

NATGUG

NEWS

Volume 7

Issue 12

JUNE 1986

OFFICIAL JOURNAL OF THE

National TRS-80

& Genie Users

Group.

INFORMATION ON THE GROUP

Membership of the group is by subscription to the Newsletter, which is published monthly. Membership details are obtainable from the Group Secretary. Membership of the group is open to anyone with an interest in computers but special emphasis is placed on equipment in the TANDY range.

Details of the Group accounts, and the constitution of the Group, are available from the Secretary.

Members requiring assistance with problems related to the TRS-80 / Video Genie may call the Secretary. An attempt will be made to put them in touch with a member who can help with the problem.

Workshops are arranged from time to time in various parts of the country.

Sub-groups exist in many areas. A list is provided in the Newsletter from time to time.

The Group maintains two software libraries (Models I and II) which are free to members. Library lists are available from the Secretary.

For confidentiality reasons, the membership list is not generally available, but members may ask the secretary for a list of members in their area, and mailshots to all members may be arranged.

Back numbers of the Newsletter are available from the Secretary.

Please send all contributions for the Newsletter to the Editor, on disk if at all possible (5.25", NEWDOS-80 v2 or Montezuma Micro CP/M preferred. any combination of density, sides or tracks, but please say what it is). Your disk will be returned.

Newsletter Editor
Geof Smith
17 Homefield Rd.,
Bushey,
Herts WD2 3AP
01-950-6345

Secretary & Newsletter Publisher
Brian Pain
24 Oxford Street,
Stony Stratford,
Bucks. MK11 1JU
0908-564271

Blandford Computers
STD (0258) 53737

While stocks last:

Tandy Model 4			
1 x 40T SSDD	£265.50	2 x 40T SSDD	£312.26
2 x 80T DSDD	£586.96	4 x 80T DSDD	£800.00

Models 100 and 1000 - Ring for Special LOW price
Model 3000 Multi user now working
Tandon cut price range of PCX & PCA machines
Amstrad cut price range of machines

Hard drives for the Model 4 - 10, 20 Meg etc.

Floppy drives - 5-1/4" 80/40T DSDD switchable
3-1/2" drives

Printers: Tandy DWP 210 - Quen Daisy wheel - Citizen 210D all
at £140.00, subject to availability

Disks: 5-1/4 at £6.97 / 10 black 40T DSDD
5-1/4 at £12.18 / 10 coloured 80T DSDD

Printer ribbons - paper - phones - all TANDY products

Part Exchange - send SAE for our secondhand list

**** VERY SPECIAL OFFER ****

TANDY 2000 SYSTEMS

26-5103 2 Disk, 26-5111 mono monitor	£ 715.00
26-5103 2 disk + colour monitor 26-5122/5140/5141	£1030.00
26-5104 1 disk 10 meg hard drive, 26-5111 mono monitor	£ 948.00
26-5104 1 disk 10 meg hard drive + colour monitor	£1262.50

Blandford meeting on Sunday, September 7th, FREE lunch.

All prices are exclusive of VAT
and are subject to change without notice

CONTENTS

Information on the group	2
Editorial	5
Letters.....	6
Automatic PDRIVE's.....	7
Bugs in BASCOM.....	8
Comment on Kermit.....	9
5-1/4" Disks on the Amstrad.....	10
Extra Memory on the Model 4	10
TRSDOS Extra Memory Patch	11
MIX - A Versatile Editor	13
Kermit - Part III	17
Readers Adverts	24

EDITORIAL

Thank you to all the contributors, we made 24 pages with one article to spare for next month - keep up the good work. I received my CPM user group newsletter this month, it is published approximately every quarter. I normally look forward to quite an entertaining read with articles from DEBUGGER etc. However, I was in for quite a shock. Apart from a number of readers letters, virtually the whole issue was devoted to Software Lists from the various CPM libraries and their new MSDOS libraries. Even the software had a certain lacklustre feel with yet more disks full of terribly exciting BASIC versions of reversi etc. It made NATGUG look positively bursting at the seams with useful / interesting information.

With respect to the letter I sent to TANDY HQ, a member phoned me up in the week to say that he'd seen a copy in his local Tandy Store. The manager confirmed that it had been sent out to all the branches. Perhaps our secretary would like to give us a report next month on the state of the membership and if there has been any direct referrals from Tandy stores (Brian ?). On the subject of Tandy stores, Colin Ashford rang me this evening to say that Tandy were selling CPM-Plus and C-BASIC for 9.95 pounds. This is in 8 inch (model II) format but includes full documentation. Tandy UK seem to be doing a bit of consolidation in their UK retailing and are about to close 8 of their 16 AT Computerworld Stores. If you remember, these were set up in conjunction with Apricot with ambitious plans to become the largest European retailer of personal computers. This joint venture split up earlier this year and obviously could not function now since Apricot have dropped their low end machines completely, the XEN now being their only micro. One presumes that the financial pressures induced by the ultra cheap Korean and Taiwanese PC clones forced this move from Apricot, how will TANDY's range stand up to this competition I wonder.

