



SYDNEY TRS-80 USERS GROUP NEWSLETTER

P.O. BOX 297, PADSTOW 2211.

Volume 9 Issue 4 DECEMBER 1988

CONTENTS

| TITLE | AUTHOR | PAGE |
|--|-----------------|------|
| SCRIBE'S SCRIBBLINGS | Brian KEEGAN | 38 |
| COMPUTING IN THE MIDDLE KINGDOM | Ivan KENNEDY | 38 |
| A Novelty Item | David SUTTON | 39 |
| Treasurer's Report for October 1988 | Gordon SYMONDS | 39 |
| Exchange Newsletters | | 40 |
| FOR SALE | | 40 |
| Buying, Using and Optimizing a Hard Disk | Brian KEEGAN | 41 |
| Helpful Hints | | 41 |
| Public Domain Disks | David SUTTON | 42 |
| More On LESCRIPT Dot-Graphics | Rod STEVENSON | 44 |
| LESCRIPT Revisited | Rod STEVENSON | 44 |
| File Handling in BASIC | Andrew MARSHALL | 44 |
| LS-DOS Corner | Gary K. BRYCE | 46 |
| Help Wanted Department | | 47 |
| Floppy Drive Bid Form | | 47 |
| What's This BOOLEAN Business ? - Part 3 | Max MORRIS | 48 |
| Signed Numbers | Bill ALLEN | 48 |

MEETING ARRANGEMENTS

* Meetings will be held on Saturday afternoons

* commencing at 1:00 P.M. as follows:-

* 10th of December at the 1st Sefton Scout Hall

* 2 Waldron Road, SEFTON

* There will be a Christmas Cake supplied by the Group

* Members are asked to bring along some extra eats etc

* 17th December in the API Rooms, First floor

* Endeavour House, 7 Smith Street, PARRAMATTA

* NOTE: THERE WILL BE NO MEETING ON 24th December

SYDTRUG Bulletin Board

CLUB-80 Bulletin Board operates for members, seven days week twenty-four (24) hours a day on (02) 332-2494. We are now using a NetComm 1234 as "intelligent" MODEM, and the following data formats are available :-

CCITT V21 (300/300), V22 (1200/1200),
V23 (1200/75) and V22 bis (2400/2400).
BELL 103 (300 FULL Duplex), BELL 212 (1200/1200)
2400 (2400 FULL Duplex)

All formats utilize 8 DATA Bits, 1 STOP Bit and NO Parity

You should set your Modem and/or software for "ORIGINATE" except for V23 (1200/75) which should be set for VIATEL or 1200 Receive/75 Transmit

Limited access is granted for visitors. Articles for publication should be left in the News Room of CLUB-80 for collection by the Editor.

WHO'S WHO

| | | |
|-----------------------|----------------|----------|
| President | Gary BRYCE | 628-5058 |
| Vice President | Peter WIGNELL | 759-8024 |
| Secretary | Brian KEEGAN | 646-3448 |
| Treasurer | Gordon SYMONDS | 744-1901 |
| Club 80 Sysop | Michael COOPER | 331-7136 |
| Hardware Co-ordinator | Errol ROSSER | 769-7646 |
| Newsletter Editor | John MERCER | 579-2915 |

Newsletter Closing Dates

Hard Copy only - 10th December 1988 -

Via Bulletin Board - 17th December 1988 -

BANKCARD and MASTERCARD

We have the facility to charge your membership fees, or renew fees to either MASTERCARD or BANKCARD. Additionally, purchases made on your behalf by the club may also be charged to your credit card. If you wish to use this service, please quote your card number, type of card, expiry date of card, and SIGN your request.

DISCLAIMER

No Patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this publication, the Sydney TRS-80 Users Group or its appointed office bearers assume no responsibility for errors or omissions. Neither is any liability assumed for damages arising from the use of any information contained herein. Any opinions expressed are those of the author concerned, and not necessarily those of the Group or its committee.

The contents of this publication are (c) 1988 by SYDNEY TRS-80 USERS GROUP. All rights reserved. Enquiries should be directed to the "Secretary, Sydney TRS-80 Users Group, P.O. Box 297, PADSTOW N.S.W., AUSTRALIA 2211. Material appearing in this publication may be reprinted in similar computer club newsletters and nonprofit publications if accompanied by the following notice:

Reprinted from SYDTRUG NEWS, P.O. Box 297, PADSTOW 2211.

SCRIBER'S SCRIBBLINGS

by Brian KEEGAN

A very busy month has just passed, for both private and group business. Sorry I haven't seen you at the meetings during November.

Well, the best news yet !! We have FINALLY got your new membership BADGES, that is, those for our present membership year. Thanks very much Darrell for persevering with DOTWRITER and producing the masters. We then had them printed on card and laminated, just wait and see the COLOUR. Then, Dennis PAGETT and I spent about four and a half hours the other Friday night, cutting them up and making the hole in each badge for the little doodad that clips onto your pocket. My hands still ache from the cutting and punching. Next year I think we will either look at another way to secure them to your persons OR acquire the correct punch to do the job.

I sent replies to about twenty enquiries this month. Some were requests regarding new membership, others about the availability of software or manuals, and still some more for technical information. Following is a list of items which someone out there could possibly help me with, so I can pass them along:

Mod III version of Maxi Manager (Complete package)
Mod III version of VISICALC (Complete package)
Mod III machines preferably disk drive version
Info on program "Master Handicapper" re dog/horse racing
Info on small hand held dedicated betting computer called a "Tote Buster"
Doco for Dick Smith "General Ledger" (Mod 4)

If you can assist with the above there are a number of members you would be grateful.

On the subject of letters and requests, if you write in seeking information or services PLEASE do not forget, it costs the group to reply to you if only in postage and stationary. It would help keep running costs down if you send a stamped self addressed envelope with your enquiry, or if requesting such items as disks or say back issues of newsletters please send in some funds to cover postage and packing. It can be in the form of stamps if you wish, we will happily refund any reasonable excess. A number of the dispatches this month involved quite some expense e.g. floppy disk, disk mailer and postage easily amounting to four dollars. In fact, in these particular instances I actually asked the recipients to return some funds to defray the costs.

