give the final value used as an "index".

The base is 0010, the offset is C, so the index is 001C. Clear as mud ? That's what I thought. This is not THE offset which we will be examining, but it is a good example of how one works.

Anyway, let's go back and look at that string of digits which start at address 001C. The rule seems to be that you start at address 1C and finish when you encounter six zeros (00 00 00). On that basis, we end up with:

5160 0081 F000 3801 0107 1001 0D20 010F 8001

Now what ? Well, these are the "clusters", and they are always gathered in groups of 3 bytes (or six nybbles, or six hex digits).

Breaking them up into their triplets gives:

516000 81F000 380101 071001 0D2001 0F8001

Right. Let's do a little substitution. Take the first three bytes - 516000, and substitute the letters PQRSTU for them. Thus P is 5, Q is 1, R is 6, and S, T, and U are 0, 0, and 0.

The manipulation goes like this:

Take S and put it IN FRONT OF P (SPQR TU)

Take R and put it AFTER U (SPQ TUR)

That's the cluster decoded. In terms of the actual digits:

516000 = 051 006

>051 is the start sector of the fraction, and >006 is the offset, but not from >051. It is not >051 to (>051 + >006), strictly speaking. There is a general rule, but its application to the first fraction makes it appear to result in a segment which runs from >051 to >057 - which is correct, confusingly.

If you look at Bruce's list of the fractured file, you will see that the first entry in the fraction list gives >051 as the Start sector. The offset is given as >006, but the length is given as >007.

Why ?

Well, if the fraction was only 1 sector in length - i.e., everything resided in sector >051, - the offset would have been >000. An offset of >006 means that the sectors >051 to >057 INCLUSIVE are used (>051 to >051 + >006):

>051  
>052  
>053  
>054  
>055  
>056  
>057  
-----

Total = 7 sectors in use

So far, so good. Now look at the next cluster, >81F000. Decoding this in the manner of PQRSTU yields a start sector of >081 for the next fraction, with an offset of >00F.

This does not however mean that the fraction runs from >081 to >081 + >00F.

Nothing as simple as that. It means that the second fraction uses enough sectors to make the combined offset for both first and second fraction come to >00F. That's the difference.

So, how do you work out how many sectors long this second fraction is ?  
From >081 to >081 + ???

The answer ? Ho, ho, ho. A little deduction is called for.

What we know:           The first fraction has an offset of >006  
                          The first and second combined have an offset of >00F

Therefore, take away the offset of the first from the combined total.

$$\text{>00F} - \text{>006} = \text{>009}$$

Remember that the offset of >006 meant a LENGTH of >007 (if it helps, remember that when you DIM an array in BASIC: DIM LENGTH(6), for example, it in fact means that there are 7 elements - 0 to 6 inclusive). The total offset of >00F really means a total length of >010 (i.e. 1 more than >00F), and in terms of LENGTHS, >010 - >007 = >009.

The length of the second fraction is therefore >009 sectors. It runs from >081 to >089 inclusive.



You could ignore all this however, and exercise a little pattern recognition.

If you look at the OFFSET and LENGTH columns in Bruce's article, you should be able to spot it. If you subtract one offset from the next one in the list, you get the length of the next fraction:

```
>00F - >006 = >009
>010 - >00F = >001
>011 - >010 = >001
>012 - >011 = >001
>018 - >012 = >006
```

The last offset is >018. This indicates that the total number of sectors used by all the fractions in the file is >000 to >018 inclusive, or >019 sectors (decimal 25). One additional sector is used to hold the information relating to the file - i.e., the one that we have been examining - so in total TYPING comprises 26 sectors. If you could have seen the catalogue of the disk on which TYPING was resident, you would have seen 26 as the number of sectors used. (However, I note that some of the directory sectors - >002 to >022 - are used by TYPING, which seems to suggest that it might have come from a Forth disk...)

Incidentally, the description of the decoding in TI-LINES V1.14, page 24, for bytes >1C et seq., is in error, and yet manages to be right 90% of the time. Try using it on the TYPING file, and watch a start sector of >038 appear, with an offset of >1010 !!