I haven't seen much interesting 8-bit software this month. However, one piece of software which has been introduced first to CPM rather than MSDOS is KAMAS. This is an 'ideas' or 'outline' processor rather in the style of Think -Tank or Ready. It has had pretty reasonable reviews in Byte and 80-Micro and may be worth thinking about if you do the sort of writing that needs a lot of organization. One final comment, I've just seen Os's latest advert and the price of the Model 2000 looks very attractive. Presumably Tandy are going to stop production before very long, but none the less, it represents a very powerful piece of computing equipment at that price (it makes the IBM PC look very pedestrian) and it was popular enough in the States for the major software companies to make sure their programs would run on it (Lotus etc). I would reckon they will sell like hot cakes in the same way as the 4P before it was discontinued.

LETTERS

Dear Ed.

Not so much a contribution as a series of questions, and a thank you list. First of all, things I need:

Does anyone have any information regarding the model 4P hi-res board? Like how much? Where from? Is it worth it? Is it fast? Does anyone have a second hand one for sale?

How do I make model 3 programs like NEWDOS 80 v2 (thank you Tony Evetts) run in model 4 mode, with a 80 x 2 screen, at 4 MHz etc?

What editor assemblers are available for the model 4 mode? Again, how much, where from, and is it available secondhand?

I need Visicalc 4 or better (?). As above.

What is DosPlus 4 (and all the other questions again)?

Are there any other TRS80 model 1 or model 4/4P users in my area (Sussex)? If so, how about sharing a subscription to 80-Micro, as my wallet is not that spacious.

Would someone mind announcing once in a while when the weekends are? This is my fifth year of membership, but and I have still attended no workshops (or whatever they are called now), since I have not heard about them in time to arrange transport.
(See Blanford Computers Advert and April, May OGGY - Ed)

Thank you to Brian, for helping with the newsletter delivery problems and for referring me to Robert Viera.

Thank you to Robert Viera, who helped me with a fairly unique little problem I was having whilst fitting my two DS 40/80 track Mitsubishi drives into my 4P. (Would you care to share the problem and solution with the rest of us - Ed)

Jon Silver - Brighton

Tel: (0273) 501901 after 7 pm & weekends.

or (0424) 216611, ext 196, from 8:30 a.m. to 5:30 p.m.

Automatic PDRIVE

I normally use LDOS as an operating system whenever I possibly can, but sometimes circumstances force me to use NEWDOS-80. This changing between operating systems tends to give rise to several headaches.

The worst problem is in trying to read an unidentified disk and you have no idea of the correct PDRIVE. Unlike LDOS, you cannot expect Newdos to work it out and it is very pedantic.

At one time I had to keep a box of disks which Pdrive'd Drive 1 to read one or other of the likely disks, 80/40 model I/III single and double density. Mainly it is a matter of familiarity and that was why I tried to keep off Newdos 80 wherever possible.

Then I found myself returning to Newdos to run Model III programs and the problem started again. I knew a disk should have been OK but none of my regular disks would read it. That was when I decided to get Login. This utility looks at an unknown disk and works out the Pdrive regardless of how the system has been set up. It will even read a 40 track disk in an 80 track drive without a hardware switch.

Login only takes a small amount of space and is called by "Login n". It defaults to drive 1 and it is claimed to recognise any legitimate Pdrive. I found it works quickly and has worked with some unlikely combinations. In fact it has only once failed to recognise a disk and that turned out to be DOSPLUS which was not fair. Although it did read a 40 track disk in an 80 track drive it may not read an 80 track disk in a 40 track drive.

I have not seen it on sale in this country and got mine sent in from Seatronics who seem to be offering some unusual hardware and software options. Apart from their memory expansion they offer a speed-up kit. This is claimed offers a real advantage in working speed over the standard machine. I have always fought shy of these ideas and the only one I ever had for the Model I, never worked from the time that I bought it. Recently I was told that the memory that I had installed may not have been fast enough. Now they tell me!

Generally my fears were based on the risk of speed against reliability and the thought that there was probably a very good reason for keeping the speed where it was. The advert seemed to make sense but I have still not risked my cash or computer.

According to Seatronics the TRS-80 always keeps the speed of the Z80 at below its top speed so that slower and cheaper memory can be used. After a few years the prices began to drop but Tandy continued for reasons of compatibility and inertia. Many of the later 4's and most of the European models have faster chips but have waits or delays put in to slow them down. It is possible to get even faster chips and the latest 256K chips have a 125 ns operating speed. This allows several speed-ups although the very

fastest operation requires a new Z80 to be fitted.

I do not know how well the kits work on the model 4 but a friend swore his was a big improvement on a model I. Search and find and all calculations in Visicalc were much quicker. In fact some of the tardiness in benchtests seem to relate to these 'wait states'. I for one would welcome a more authoritative opinion.

Derek Traylor.

(There is also a public domain program called CAT by Steve Ridall that does essentially the same job in automatically setting up the PDRIVE tables for unknown NEWDOS disks. Ed)

BUGS IN BASCOM

The Model III version of Microsoft's Basic Compiler, which Tandy sold at about #129 and Harding's were selling early this year at a substantial discount, has one serious bug. It occurs in sequential input using INPUT# or LINE INPUT#. If the last byte of a physical sector on the disk is a carriage return (ODH), the compiled program goes back to read from the beginning of that sector instead of going on to the next sector. The fault appears to occur on all DOS, so must be a fault in the compiler or the runtime routines.