Although I haven't been to the meetings this month, I have been busy on the group's behalf. I have followed up and cajoled a couple of members who are getting out of certain areas of computing to, in one case donate, and in the other case to agree to pursue the arrangement of an auction, of excess software and manuals.

As a matter of interest, who would be interested in an auction (trash or treasure sale ??) in the near future. The group would obviously appreciate the donation of a percentage of the sales and anything you don't sell would be gratefully received into our library.

I am also following up some overseas sources of software, mostly public domain, for the group.

It is good to see the meetings, especially at Sefton, being better attended, in fact they are becoming quite interesting of late, KEEP IT UP.

We have had a number of comments in recent times from members who work or for personal reasons cannot attend our Saturday afternoon meetings. We are actively following this up, there may be a chance of a mid-week evening meeting. Anyone interested please let your thoughts be known to the committee. To date we have had no luck with a venue in the eastern suburbs, a few leads came to dead ends.

See you later...BLK

Worth Repeating

Not very long ago it would have been regarded as shocking to find girls in their teens READING the kind of books they're now WRITING.

-- Robert Clurman

COMPUTING IN THE MIDDLE KINGDOM

by Ivan KENNEDY

Last month (September) I spent three memorable weeks in China. This happened because someone, perhaps mistakenly, thought I was an expert on acid formation in ecosystems (acid soil and acid rain) that can come about because we burn too much coal with too much sulphur in it, or farm our soils using ammonium-based fertilisers too much and so on. It is true that I can sometimes write chemical equations that balance for charge and mass and many of you will remember trying to write such equations in high school. Like computing, it's usually simple enough once you get the hang of it.

Whatever the reasons for my good fortune, the trip to China was truly an experience of a lifetime. Despite its low per capita income, China is an immensely rich country - in human resources, culturally and in physical resources. The sight from my CAAC airliner of the sheet of glistening water moving through rivers, canals and rice fields across the country from west to east brought home to me really for the first time how it is China can so readily sustain one billion people. To an Australian, familiar with our arid interior in a country merely moistened at the edges, this was an amazing sight!

Now that China has done away with the communes and introduced private enterprise in agriculture (full family property rights, with inheritance), farm production has surged ahead and has apparently taught the USSR a lesson, since I notice Gorbachev announced last week that Russia is about to attempt to reorganise its agriculture along similar lines. I wonder if the long border between China and Russia and the thought of a China with the productivity of Japan (based on free enterprise) and ten times the population has more to do with the recent changes in Russia than interaction with the West. My three weeks in China taught me not to discount the possibility of such economic growth in China and we in Australia should sit up and take notice. While there, the news that our next communication satellites would very likely go up riding Chinese Long March rockets was announced.

Anyway, my trip to China meant about a month away from my 4P. Definite problems of keyboard withdrawal loomed! However, I had craftily made certain plans. One was to purchase an NEC laptop equivalent to a Model 100 in passing through Hong Kong. And was to buy some 256K RAM chips so that I could complete my "All the Way to 320K" project (see 80 Micro, October '87, page 60) for a reasonable outlay. All the other components required to make up two circuit boards had come to about \$40, but the eight chips at \$12-15 each in Sydney seemed to add too much. I also had a brief from David Heyes to keep my eyes open for certain Model I items.

As it turned out, the 'Spirit of Australia', Qantas, let me down. In Hong Kong JetSet had booked me into the Regal Riverside Hotel in the New Territories. I expected to have four hours' shopping time to go by taxi to the nearest computer stores and make my selection. Arriving in Hong Kong two hours late dashed my well-laid plans (not so well-laid, it seems - isn't there something about 'mice and men' that could go here?). All I managed to buy was a Philips Model rechargeable shaver at the local Newtown shopping plaza where I travelled by shuttle bus from my hotel. At least this 'state of the art' appliance that I acquired with MasterCard for \$110 (so very easy to use and no hidden charges, just the current exchange rate) has been a good buy. It warmed my mercenary heart to see in K-Mart in Chatswood recently that these cost \$250+ here. MasterCard worked just as well for purchases in Beijing and Shanghai.

While in China itself, I kept my eyes open for things computing, in between giving lectures and discussions at the various research institutes - and of course (since you won't believe otherwise) banqueting and sightseeing. In Beijing my entry into a computer shop revealed that a stock IBM XT 8088 costs about Y60,000, or \$20,000. Since this would be forty years' salary for the average university professor (who earns much less in China, I might add, than your average taxi driver or free enterprise farmer), clearly home computers are not the going thing in China. I then understood much better why Mei Chi Wang, a Chinese colleague, who worked with me in Sydney about two years ago, was so keen to take back a very basic Commodore to her teenage son when she returned to Shanghai. As for 256K RAM chips, I didn't even ask.

In fact, the oldest computer of all, the abacus, is still almost totally in evidence in the shops of China. The speed of use of these is truly amazing (faster than keying on a calculator), apparently because of practical short-cuts, and I was shown on one occasion that they can be used to play games on. Electronic computers are being applied on a fairly wide scale however, but not an Apple in sight. There was no evidence of local manufacturing of computers yet that I could see, but the laboratories I visited had IBMs. A

romanised (latin characters) version of Chinese called Pinyin has been introduced, that all Chinese kids learn in school. This is useful for computing, since a qwerty keyboard can be used to write in Pinyin Chinese, and the software (WORDSTAR, running under ChineseCharacter-DOS) can convert this directly to Chinese characters. In another reform on the mainland, Chinese is written from left to right as we do and down the page. Several times on television, I even saw computer screens with text in two columns, both in Pinyin and Chinese characters. There are no Chinese books written in Pinyin though, which might be easier for westerners to handle. They also use other means of handling the problem of Chinese characters.

In Nanjing, the only computer-related aspect of my trip concerned a meeting with Geoff Colwell, from Canberra, who was there to provide training in statistical analysis with computers.

The third main city I visited was Shanghai. I didn't get to any computer shops there, but had lunch as a guest of the husband of my friend Mei Chi Wang, who is director of the East China institute of computer technology. Their son has recently enrolled in computer science at university having completed his HSC equivalent. Much of the work of the institute seemed to be related to mainframe applications, however, and the people's computer is plainly still some time away.