I could end by saying "It's easy when you know how" only I don't somehow think that you'd agree with me...

~~~~~  
(Continued from the bottom of page 17)

FICTION

The guests' attempts at organising themselves were not being met with success. Tentacles waved in protest as a four-legged bicranial leaped onto one of the tables and galloped to an empty seat, scattering cutlery, crockery, and tastefully-arranged bouquets of artificial flowers. In the midst of the chaos, conducted, it must be said, in relative silence, a green proboscis was seen to edge cautiously towards a large bowl of blue jelly perched precariously on a tiny dais in the middle of the top table. With a loud schhluurp! the jelly shrank and then disappeared abruptly.

It fell to Syn, Guardian of the Vernier Gauge, known to all and sundry as Syn, The Sizer, to set the proceedings in some semblance of order.

A stockily-built being from the LPC System, Syn commanded respect throughout this part of the galaxy. His home planet was constructed almost entirely of pitch, and had the dubious distinction of being the first world to lose all its cities, towns, and hamlets when fire was first discovered. (They all melted.)

Syn and his fellow inhabitants of the LPC System were prone to getting pretty excited about pitch, and an official at the wedding had, with some forethought, provided Syn with a large container of the black tar before the service began, with instructions to assist in the smooth running of the proceedings generally. Syn hadn't decided whether to spend or eat this windfall; after all, when your home planet's economy revolves around pitch, what you don't eat, you spend. Or wear.

Out came Syn's instrument of office, a large vernier gauge encrusted with precious jewels, and pitch, of course. He methodically examined each guest in turn, arranging them according to width, using a Straight Insertion Sort algorithm, and in no time the guests were seated.

## FICTION

Everyone was quiet and satisfied, apart from a blob of orange gelatine which complained bitterly that Syn had attempted unlawful intercourse no less than ten times with her, in full view of everybody. The furry insect to her left explained that when Syn had picked her up and placed her on a chair, it had not been his fault that all of his fingers had penetrated her squishy exterior.

The blob absorbed this for a moment, and then noted that Syn had only nine fingers. She was about to argue the point with the insect, but thought better of it. There was always the prospect of a juicy paternity suit.

Syn now found himself to be the only unseated individual, and as he looked around for a likely space, the groom's father lurched awkwardly to his foot and roared:

"A speech! A speech! Give us a speech, Syn The Sizer!"

and then fell over backwards.

Syn would have demurred, but realised that a speech would give him the chance to walk around and look for a vacant seat.

"My Lords, Ladies, Gentlemen, SDP Voters, and holders of the title Man Of The Match. We are gathered here today to bear witness to the marriage of Princess Salmon -"

He bowed to her as he spoke,

"- and Prince Tronic."

He bowed to the young man seated next to her.

"With this symbolic conjunction we embark upon a road destined to lead us out of captivity and into freedom. Even as I speak, the Mint Imperial Guards of the REMpire are searching every sector of this disk. Fortunately for us, the Bit Maps they are using have been modified by the Lord High Protector, but it will not be long before their cyclic redundancy checks reveal their parity error."

In a corner of the marquee, hidden in the shadows, the Lord High Protector nodded his weary acknowledgement. His clothes were covered in a fine powdery dust, a souvenir of his illegal entry through the sleeve of the jacket of the spiral disk, conducted against the vector of rotation. His exit through the Index Hole had gone unremarked, a narrow escape which had nearly cost him an arm and a leg.

"It is therefore imperative,"

Syn continued,

"that we sup up sharpish before the pubs close and catch the last S100 bus home before Closedown."

He nodded curtly to indicate the completion of his speech, and also because he had seen a spare seat next to a blob of orange gelatine.

However, not a million light years away, something was happening which was to change the lives of everyone present at that fateful Reception.

~~~~~

-----  
C L O S E F I L E  
-----

FROM EWERS TO USERS  
-----

There is always something that a so-called Old Hand can pick up from the Not So Old Hands. I have always described members of OXON TI USERS as OTIUers, and felt that it did not come "trippingly off the tongue". Young RICHARD OWEN, who in this issue is trying his hand at debugging his peers (if they will let him), has found a better epithet: OTIUers. Instead of Ewers, we have become Users, which is what we should have been all the time. It's always the little things that trip the unwary (me).

HERE IS THE LATE NEWS:  
-----

I have decided to follow the example of some of the computer manufacturers, and turn an undesirable attribute into a desirable one: from this issue, TI-LINES will contain a new feature: Built-In Lateness.

Yes, many famous figures in History are now known to have been Late in their time. Lewis Carroll's White Rabbit was Late, as was the Late Show. A famous conqueror, Alexander The Late, thrived on this feature, until he accidentally arrived on time for a battle and was killed, whereupon he became Late again.

Few other publications in this field can claim to have this unique feature, and no doubt they will rush to emulate TI-LINES' example.

So don't forget, you saw it first, late, in TI-LINES...

THE MORE, THE MERRIER ?  
-----

I have recently been contacted by two more overseas User Groups with a view to exchanging newsletters. This will afford us greater opportunities as far as receiving news and information is concerned, and will probably give the other groups the opportunity of seeing their articles regurgitated in yet another TI User Group publication. I am trying to find time to increase the pool of original information/articles, and part of that effort is reflected in the start of a new series this issue on Sorting and Searching, followed by a future series for the not-so-experienced programmer. I am still putting together material for the booklets (the outline for the second is almost completed, and I should begin writing the first within the next few weeks - provided I can review the reviews of the TIHOME SOFTWARE COLLECTION which have almost all been returned.

Watch this space (but not for too long; your eyes will go funny...)

Peter Brooks

This page is blank