There are two possible ways round it. If you are using INPUT#, a comma instead of a carriage return gets over the sector boundary, so it may be worth including a random access routine in your programs to test the last byte of each sector for ODH and change it to a comma if it is. Instead of LINE INPUT#n you can use INPUT\$(1,#n) for single-character input and test for ODH to build up the string. That is of course much slower but seems reliable.

Mention of INPUT\$ is a reminder that you can use most of the additional commands and statements available under CP/M MBASIC (versions 4.51 to 5.3) or Model 4 Basic in a compiled Model III program, although of course you can't test them with the interpreter. You can choose whether to compile using 4.51 (i.e. Model I/III Basic) lexical conventions (spaces don't matter, 2-character variable names) or 5.3 (i.e. Model 4 Basic) ones (must have spaces, can have long variable names with reserved words embedded). You can also do without line numbers if you want, and you have double-precision transcendental functions. You can't use Basic statements peculiar to non-Tandy DOS (e.g. Apparat's extensions). Despite that and the bug, however, BASCOM seems good value at Harding's price (less than one-seventh of the standard prices for the CP/M and MS-DOS versions, which do no more). It also allows greater program portability (if the program compiles and runs under Model III BASCOM, it'll probably compile and run under CP/M BASCOM and MS-DOS BASCOM).

The bug doesn't occur in the CP/M and MS-DOS versions, both of which I use regularly.

The MS-DOS version (at least v. 5.35, which we have in the office) does, however, have a similar bug. It can handle strings up to 32K long, because it uses 4-byte string pointers instead of the old 3-byte ones. This excellent feature hasn't been implemented properly. If you read a string longer than 255 bytes from a sequential file, the string space is corrupted and the program will eventually crash - not always at the same point, which makes you think there's something wrong with the hardware! You're lucky if you get an error message. Again, single-character input with INPUT\$(1,#n) is the way round it, but slows down performance severely. It's probably wise not to try to write a string longer than 255 bytes to a sequential file either. Moreover if you are handling several very long strings in memory you can get a memory overflow and the program will crash without an error message. As usual, 'software support' in such cases is a joke - more annoying when you've spent an IBM-style sum of money than a TRS-80 style one. When will Microsoft issue a product which works properly as documented?

Chris Currie

COMMENT ON KERMIT.

Kermit has gained some acceptance as a file transfer protocol. I feel this is a pity as we had a perfectly acceptable standard in the Christensen protocol. I suppose we must now accept the Kermit protocol but I feel we should not enthuse over it as the Christensen protocol was perfectly adequate. It had good error checking - I have transferred hundreds of files using it and never got a bad byte !. The problems of having two protocols far outweighs any shortcomings of the Christensen one.

I suppose in these days of unemployment the possibility of someone being able to market a Christensen - Kermit convertor will at least provide someone with an income.

Roy Barber

(* I think you slightly misunderstand the situation Roy. In the computing field outside of micros, literally dozens of communication protocols exist, each being somewhat tailored to a particular machine environment. The x-modem protocol was primarily designed for the CPM environment for communication over a telephone line. KERMIT was designed to be as machine independent as possible with the primary aim of mainframe to micro direct communication. You just don't find the X-modem protocol implemented on DEC-20, IBM-360, CRAY's etc whereas the KERMIT protocol has been implemented on almost any machine that you care to name. As usual it's horses for courses and for binary file transfer between like micros, X-modem is more efficient. - ED).

5-1/4 DISKS ON AMSTRADS.

Anyone contemplating using an old 5 1/4 disk drive on an existing Amstrad 3 inch system should be aware of the fact that the Amstrad controller requires to see a 'drive ready' signal on pin 34 of the disk drive. Many older disk drives do not produce this signal.

The problem can be overcome by a 'bodge' on the drive. The easy way is simply to tie pin 34 on the drive permanently to ground with a wire link. Under this condition the drive is always ready and the controller may need to have two or three tries before it gets a good read from the disk. This will normally only be apparent to the user as a slight delay in response at the first disk access.

A better way is to AND together the 'drive select' signal (pin 12 for a second drive) with the 'motor on' signal on pin 16. This can be done using an open collector 74LS32 piggy-backed on top of one of the existing drive chips. The output of the AND gate is taken to pin 34 on the drive. (Check first that pin 34 is not connected in the drive).

Neither solution is perfect but in most cases it will get the system working on the 5inch drive. I have done the AND gate mod on a couple of drives for Amstrads and they work OK.

Perhaps the ideal solution would be to delay the 'ready' signal a few hundred milliseconds to allow the drive to get up to speed but I feel it is probably not worth the extra complication.

EXTRA MEMORY ON THE MODEL 4.

Many Model 4 users will have read the review of the Alpha Technology Supermem board in 80 Micro. This allows up to about a megabyte of banked ram to be added to the model 4. In a subsequent issue Roy Soltoff of Misosys offered the patches to TRSDOS 6.2 to allow the operating system to take advantage of the extra memory.