So there it is. Great prospects for developments re computers and computing in China, but early days as yet. One hundred million 4Ps could really set them going. A million Radio Shacks needed, all computers transported by bicycle - Tandy, don't miss another chance.

Always the optimist, I made another attempt to shop in Hong Kong on returning to Sydney. This time I was thwarted by the long queue for taxis at the airport. The thought of missing my connection to Qantas was too much so I bought a gift for my wife (WIFE/CMD to paraphrase our past vice-president, Jimbo) to add to several others purchased in China and a book for myself ("Life and Death in Shanghai", by Nien Cheng). The latter was a good antidote for any too-uncritical attitude I might have developed towards China and its recent past, describing as it does the terrible experiences of a woman who was imprisoned during the insane period of the Cultural Revolution led by the 'Gang of Four' (and Mao Tse-tung?) and whose beautiful daughter (born in Canberra in 1943) was murdered by the Red Guards.

Failure is never easy to take, but at least I made it back to my 4P keyboard before permanent, irreversible damage occurred.

By the way, does anyone have eight 256K ram chips for sale, so I can still go all the way to 320K by Christmas - or even a Tandy 100 or equivalent as a consolation????? But cheap. Please!!

A Novelty Item

by David SUTTON

Although not really anything to do with computers, the following trivia passed across my desk the other day. You know, one of those 'works of wisdom' that seem to originate from nowhere, everybody tries to do it, and then it disappears again, lost forever. It is not a knowledge test, purely a creativity test which will require some lateral thinking.

The following sentences are well known except the words have been replaced by letters. Question 1 has been completed for you.

Good luck, answers next issue.

- Q1. 16 = O in the P 16 Ounces in the pound
 Q2. 7 = W of the W
 Q3. 9 = P in the S S
 Q4. 90 = D in a R A
 Q5. \$200 = P for P G in M
 Q6. 26 = L of the A
 Q7. 6 = S on the A F
 Q8. 32 = D F at which W F
 Q9. 85 = N of P K's on a P
 Q10. 12 = S of the Z
 Q11. 1001 = S of the A N
 Q12. 54 = C in a P (with J's)
 Q13. 18 = H on a G C

Treasurer's Report for October 1988

by Gordon SYMONDS
INCOME:

| | Current Month | Year to Date |
|--------------------------|------------------|-----------------|
| Members' Subscriptions: | | |
| New: Joining Fees | 10.00 | 60.00 |
| Club | 18.00 | 118.00 |
| Renew: Rejoin | | |
| Club | 20.00 | 1030.00 |
| B.B.S. | 10.00 | 300.00 |
| | 50.00 | |
| Members Purchases: | | |
| P/D Disks | 83.00 | 103.00 |
| Blank Disks | 100.00 | 725.50 |
| Subs/Publications | | 45.00 |
| Ribbon Reinking | 11.00 | 21.00 |
| Photocopying | 1.20 | 13.00 |
| Provision for Postage | 1.00 | 8.00 |
| | 196.20 | |
| Sale Surplus Consumables | | 10.00 |
| Sale Donated Items | | 100.00 |
| Capital Realisation: | | |
| Modem Ex-BBS | | 100.00 |
| Hard Disk Ex-BBS | | 100.00 |
| | 254.20 | 2735.00 |
| Cash Management A/c: | | |
| Interest | | 110.50 |
| | 254.20 | 2845.00 |

EXPENSES:

| | Current Month | Year to Date |
|----------------------------|------------------|-----------------|
| Newsletter Costs | | |
| Printing | 125.00 | 755.00 |
| Postage | 2.40 | 299.50 |
| Sundry: | | |
| Printer Ribbons | | 45.20 |
| Paper | 52.00 | 52.00 |
| | 179.40 | |
| B.B.S. Costs: | | |
| Telephone Rental | | 33.50 |
| Committee Expenses | | 30.00 |
| Meeting Rental: | 100.00 | 250.00 |
| Consumables | 124.00 | 297.70 |
| Bank & Government Charges | 8.89 | 37.00 |
| Purchases for Members: | | |
| Blank Disks | 400.00 | 1300.00 |
| Publications Subscriptions | | 219.80 |
| | 400.00 | |
| Capital Expenses: | | |
| Mysosis Disk Notes | | 12.40 |
| | 812.29 | 3315.00 |

Worth Repeating

A real test of maturity is the ability to remain equally unruffled when the elevator boy calls you "Pop" and the senior partner calls you "Son".
-- Ivern Boye

Science is immeasurably ahead of nature. For example, in the modern household the children are about the only things left that still have to be washed by hand.
-- Bill Vaughn

If I were reincarnated, I'd want to come back a buzzard: nothing hates him or envies him or wants him or needs him; he is never bothered in danger, and he can eat anything.
-- William Faulkner

There's nothing wrong with the average person that a good psychiatrist can't exaggerate.
-- "Toronto Star"

Twenty million young women rose to their feet with the cry, "We will not be dictated to," and proceeded to become stenographers.
-- G.K. Chesterton

Exchange Newsletters

Some of what is included in our library

June 1988

"Adelaide Micro User News"

RS-232 Connections Made Easy - Some useful information about communications, not only RS-232: Four Function Calculator (3) - A thinly disguised excuse for writing about Z-80 assembly language: More On Lescrypt Dot-Graphics - A follow up to his previous review. This includes some demonstrations of the capabilities when using a printer with pin addressable capabilities: DOS PLUS's FILE/DVR Feature - This appears to be similar to Misosys's DiskDISK utility for use with LDOS/LS-DOS.