I have obtained a copy of these patches from MISOSYS and as Roy stated they may be published in club magazines etc. I am including them here for interested readers. I received the patches in printout form and hope I have not made any slips in copying them in. If you think I have please let me know.

Roy Barber, 44 Dallin Road, Bexleyheath, Kent DA6 8EJ. 01 304 3856.

Patches in patch file format follow courtesy of Roy Soltoff:

.BOOTAT/FIX -07/18/85

.Copyright 1985 Roy Soltoff, All rights reserved

.Patch to BOOT/SYS (TRSDOS 6.2) for Alpha Tech board

.Apply via: PATCH BOOT/SYS.LSIDOS BOOTAT (O=N)

.
.Patch low memory pointer at 206H
D00,06=19 10

.
.Patch byte I/O handler at 65EH
D04,5E=88
.Patch byte I/O handler at 67DH
D04,7D=88

.
.Patch KCK@ routine at 7F6H
D05,F6=2F
.Patch KCK@ routine at 7F9H
D05,F9=81
.The following patch line is temporary !!!!
D05,F8=57

.
.Patch ENADIS DO RAM and @BANK routines at 817H
D06,17=CD 4A 08 21 03 0C 39 30 09 E1 ED 73 74 08 31 40
D06,27=03 E5 21 55 08 E3 E5 21 7F 08 23 C5 0E 43 ED 40
D06,37=23 70 47 3A 78 00 30 02 E6 7F 77 E6 FC F6 82 ED
D06,47=41 18 1C F3 22 7C 08 F5 E1 22 77 08 AF C9 CD 4A
D06,57=08 2A 2F 08 7E CB 7F CB FF 2B C5 0E 43 46 2B 22
D06,67=2F 08 32 78 00 D3 84 ED 41 C1 20 03 31 00 00 21
D06,77=00 00 E5 F1 21 00 00 FB C9 00 00 00 00 00 00
D06,87=00 E6 7F FE 1F 30 15 05 FA CB 08 0E 86 28 1D 0E
D06,97=46 05 28 18 05 28 0E 05 3A 02 02 C8 C3 ED 0D E5
D06,A7=21 14 04 18 0B CD B3 08 C0 78 0E C6 E5 21 10 04
D06,B7=CD 10 10 78 E6 07 07 07 B1 32 C8 08 AF 3E 08
D06,C7=CB 46 E1 C9 E5 21 05 80 39 E1 38 D0 C5 0E 46 CD
D06,D7=A6 08 78 C1 20 C6 CD F4 0F E6 7F 32 02 02 A9 B0
D06,E7=4F CB 79 06 00 C8 E3 BF C9

.
.Patch video driver at 0C8FH
D0A,8F=2F

.
.Change pointer in FDC driver to account for patch at 0E3FH
D0C,3F=18 10
.Add code at end of FDC driver for @BANK and TASKER at 0FF4H
D0D,F4=B7 28 01 3C D3 43 3A 02 02 47 79 C9 21 77 00 CB
D0E,04=F6 21 02 02 C9 32 02 02 21 77 00 C9 47 0F 0F 0F
D0E,14=E6 03 85 6F C9
.EOP

.SYS1AT/FIX - 07/17 85
.Copyright 1985 Roy Soltoff, All rights reserved
. Patch to SYS1/SYS (TRSDOS 6.2) for Alpha Tech board
.Apply via: PATCH SYS1/SYS.LSIDOS SYS1AT (O=N).

.
.Patch cleanup routine at 1E77H and 1E7AH
D00,9C=7F
D00,9F=2F

```
.
.Patch restoral of bank 0 at 1E0H
D00,E5=88
.EOP

.SYSOAT/FIX - 09/10/85
.Copyright 1985 Roy Soltoff, All rights reserved
.Patch to SYSO/SYS (TRSDOS 6.2) for Alpha Tech board
.Apply via: Patch SYSO/SYS.LSIDOS SYSOAT (O=N)
.
.Patch the SVC table at 1CCH
D08,6E=88
.
.Patch the disk I/O handler at 19FDH
D09,5B=88
.
.Patch the TASKER at 1C06H
D08,8C=CD 00 10 7E F5 DB 43 F5 AF 77 D3 43
.Patch the TASKER at 1C3DH
D0B,C3=D3 43 F1 CD 09 10 CB B6
.Patch the TASKER at 1C91H
D0C,17=F1 E3
.
.Patch system initialisation routines at 1E3FH
D0C,CD=D3 84 26 FF 56 01 43 1F 71 C5 48 0D 06 03 3E 66
D0C,DD=EF C1 ED 41 5E 34 00 00 00 7E BB 00 00 00 73 28
D0C,ED=09 C5 48 0D 06 01 3E 66 EF C1 10 DD CD 47 21
.
.Patch system initialisation routines at 2147H
D0F,E1=ED 41 72 6C 22 0E 04 22 1C 00 21 10 04 CB 86 11
D0F,F1=14 04 0E 04 ED B0 3A 6F 00 C9
.EOP

(Thats it ! What does the (O=N) parameter do ? , Roy Barber).
```


MIX - A veratile editor

Mix is a general purpose text editor - plus. It has all the usual editing functions expected of any good word processor plus:-
control key commands which can be configured to your personal taste,
split screen editing and ability to edit two files at the same time, optional line numbering, and auto line numbering
auto-indent ,
user defined macros
- and an almost free-ware price tag.

It is available for both CPM and MS DOS and we have the CPM version for our 4P.

Before using Mix for the first time it must be configured for the terminal it is to be used with. For Tandy users this just involves selecting the appropriate number (for Montezuma or Tandy's CPM) from a long list of options presented by the configuration programme, SETEDIT. This automatically puts the correct screen codes into a setup .TXT file which can be saved as an Ascii or binary file. The binary file is used by Mix to configure itself when loaded. The .TXT file is the one you use to change key assignments and macros.

Mix comes set up with Wordstar control codes but the authors appreciate that these are not universally loved and changing them is simply a matter of including TK (translate Key) commands in the Setup.TXT file. Defining macros to load automatically with the editor is also done by including appropriate MK commands in the setup file. The TK and MK command can be used while editing too, to produce temporary key assignments and macros. As part of the documentation MIX demonstrates how to create quite complex macros, which can use up to 10 integer variables so that, for example, you can specify the height and width of a box when entering the command to draw it. The examples include macros for drawing lines and boxes, changing case, moving text between windows and automatic line numbering for BASIC

Do you use more than one programming language? You can have a MIX setup file for each. For example we have a PASCAL.EDT file which produces PROCEDURE from @P, BEGIN for @B etc and a dBASE.EDT which writes DO WHILE .NOT. EOF when @D is pressed.

The MIX programme files are actually named EDIT (.COM and .OVL) so to use MIX enter EDIT and it loads and configures itself with the default setup file SETUP.EDT and opens a buffer for text entry. To edit an existing file enter EDIT filename and to use a non-default setup file enter EDIT (filename), your-setup-filename

MIX starts with only one text buffer, to give maximum memory for text entry. Opening a second 'window' opens another buffer into which you can enter text, load another file, or another part of the file in buffer 1. To view the buffer contents you can either toggle between two full screens or see both together by splitting the screen horizontally or vertically. Macros allow scrolling and paging of the non-active screen and let you move text easily

between them. Obviously with two active buffers the amount of text which can be in memory will not be large but MIX pages easily through files, writing the output to temporary disk files which are renamed on exiting, leaving the original as backup. You can also go directly to any part of the file which is not in the buffer by using its line number.

Like many word processors MIX has EDIT and COMMAND modes. Most of the things you are likely to want to do can be done from either. In EDIT mode the editor responds to control key sequences, like Wordstar (except that you can decide which commands to assign to which keys). In command mode the commands are fixed but mnemonic two key sequences. Although any available command can be mapped to control keys the key assignments take up memory and things that are not done very often, like disk I/O or switching line numbering on and off, are probably better left as COMMAND mode commands.

Mix comes with a set of Help files for Wordstar control codes but making up your own version is easy - Mix appears to use the public domain CPM HELP system and the help file is just a text file. This is what ours looks like and it gives some idea of the range of functions available.

ESC(BREAK) - Command Mode

␣ - cancel CM

(␣CC - hold CTRL for both chars <>CC - command mode)

```
+-----+
!␣QA - append from file      !<>HE - help                !
!␣PP - page through large file!<>SF filename - show file    !
!␣QQ - repeat last command   !␣QD - DIR                    !
!␣QU - restore line as was    !␣QS - status                   !
!<>LN - show lines numbered tog!<>AL/AUTO - auto line number  !
+-----+
<>MK "key" "full $/commands" - define macro for string or series
of CMs
```

```
+-----+
!␣V - insert toggle          !␣<- (␣G) - deletes char under !
!<>IR - auto CR line insert   !␣DW - delete word              !
!␣N - insert line            !␣DL /␣Y - delete line          !
!␣QN - insert nn lines       !␣DE - delete to e o line       !
!␣QO - open line             !␣DF - delete file              !
!␣QJ - join line             !                                  !
!␣DU - dupl line above cp     !␣U - undelete line (at cp)     !
!␣PP - page through large file!␣QW - undelete word           !
+-----+
```

```

+-----+-----+
!EQP - prev cursor position      !!EO - other window      !
!                                !!                                !
!<- -> (EH EL) - char move      !!<>SS r c - split at r/c    !
!v e arrows (eJ eK) - line move!!<>SS 0 0 - rejoin screen  !
!Sh <- (eA) - start of line    !!eSH - split horizontally   !
!Sh -> (eF) - end of line      !!eSV - split vertically     !
!Sh v e arrows(eC/eR) pg up/dn !!                                !
!eSe->/e<- - scroll -><-      !!eS e arrows - scroll alt w'dow!
!eES/F/G/B- end of screen file +-----+
!eTS/F/G/B- top global file/blk!!TL Xfer line (eQQ 1st if many)
!ePP - page through large file !eTP Xfer para (between blanks)
!<>PU PD - para up down        !
!<>WR WL - word R L            ! <>CD - chg dsk (rst) not ramdisk
!<>WU WD - window up down      ! eW - Write (n lines)
!<>PO - goto row col           +-----+
!<>GO nnnn - go to line no     !!eSeE save & empty buff,no exit!
! (10000 - eof, buffer empty) !!eSeX save & exit
! (-1 -eof, text remains)     !!eQeC cancel - no exit
!<>+/- - scroll dn/up nn lines !!eX - Exit no save eSA - save !
+-----+-----+