"LLIST", Newsletter of the Calgary Color Computer Club

CoCo-3 DOS Upgrade - A BASIC program to change some of the "defaults" in RSDOS

"Canberra Micro-80"

PCplay - Layout of Directory and FAT: Busted Printheads - A saga of problems with replacing a CITH print head

"Computer News 80", P.O. Box 680, CASPER WYOMING 82602-0680

Chips and Who Left the Gate Open - Computer News 80's experiences in expanding memory on a number of Model 4s: PD Reviews - MODSCRIP, an enhancer for Disc Scripsit; Utility 4, five programs for the Model 4: Hints, Tips & Visits with your Fellow Readers - A potpourri of readers contributions: The Disk Maze - Views on disk quality, guarantees etc.: A Visit With David Goblen - Passwords, Non-modifying Model I disk reads, TRSDOS 1.3 and the XLR8er, and More patches, 80 MICRO: Open Forum - A regular Question and Answer session covering a wide range of topics

"GEMS", Newsletter of the Greensbro Eighty Microcomputer Society

Tidbits From Here and There - The title says it all: A Giant Leap from a Model I to a PC Compatible - The final installment of the two part series on file transfer

"HAWTUG NEWS" Newsletter of the Hawaii TRS-80 User Group

All the News That's Fit to Print - This and that: Dieting Under Stress - Humorous item

"Thuggery", Newsletter of The Hobart Users Group, G.P.O Box 1271N, Hobart Tasmania

QuickBASIC 4.0 - A review of MicroSoft's BASIC Compiler: Bert's Ramblings 27H - A reprint from the "Australian Accountant": MS-DOS Message - A look at MS-DOS re-direction and piping facilities: Fractals - Part I - A somewhat esoteric mathematical concept: Tandy THOR-CD - From Tandy User Group Newsletter (USA) May 1988

"NATGUG News", newsletter of the National Amstrad, Tandy & General User Group

PD Software, Trojan Horses & Viruses - Information on SOME known examples: Getting the Most Out of the XLR8er - Running ProWAM, DDUTY and RamDISK concurrently: More MS-DOS - A follow-up to 'Simple, Fussy or Messy' in April issue: Fido Networks - Some interesting information about this rapidly growing international system of electronic mail: BCPL - A brief overview of this programming language: ADE: A Drive Emulator - A review of this discontinued line. DiskDISK is the latest one: Batch Files - Some commands that relate to BATCh files

"National Capital Tandy Computer Users Group"

Generic Cadd - A review of one of the lower priced "Vector Graphics" programs for MS/PC-DOS machines: Simple Laptop/MS-DOS Transfer System - A method which requires no additional communications program, just a null modem and RS-232 cable: Salvaging Damaged Floppy Disks - A method of recovering data from disks with damaged envelopes: Living With DOS: Hard Disk Primer - A wealth of information mainly for MS/PC-DOS users: Parking "IT" May Save Users Some Problems - About parking the head on a hard drive: DOS 2.X COMMAND.COM Reloads From Specified Drive - For MS/PC-DOS users: Changing Dates of Existing Files - For MS/PC-DOS users: Learn to Play - An exhortation to discover more about your machine and its capabilities. Although it is directed at MS-DOS users it can be applied in large part to Z-80 machines: Quick Sub-Directory Listing - Mainly for MS/PC-DOS users with hard drives: "Abort,

Retry, Ignore" Requires Caution - Another example of stupidity on the part of the MS/PC-DOS designers: Are You Sure - A timely warning for MS/PC-DOS users: The Lighter Side of DOS - About how to eliminate a lot (if not all) of the dead wood from MS/PC-DOS: Undocumented Lotus 1-2-3 Release 2 Features - Very short and sweet: Lotus 1-2-3 Tips - About saving current cell locations and identifying worksheets: Personal Ancestral File - A very favourable review of a Genealogical program distributed by the Church of Latter Day Saints in USA: Reassigning Your Keyboard - For MS/PC-DOS users

"The Interface", Newsletter of The San Gabriel Valley Tandy Users Group

Disk Caching vs Ram-Disk - Some interesting information: Dos Utility Tips - Assumably MS-DOS: Double Draw - Using ProDraw and TrsDraw with DoubleDuty: PFS:First Publisher - A brief overview

"T-BUG", Newsletter of the Tandy Business Users Group

SCO XENIX Printer Redirection - If you use XENIX this may be of interest

"Bits & Bytes", Newsletter of the TRS-80 System 80 Computer Group, June 1988

Solutions For Date Stamping TRSDOS 6.1 - A plug for 6.3: How to Convert a Model I Expansion Unit to Work on a System 80 - A chronicle of challenges: Testing Your Machine's Ports - A SHORT BASIC PROGRAM: Parallel Printer Switch - A very inexpensive example: Disk Recovery - A novel if questionable approach: JCL Fi for Model 4 Under TRSDOS - A very simple short introductory HDFMTAPP/CMD Fixed For 15 Meg Drives - A patch for the Newdos program: Change Case With Lazywriter - A plea for compatibility

"WNYTUG News", Newsletter of Western New York Tandy Users Group

Inside DOS - Part 9 - MS-DOS's ASSIGN command revisited

"Winnipeg Micro-80 Users Group Newsletter"

Modifying the Multipack Interface for Use With the CoCo 3 - How to overcome the incompatibility between multipack Interfaces and the CoCo 3: Computers and Safety - Investigating the effects (if any) radiation and other aspects of long hours spent at the VDU: New Expansion Plans For DeskMate - An announcement from Tandy

FOR SALE

Tandy Model 3 RS-232 Boards - Complete

Catalogue Number - 26-1148

Price \$20 plus shipping charges (if applicable).

For further information contact the committee

COMPUTER TABLE

Adjustable height, approx 5' X 3' laminated top with steel legs. A nice solid piece of office furniture.

Price \$120

Phone DENNIS after 6pm on :- (02) 772-4848

Two only NEC Spinwriters 7715, with Cut Sheet Feeder

Price \$500 each

Also One Original Microbee computer - ROM needs replacing (C \$125)

Price AS IS \$80

See Dave HEYES

Buying, Using and Optimizing a Hard Disk

by Brian KEEGAN

[Based on an article appearing in "Dynamic Memories", Oct.'88 by Lewis PERDUE

Note:- The computer industry, like most other industries, is developing rather rapidly these days. With apologies to the author, the information presented in some parts of the original article, while probably correct when the article was written, would appear to be a bit dated at this point in time. I have applied journalistic licence in updating a few parts to reflect more accurately the situation, especially here in Australia. BLK]

Not too long ago, most PC users were content with their dual-floppy systems and felt that hard disks were not only too expensive, but were also an unnecessary frill with few valid uses for the average user.