```

```

+-----+-----+
!eBS - block start mark        !!eQF - find string
!eBE - block end mark          !!eQG - global find/replace
!eBC - block copy              !!eQR - replace with query
!eBM - block move              !!eQ->- next string (find)
!eBD - block delete           +-----+
!eBH - hide block             +-----+
!eBO - output (hold) block     !!eQI - auto indent
!eBI - read (unhold)output blk!!eI - tab ->
!eBP - print marked blk        !!
!eBW - write marked blk to file!!<>TD - delete tabs
!eBR - read blk (fil/start/end)!!<>TS/TC - set/clear tab at cp
!eBU/eBL - Upp/Low case block !!<>TB n(1,n2,n3) - set tab
!                                !! every n/at n1,n2,n3
!<>FI - fill block             +-----+
!<>JU - fill with justify      ! <>CN - centre line
+-----+-----+ <>CLU/CLL - change to U/L case

```

<>BC - blank compression toggle <>VS/HS - change scroll defaults
 <>SV N "value" -assigns value to \$ var N (0-9)

MACROS etc.

Parameters -: INTs - separate with blanks
 STRING - " "s or not if single BUT all except last
 in " "s if multiple
 un "'d chars literal BUT "#nn" is Hex, "=cc" is CM,
 "ec" is CTRL
 to use literally in quoted \$ use as ##,==,#5E

TRANSKEY (TK) allocates a single CM or char to key(s)
 TK "eX" "=QT" TK "!" "#7C" (! prints !)

MACKEY (MK) allocates a string or series of CMs to key(s)
MK "@P" message string OR just MK & answer prompts
CM lists start with "=CM" and end with =NL"
OR must spec each param separated by =NL
MACNAME (MN) as MK but to <>CM
MN "CM" "=commands"
SETVAR (SV) assigns value to string var 0-9 (eg SV 0 "11")
used as CMs - "=0 in macros
GETVAR (GV) gets value into var from keyboard with prompt
"=GV5=NL' prompt '=NL=.....

Mix is a large programme, by CPM standards - 22 K and a 27 K Overlay file. Like its Wordstar parent its many functions involve a lot of overlay activity and this can be a bit slow unless you have 128 K and run the programme from the Ramdisk. It is no more unacceptable than any other overlay based programme though and with a Ramdisk the 'disk accesses' are of course not noticeable.

In summary, Mix does everything that anyone is likely to want from a text editor and at #40 (from Grey Matter or \$29 direct from the US) it will probably tie with Turbo Pascal as our software Bset Buy of this year.

Ariela Taylor

(© represents the control key in the above text - Ed)

KERMIT PART III.

Last month we had a look at the design features of the ingredient at the heart of the communication protocol - The Packet. This month we will look in detail at the nitty-gritty of packet structure.

KERMIT packets have the general format:

```
+-----+-----+-----+-----+-----+-----+
! MARK ! char(LEN) ! char(SEQ) ! TYPE ! DATA ! CHECK !
+-----+-----+-----+-----+-----+-----+
```

where all fields consist of ASCII characters, and the char function converts a number in the range 0-94 to a printable ASCII character by adding 32 (20H).

The fields of a KERMIT packet are as follows:

MARK -- Start-of-packet character, normally SOH (ASCII 1).

LEN -- The number of ASCII characters, including quoting characters and the checksum, within the packet that follow this field, in other words the packet length minus two. Since this number is expressed as a single character via the char function, packet character counts of 0 to 94 are permitted, and 96 is the maximum total packet length.

SEQ -- The packet sequence number, modulo 64. The sequence numbers "wraps around" to 0 after each group of 64 packets.

TYPE -- The packet type, a single printable ASCII character, one of the following:

- D Data
- Y Acknowledge (ACK)
- N Negative Acknowledge (NAK)
- S Send Initiate (Send-Init)
- R Receive Initiate
- B Break Transmission (EOT)
- F File Header
- Z End of file (EOF)
- E Error
- G Generic Command. A single character in the DATA field, requesting host independent remote execution of the specified command. The single data character may be :
 - L Logout, Bye.
 - F Finish, but don't logout.
 - D Directory query.
 - U Disk usage query.
 - E Erase file.
 - T Type file.
 - Q Query server status.
- C Host Command.
- X Text display header.

DATA -- The "contents" of the packet. If contents are required in the given type of packet, they are interpreted according to the packet type. Nonprintable ASCII characters are prefixed with quote characters and then "uncontrollified". Characters with the 8th bit set may also be prefixed, and a repeated character can be prefixed by a count. A prefixed sequence may not be broken across packets.

CHECK -- The block check sequence, based on all the characters in the packet between, but not including, the mark and the check itself, one, two, or three characters in length as described last month, each character transformed by char. Normally, the single-character checksum is used.

The packet may be followed by any line terminator required by the host, carriage return (ASCII 15) by default. Line terminators are not part of the packet, and are not included in the count or checksum. Terminators are not necessary to the protocol, and are invisible to it, as are any characters that may appear between packets. If a host cannot do single character input from a TTY line, then a terminator will be required for that host.

Here are four KERMIT data packets:

```
©AE"D No celestial body has required J
©AE#Das much labor for the study of its#
©AE$D#M#Jmotion as the moon. Since ClaA
©AE%Dirault (1747), who indicated a way7
©AE&D of#M#Jconstructing a theory conta5
```

The "©A" represents the SOH (or Control-A) character. In the final packet shown, "E" is the length. The ASCII value of the "E" character is 69, less 32 (the unchar transformation, which is the opposite of char) gives a length of 37. The next character tells the packet sequence number, in this case 6 ("&" is ASCII 38). The next is the packet type "D" for Data. The next characters, " of#M#Jconstructing a theory conta", form the data; note the prefixed carriage return and line feed. The final character, "5" is the checksum, which represents the number 21: ie. the character "5" has an ASCII value of 53 which when "unchar'd" = 21 (all numbers in this paragraph are in decimal).

Effects of Packet Corruption

What are the consequences of transmission errors in the various fields? If the SOH is garbled, the packet will be treated as inter-packet garbage, and lost. If any other character within the packet is garbled into SOH, the current packet will be discarded, and a new (spurious) packet detected. If the length is garbled into a smaller number, then a character from the data field will be misinterpreted as the checksum; if larger, then the program will probably become stuck trying to input characters that will not be sent until one side or the

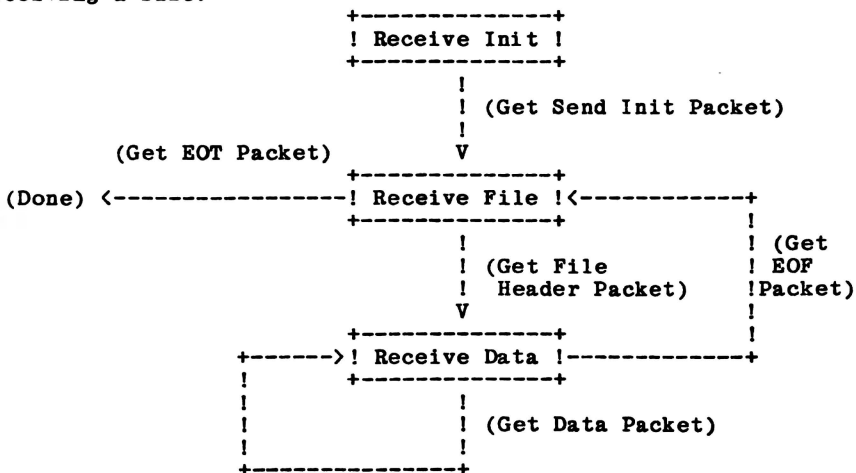
other times out and retransmits. If the sequence number, type, any of the data characters, or the checksum itself is garbled, the checksum should be wrong. If characters are lost, there will most likely be a timeout. If noise characters are spontaneously generated, they will be ignored if they are between packets, or will cause the wrong character to be interpreted as the checksum if they come during packet transmission.

Most kinds of errors are caught by the checksum comparison, and are handled by immediate retransmission. Timeouts are more costly because the line sits idle for the timeout period. The packet design minimizes the necessity for timeouts due to packet corruption: the only fields that can be corrupted to cause a timeout are the SOH and the packet length, and the latter only half the time. Lost characters, however, can produce the same effect (as they would with a fixed-length block protocol). Had a distinguished end-of-packet character been used rather than a length field, then there would be a timeout every time it was corrupted. It is always better to retransmit immediately than to time out.

Operation of the Protocol

The KERMIT protocol can be described as a set of states and a set of transitions that define, for a given event, what action to take and what new state to change to. The inherent simplicity of the design, particularly the requirement that each packet must be acknowledged, reduces the number of states and actions, and the amount of state information that must be maintained, to a minimum.

Here is a simplified version of a state diagram for KERMIT receiving a file:



For simplicity, some transitions are not shown in the diagram:

- If in any state a bad packet is received or a timeout occurs, a null transition back to the same state occurs, with a NAK for the expected packet.

- In any state an error may occur that can cause the transfer to terminate. For instance, the target disk might fill up. The side that encountered the error sends an error packet, containing an informative error message, and quits. Upon receipt of the error packet, the other side displays the message on the screen (if it is in control of the screen) and also quits.

- Actions that are taken on each transition, such as opening a file when a File Header packet is received, are not shown; in particular each packet successfully received is ACK'd.

The receiver starts out in Receive Init state and waits for the other side to send a Send-Init packet. If any other kind of packet is received, or the Send-Init does not arrive within the timeout interval, a NAK is sent. Timeouts or NAKs can occur up to a threshold which, when exceeded for a particular packet, causes the protocol to assume that the connection has become unusable, and to give up. After the Send-Init arrives, the state becomes Receive File; KERMIT waits for a File Header packet containing the name of the file which is to come. When the file header arrives, KERMIT opens a new file using the name provided (perhaps transformed to suit local naming conventions, or to avoid a name collision), and switches to Receive Data state. KERMIT then receives the contents of the file, until an EOF (End Of File) packet arrives. At that point KERMIT switches back to Receive File state. If another file is to be sent, another File Header packet will follow, otherwise an EOT (End Of Transmission) packet will terminate the transfer. The distinction between EOF and EOT, plus the File Header itself, allows files to be sent in groups. EOF marks the end of a file, EOT marks the end of a group. This distinction also allows the two sides to disconnect cleanly: the EOF must be ACK'd before the sender will believe the file has been transmitted correctly; the EOT will follow, but if the ACK which is sent in response is lost, no harm will be done since both sides are terminating anyway.