No longer. Many of today's powerful software packages are contained on four, five, six or even up to ten floppy disks. Many of these programs, such as Desktop Publishing software, simply have no provision to work on anything other than a hard disk. And even those that do allow for floppy use, require the user to change disks so often that users find themselves spending more time changing disks than working the application. Woe be unto you and your children's children if you put the wrong one of the six disks in the machine. A system crash with loss of all data is frequently the result.

And even if you are an octopus with lightning fast tentacles and unerring accuracy, lack of storage will clobber you the day you have built the worlds best Lotus spreadsheet that amounts to say 500K bytes. Ask yourself, how are you going to save the dang thing? NOT, certainly to your 360K floppy disk drive. Kiss that spreadsheet goodbye.

Likewise, a mailing list or other database too large to fit on a floppy will require you to do multiple searches for every disk the database is contained on.

In addition to simplifying the use of software and offering greater storage capacity, hard disks access and store data much faster than a floppy drive. Using a hard drive can speed up your computer's operations two to five times.

Today, it is possible to purchase hard drives for personal computers with storage capacities ranging from five megabytes to a gigabyte (1000 megabytes). And the cost is no longer prohibitive. A 20 megabyte winchester drive with controller card from a reputable vendor should cost between \$300 and \$400 with larger capacities costing proportionately more.

[NOTE:- In Australia they are just starting to approach these prices, especially for the outdated standard drives which are being replaced with the newer RLL technology. BLK]

Practically all the hard drives used in personal computers are of the Winchester type and for that reason this article will be based on this type of drive (rigid disks are also the same animal). Be aware that this is not necessarily so in the field of mainframes and mini-computers. Hard drives for PC's come in a variety of physical sizes, the most common being the 5.25" half height drive. The 5.25" is the size of the platter or storage surface, however it is being fast replaced by units with 3.5" platters which have made the drive-on-a-card concept possible. Originally the 3.5" platter drives were more expensive than the 5.25" variety, but they are now more the rule than the exception.

A hard disk is useless without a bus adapter board, known as a controller card, which plugs into one of your PC's expansion sockets. Controller cards usually support many different hard disk platter and head configurations. You must therefore make sure that the controller card is suitable for the drive you intend to control (most times they come as a matched set), AND that your PC and the operating system you are using will support the controller card/drive configuration you have purchased. From memory (mine not the PC's), DOS version's earlier than 3.2 do not support drives over 30 Meg. That is, you must partition or divide them into a number of drives, each less than 30 Meg, during the format process (more about this process later).

If you have one of the older machines, with its slots filled with memory, serial, printer and video cards, OR a new machine full of EMS, EGA, mouse etc. cards you might consider replacing your floppy drive controller card with a combination hard disk/floppy controller card. An added benefit is that these combination cards usually also

support the newer 30 track and 3.5" drives on the floppy side (things. Be careful here though, as a lot of machines use a multifunction card which has the floppy disk controller on it. If this is so and you wish to utilise a combination hard/floppy disk controller you will have to remove the multifunction card, and replace all the other functions with a new function card which is not fitted with floppy disk controller. In general, you cannot have two cards in machine supporting a single function, as the machine won't know which card to output the information for the peripheral.

If this is your first attempt at expanding your machine, save yourself a lot of hassle and purchase the hard drive with a matching controller card from the same vendor, and leave the rest of the card as is. This advice is based on past experience, in that if you carry out a number of modifications together, it is very hard to determine WHICH modification is causing you trouble. Also, if you have any troubles getting the hard drive formatted, the drive vendor will blame the card and vice versa. Of course, it's NEVER their merchandise that's dud.

In reality, hard drives and their controller cards are these days very reliable, and so long as you don't do silly things with your DC (e.g. try to reformat the drive AFTER you start using it), OR bounce the machine case while the drive is running, you should have a long and pleasant relationship with your drive. I have had a SEAGATE 2 Meg in my machine for about 4 years now, which apart from a dud controller card a couple of years ago has given me no problems at all. The controller cards are now so cheap that I keep a SPARE controller card so at any sign of trouble I can initially try replacing the card before going off the deep end trying to recover that data which haven't backed up for a week or so.

POWER UPGRADE MAY BE REQUIRED (PC, XT and clones)

Most of the later IBM compatibles come fitted with a 150W PSU along with extra power connectors, so it is no great problem to run hard disk. However, problems can occur when you want to fit a hard drive to an earlier IBM clone or if you are one of those rare people with an early model IBM machine. These early machines usually can't be fitted with a 65 watt PSU. This is really only enough power to drive a basic motherboard and a couple of drives. If this is the case for your machine then also purchase a replacement 150 watt PSU for around 100-150 dollars either local or mail order. It will usually come with all the power connectors you require. They usually just clip into the bottom of the case and a couple of screws through the back of the case.

Another problem encountered by those wishing to upgrade, especially an earlier machine, is that the BIOS will not support such things as memory upgrades, EGA cards OR hard drives. You need a BIOS date stamped later than 10/27/82, that's Yankee for 27th Oct '82. To find out load up DEBUG, when it comes up with the prompt ":", type in D F000:FFF0 and hit enter.

On the screen should be a series of numbers, embedded in which should be a recognizable six digit code indicating the BIOS date.

Next month I'll continue this article with some further information on installing the drive and configuring it...BLK

Helpful Hints

David HAYES has provided the following tips for any of our readers who may need to replace missing feet on the Model I Keyboard and/or Expansion Interface.

There are two alternative feet available to suit the Model I. Part no. 64-2365 is a clear foot, and part no. 64-2346 is a black foot. These should be available at your local Tandy store, and the price is about \$2-99 for a pack of one dozen feet. These should also fit the System 80.

For the System 80 Expansion Interface, try your local hardware store and ask for the washer that goes into a cistern float valve. You would be advised to take with you the measurements of the old feet in case there is more than one size available. We understand that Enmore Building Supplies should be able to supply these if you are that area.

Public Domain Disks

by David SUTTON

Here are the balance of the public domain software referred to last month as coming from Warwick Sands of NEWDOS 86 fame.

One file of particular importance is called ND86/ZAP and contains several zaps that will need to be applied to NEWDOS86.

Most of the file and extensions are self-explanatory, however a few conventions do exist and need explaining :-

/MD1 A /CMD file suitable for the Model 1 only
 /MD3 A /CMD file suitable for the Model 3/4/4P only
 /CMDA /CMD file suitable for any Model running NewDos86
 /N80 A /CMD file suitable for NEWDOS80 only
 /N86 A /CMD file suitable for NEWDOS86 only
 Both /N80 and /N86 will run on all models.

Any files with an asterisk (*) at the front indicates that the file is a PSFile and should have the attribute set with the command:-

ATTRIB,filespec,PSF=Y

This needs to be done whenever a single file copy occurs.
 A COPY;x,y,, automatically carries the flag across.
 A COPY,filespec,to,filespec may not.

Files associated with TYPE/CMD

TYPE/ILF Extraction file
 TYPE/CMD An enhanced PRINT routine for ASCII files
 TYPE/TXT
 TYPE/TECH/TXT

Printer Driver programs for TYPE/CMD
 BROTHER1/ASM for an EPSON look-alike
 BROTHER1/DVR
 BROTHER2/ASM for an IBM look-alike
 BROTHER2/DVR

COPAL/ASM is for a printer of unknown origins but very close to
 COPAL/DVR an EPSON

DWP410/ASM for the TANDY DWP-410
 DWP410/DVR

GP100/ASM is a Dot matrix printer (Seikosha) distributed by
 GP100/DVR Dick Smith

ITOH8510/ASM is for the ITOH8510 also known as the PROWRITER
 ITOH8510/DVR

ITOHF10/ASM is for a Daisy Wheel printer once sold by Dick Smith
 ITOHF10/DVR

LPV/ASM is for the old TANDY LINE PRINTER V
 LPV/DVR

LPVIII/ASM is for the TANDY LINE PRINTER VIII
 LPVIII/DVR

STAR/ASM is for an early Model Star Dot Matrix printer
 STAR/DVR

TOSH1350/ASM is for a TOSHIBA 1350 24 pin printer with a new
 TOSH1350/DVR FONT ROM installed.

Additional files for TYPE/CMD
 PRINT5/ASM Reference file for TYPE/CMD
 LZ35/ASM Subsidiary File for Printer Drivers
 VECTORS/ASM A list of addresses within TYPE/CMD required to
 create your own printer drivers.

Files not associated with TYPE/CMD

Reference files used by nearly all NewDos86 Assembly routines
 MACRO/LIB Library files required for Assembly of NewDos86
 MODEL1/ASM routines
 MODELIII/ASM
 PARAM/ASM

NewDos86 Utility files. BASIC and Assembler

CAPTURE/ASH A BASIC utility to capture text sent to the screen
 CAPTURE/BAS into a string array
 CAPTURE/TXT
 *CAPTURE/CMD

CREATE/BAS Warts and all BASIC program to create /PDFfiles.

*DEVICE/MD1 Installs high-memory devices Model 1
 *DEVICE/MD3 Installs high-memory devices Model 3/4/4P
 DEVICE/TXT

FILTER/ASH Routine to provide installable filters for
 *FILTER/MD1 NewDos86.
 *FILTER/MD3
 FILTER/TXT
 TRAP/ASM A sample filter program
 TRAP/FLT

LOCATE/ASH A BASIC utility that displays the current cursor
 LOCATE/TXT location
 LOCATE2/ASM
 *LOCATE/CMD

LOGON/ASM An Automatic Disk Format Utility for NewDos86
 LOGON/TXT
 *LOGON/MD1
 *LOGON/MD3

LOOK/BAS Displays contents of /PDFfiles

MAKELIB/ASM A routine for creating multiple files
 MAKELIB/TXT
 MAKELIB/MD1
 MAKELIB/MD3
 PSF/ASM A demonstration of how to reload PSFiles in the
 PSF/TXT DOS overlay area

ND86/ZAP A list of zaps required for NewDos86

NEWDATE/ASM NewDos86-LDOS-TRSDOS date stamping conversion
 NEWDATE/TXT program
 NEWDATE/MD1
 NEWDATE/MD3

ORDER/BAS Sorts a file of EQUates

PAGE/ASM Pause Printer at end of page
 PAGE/TXT
 *PAGE/CMD

PLOTMATH/BAS Creates shapes for #PLOT

PRINTER/ASM Program to create a printer initialisation program
 PRINTER/N80
 PRINTER/N86 The file that this program produces needs PSF
 PRINTER/TXT attribute

RAMDISK/ASM Installs a RAM driver for the Models 4/4P under
 RAMDISK/MSG NewDos86
 RAMDISK/TXT
 RAM/MD3

SCHEDULE/ASC A demo program using INPUT TO USING
 SCHEDULE/BAS
 STUDENT/DAT

SLEEP/ASM Put the computer to sleep
 *SLEEP/CMD

SORTDIR/ASM Physically sort the directory
 SORTDIR/MD1
 SORTDIR/MD3
 SORTDIR/TXT

TABIT/ASM A file processor that converts spaces to tabs
 TABIT/CMD and vice-versa.
 TABIT/TXT

UTILITY/BAS A BASIC program to repair the GAT
 UTILITY/TXT

*UNKILL/MD1 The unkill files
 *UNKILL/MD3
 UNKILL/TXT

VID/ASM An 80x24 Video driver for Models 4/4P
 VID/TXT
 VID1/ASM

VID2/ASM
 *VID/MD3
 VIDUTIL/ASM BASIC files to access the screen in 80x24

ZAPVID/BAS A utility program to append a small routine to /CMD files to enable/disable 80x24 mode.