The state transitions for a sending KERMIT are similar. In each state, instead of waiting for particular packet types, KERMIT sends the appropriate packet and waits for an ACK. If the ACK does not arrive within the allotted time, or a NAK appears instead of an ACK, the same packet is retransmitted. A send operation begins with a Send-Init packet, includes one or more files, each starting with a File Header, followed by one or more data packets, followed by EOF. When all the specified files have been sent, an EOT packet closes the connection and terminates the operation.

direction of transmission.

PADC -- The control character I need for padding, if any, XOR'd with 40H to make it printable. You respond in kind. Normally NUL (ASCII 0), some systems use DEL (ASCII 177). This field is ignored if the value NPAD is zero.

EOL -- The character I need to terminate an incoming packet, if any. You respond in kind. Most systems that require a line terminator for terminal input accept carriage return for this purpose. (Can you see the Catch-22 here?)

QCTL -- The printable ASCII character I will use to quote control characters and prefix characters, normally "#". You respond with the one you will use.

QBIN -- The printable ASCII character I want to use to quote characters which have the 8th bit set, for transmitting binary files when one or both systems cannot use the parity bit for data. Since this kind of quoting increases both processor and transmission overhead, it is normally to be avoided.

CHKT -- Check Type, the method for detecting errors. "1" for single-character checksum (the normal method), "2" for two-character checksum, "3" for three-character CRC-CCITT. If your response agrees, the designated method will be used; otherwise the single-character checksum will be used. Other check types may also be added.

REPT -- The prefix character I will use to indicate a repeated character. This can be any printable character other than blank (which denotes no repeat count prefix), but "ç" is recommended. If you don't respond identically, repeat counts will not be done. Groups of 4 or more identical characters may be transmitted more efficiently using a repeat count, though an individual implementation may wish to set a higher threshold.

CAPAS -- An extendable bit mask encoded printably, to indicate whether certain advanced capabilities, such as file-attribute packets, are supported.

RESERVED FIELDS. The next four fields are reserved for future use. Sites wishing to add their own parameters to the initial connection exchange should start at the fifth field after the capability mask in order to remain compatible with other KERMIT programs.

Naturally, the three prefix characters must be distinct and should be chosen to be uncommonly used printable characters, to minimize further overhead from having to prefix them when they are found in the data.

Trailing fields within the DATA field may be omitted, in which case they will assume appropriate defaults. Defaults for intermediate fields can be elected by setting those fields to blank. Every parameter has an appropriate default, and in fact

the entire data field of the Send Init packet or its ACK may be left empty to accept all defaults. The more exotic parameters are at the end, and reflect more recent developments in the KERMIT protocol; earlier implementations can still communicate with newer ones, since there will not be agreement to use these options. The Send-Init mechanism preserves compatibility from the very earliest KERMIT to the very newest.

There is no protracted negotiation; everything must be settled in a single exchange. Some parameters, however, are outside the scope of this exchange and must be set even before the very first packet is sent. For instance, if the receiving computer can only read characters with odd parity but the sending computer sends them with even parity, the Send-Init packet will never arrive successfully. In cases like this, the user may have to issue some preliminary commands to inform one or both KERMITs about the vagaries of the other system. Another example is the packet terminator (EOL) mentioned above -if the receiving KERMIT requires one that the sending KERMIT doesn't know about, the Send-Init will never get through. For these reasons, most implementations of KERMIT provide SET commands for all the parameters listed above, and some others as well.

Next month should see us through to the end of this series (Thank God I hear some cry !) where we will look at some of the rules and heuristics (inductive reasoning from past experience) that Kermit uses to increase the efficiency of transfer and look at the user interface (A trendy phrase for command set).

Members Advertisements

TANDY MODEL 1 COMPUTER. with 48K expansion interface
Extras include keyboard cover, disk drive, RS232, cassette leads
and recorder, books and manuals + many programs.

£350 o.n.o.

Peter Cockayne
1 St Francis Rd.
Whitchurch
Cardiff CF4 1AW.

1 tape based VIDEO GENIE with many tapes (mainly games) £30

1 disc based GENIE 1 (twin disc drive, RS232 many
discs with Tandy compatible software plus DESK £190

1 MODEM 300 baud, originate only with telephone £35

1 MODEM Acoustic, 300 baud originate and answer £35

Original Software.

Multidos 1.7 for Model 1 Double density £25

Dotwriter with many fonts £20

Copycat - copying utility £10

Duplidisk v2.3 £ 5

Modem 80 £ 5

The above is offered O.N.O. - buyer collects

Don Harrison,
24 Gifford Gardens
Hanwell

London W7 3AN Tel. 01-575-2670