PDUTIL51 : Utilities

README/TXT General info on programmes on this disk
 DFLIST/CMD Displays killed files if still intact
 DFLIST/DOC Doco for DFLIST/CMD
 DFLIST/SRC Source for DFLIST/CMD
 DIRSORT/CM Directory sort utility
 DIRSORT/DOC Doco for DIRSORT/CMD
 DIRSORT/SRC Source for DIRSORT/CMD
 DIS/CMD Disk directory utility
 DIS/DOC Doco for DIS/CMD
 DIS/SRC Source for DIS/CMD
 RAMSPOOL/CMD Print spooler
 RAMSPOOL/DOC Doco for RAMSPOOL/CMD
 RAMSPOOL/SRC Source for RAMSPOOL/CMD
 SYS/CMD Load System files to memory
 SYS/SRC Source for SYS/CMD
 UNKILL/CMD Restores killed files if still intact
 UNKILL/DOC Doco for UNKILL/CMD
 UNKILL/SRC Source for UNKILL/CMD

PDUTIL52 : Utilities

ROMFIX/CMD Fix ROM for fast boot
 ROMFIX/ASM Source for ROMFIX/CMD
 BOOTROM/ASM Source for fast boot M4
 CAT/CMD Categorize disk space
 CAT/DOC Doco for CAT/CMD
 CAT/SRC Source for CAT/CMD
 HEX/CMD Binary-Asc, Hex-Binary
 SAVE/CM Key to disk text processor

PDUTIL53 : Utilities NEEDS PDUTIL54....NEWDOS86 ONLY

TYPE/CMD Enhanced print routine for ASCII files
 TYPE/TXT Doco for TYPE/CMD
 TYPE/TECH/TXT Doco to help create your own drivers
 LZ35/ASM Source subsidiary file for print drivers
 VECTORS/ASM Addresses in TYPE/CMD for custom drivers
 PRINT5/ASM Source reference file for TYPE/CMD

PDUTIL54 : Utilities NEEDS PDUTIL53....NEWDOS86 ONLY

Print drivers for TYPE/CMD
 BROTHER1/ASM - EPSON
 BROTHER1/DVR
 BROTHER2/ASM - IBM
 BROTHER2/DVR
 LPV/ASM - TANDY LP5
 LPV/DVR
 COPAL/ASM - CLOSE EPSON
 COPAL/DVR
 TOSH1350/ASM - TOSHIBA 1350
 TOSH1350/DVR
 LPVIII/ASM - TANDY LP8
 LPVIII/DVR
 DWP410/ASM - TANDY DWP-410
 DWP410/DVR
 GP100/ASM - SEIKOSHA
 GP100/DVR
 ITOH8510/ASM - ITOH 8510 PROWRITER
 ITOH8510/DVR
 ITOHF10/ASM - DICK SMITH DW
 ITOHF10/DVR
 STAR/ASM - EARLY MODEL STAR
 STAR/DVR

PDUTIL55 : Utilities NEWDOS86 ONLY

LOOK/BAS Displays contents of /PSFiles
 PSF/ASM How to load PSFiles in DOS overlay area
 PSF/TXT Doco for PSF/ASM
 MAKELIB/ASM Creates multiple files - source
 MAKELIB/MD1 Creates multiple files - M1
 MAKELIB/MD3 Creates multiple files - M3
 MAKELIB/TXT Doco for MAKELIB/MD1 /MD3
 ND86/ZAP Zaps required for NEWDOS86
 NEWDATE/ASM Source for LDOS-TRSDOS date stamping conversion
 NEWDATE/MD1 Date stamping conversion - M1
 NEWDATE/MD3 Date stamping conversion - M3

NEWDATE/TXT Doco for NEWDATE/MD1 /MD3

PDUTIL56 : Utilities

PRINTER/ASM Source - creates printer initialisation
 PRINTER/N86 N86 only - creates printer initialisation
 PRINTER/N80 N80 only - creates printer initialisation
 PRINTER/TXT Doco for PRINTER/N86 /N80
 TABIT/CMD Converts file spaces to tabs and vice-versa
 TABIT/ASM Source for TABIT/CMD
 TABIT/TXT Doco for TABIT/CMD
 SORTDIR/ASM Source - physically sort a directory
 SORTDIR/MD1 M1 only - physically sort a directory
 SORTDIR/MD3 M3 only - physically sort a directory
 SORTDIR/TXT Doco for SORTDIR/MD1 /MD3

PDUTIL57 : Utilities

RAM/MD3 Installs RAM driver
 RAMDISK/ASM for M4/M4P under
 RAMDISK/MSG NEWDOS86
 RAMDISK/TXT

UTILITY/BAS Repairs GAT
 UTILITY/TXT Doco for UTILITY/BAS

VID/ASM An 80x24
 VID/MD3 Video
 VID/TXT Driver for
 VID1/ASM M4/M4P
 VID2/ASM
 VIDUTIL/ASM
 ZAPVID/BAS

PDUTIL58 : Utilities

TRAP/ASM Sample ASM filter programme
 TRAP/FLT Sample filter programme
 FILTER/ASM Source for FILTER/MD1 /MD3
 FILTER/MD1 Install filter routine - M1
 FILTER/MD3 Install filter routine - M3
 FILTER/TXT Doco for FILTER/MD1 /MD3
 SCHEDULE/ASC Demo
 SCHEDULE/BAS Demo
 STUDENT/DAT Demo
 PLOTMATH/BAS Creates shapes for #PLOT
 LOGON/ASM Source for LOGON/MD1 /MD3
 LOGON/MD1 Automatic disk format utility - M1
 LOGON/MD3 Automatic disk format utility - M3
 LOGON/TXT Doco for LOGON/MD1 /MD3

PDUTIL59 : Utilities

MACRO/LIB library file for assembly of ND86 routines
 MODEL/ASM library file for assembly of ND86 routines - M1
 MODELIII/ASM library file for assembly of ND86 routines - M3
 PARAM/ASM library file for assembly of ND86 routines
 CAPTURE/ASM Source - capture screen output to string array
 CAPTURE/BAS BASIC - capture screen output to string array
 CAPTURE/CMD - capture screen output to string array
 CAPTURE/TXT Doco for CAPTURE/CMD
 CREATE/BAS Creates /PDFiles
 DEVICE/MD1 Installs high memory devices - M1
 DEVICE/MD3 Installs high memory devices - M3
 DEVICE/TXT Doco for DEVICE/MD1 /MD3
 UNKILL/MD1 Unkill files - M1
 UNKILL/MD3 Unkill files - M3
 UNKILL/TXT Doco for UNKILL/MD1 /MD3
 LOCATE/ASM Displays current cursor location
 LOCATE/CMD Displays current cursor location
 LOCATE2/ASM Displays current cursor location
 LOCATE/TXT Doco for LOCATE/CMD
 SLEEP/ASM Puts computer to sleep - source
 SLEEP/CMD Puts computer to sleep
 ORDER/BAS Sorts a file of EQUATES
 PAGE/ASM Pause printer at end of page - source
 PAGE/CMD Pause printer at end of page
 PAGE/TXT Doco for PAGE/CMD

PDCOMM08 : Communications

COM300/CMD Terminal programme 300 Baud
 COM1200/CMD Terminal programme 1200 Baud
 CINTV11/ASM Initialise routine
 CXMODV21/ASM XMODEM routine
 COMHV731/ASM Source for COM300/CMD COM1200/CMD

More On LESCRIPT Dot-Graphics

by Rod STEVENSON

[Reprinted from "Adelaide Micro User News", G.P.O. Box 214, ADELAIDE S.A. June 1988]

Last issue I enthused on the extensive capabilities of LESCRIPT; among them the capability of printing dot graphics directly from the text. Unfortunately Ed had to remove my examples because of printer incompatibility. For the more common printer functions of subscripting, superscripting, italicising, bold facing, underlining, and character pitch there is not a problem: LESCRIPT provides a positively huge range of printer drivers to overcome the incompatibilities between printers requiring different control codes. And probably this is what is important, since I imagine few 'real' users will want to 'fiddle' as I do!

The actual incompatibility that was the problem was a difference in codes for the printer-ROM characters in my Gemini and Ed's Epson. In this case it didn't really matter in showing that it is possible to send such characters from within the text - the character sent is defined by its own code appropriate to the printer in use, and is assigned to one of the <shift clear> keys. On my Gemini printer all the codes from 128 to 255 are used for special characters, so it is a simple matter to include such symbols within the text wherever they are required.

Even if the printer doesn't have the character you require, LESCRIPT allows one to be printed in the dot graphics mode. Because I don't much like the printer-ROM pound sterling symbol, I designed my own. Although decimal numbers are probably more correct in metric Australia, at times I find I'd like to be able to use the vulgar fractions of one quarter, half and three quarters, so I created them. Perhaps there is a need to print the smiling and glum face characters in the Model 4 character generator ROM? Or even totally individual constructions such as my signature!

To produce the pound sterling character requires a printer code of 1B4C08007E8012804204, and while some may find such planning and construction tedious, I enjoy it! Briefly, after the first four codes, it's just a matter of adding the hex values of the vertical dots required for each column of the design. Being a trial and error situation makes it easier.

The <shift clear> keys can also be redefined to be printer control codes such as tab (09), or backspace (08), or even as 00 which will take up space on the printed output but will allow the screen line to be spaced with the allowance for a printed character wider than the single redefined screen character.

LESCRIPT Revisited

Rod STEVENSON

[Reprinted from "Adelaide Micro User News", G.P.O. Box 214, ADELAIDE S.A. August 1988]

A few issues ago I enthused on the features offered by LESCRIPT word processor. Now that I've been using it for 'real' I thought it appropriate to give a progress report on my adventures.

The following is a complete list of the bugs I've found so far, since this list is small I still consider LESCRIPT to be worthy of commendation. So this is by no means a list of complaints, merely an unbalanced view of a user's actual experiences. All the previous ones I made about being a worthy successor to LESCRIPT still apply. Then, I recognise that I'm by no means widely experienced with word processors. Nor do I wish to look further: from talking to others (and here is one of the great benefits of such a group as ours) I realise that the only other word processor worth trying is ALLWRITE, perhaps we can have a piece on that from a user? Members do use word processors I know, but from the comments I hear, they put up with deficiencies that I'm not willing to accept as necessary. So how do we have an ALLWRITE devotee willing to expose himself?

My greatest concern with the bugs I've encountered in LESCRIPT is the inconsistency of their operation. For if there is a deficiency in a program that will always produce a predictable problem, I'm more happy to work within that limitation, knowing what are the risks inherent. My worry is that most of the following are not.

The major area of concern is the formatting to print in columns. With straight text printed in columns there is no problem that I've discovered. But introduce blank lines, printer controls, columns of uneven width or columns of less than 30 characters, and there are problems of lost text and double characters printing at the start of the next column. The chaining facility works well in allowing a document greater than memory, and it keeps track of header and footer format commands from the first file; but if printing in columns it loses text and inserts blank lines. Not to think that columnar printing is impossible though: this issue was produced on LESCRIPT, although the front top section was printed separately and stuck on.

In the previous episode I mentioned the slowness of some functions, such as the controls to go to the bottom or top of text. As well as being slow, I've found that these sometimes mess up the formatting on the screen. I don't know if it's only on the screen or in memory that the text is altered: it happens when using tabs to columnise tables of data or clauses in a document. And it is erratic.

The facility for opening a 'window' to type in more text is also erratic in that it sometimes will overwrite previous text. At first I thought this was if I opened the window too far down the screen, or attempted to insert too much new text, but I've since found there's more to it than that.

Although LESCRIPT uses the second bank of memory on the Model 4 to allow 90K of text, when it gets to more than 70K the message "memory full" appears when attempting some functions. At this point the only workable way is to start a new file. This I see, not so much as a bug as an unexpected happening, since it's only reasonable to expect the various functions to require some space in which to operate; particularly if they have to manipulate a large amount of text.

I've found, at times, there's a problem of losing keyboard characters. I've timed my typing to be only 40 wpm, but I suppose I might do bursts at a somewhat faster speed. Even so, I consider this is a real problem if LESCRIPT is intended to be used by professional typists in a work situation. When I go out of LESCRIPT back to DOS I find the clock speed has been changed from 4 to 2 MHz on my Model 4, so I assume that this is what LESCRIPT operates at, and may be the reason for losing characters. Although that's not at all certain, since it may just be messing about with the ports that control the clock on its return to DOS.

So that's all the problems I've found in actually using LESCRIPT, and I must admit that most of them are not a worry for 'normal' usage. I pass them on for possible interest to others. And again ask for something on ALLWRITE for these pages.

File Handling in BASIC

by Andrew MARSHALL

[Reprinted from "Thuggery", newsletter of The Hobart Users Group, G.P.O. Box 1271N, HOBART TASMANIA 7001, July 1988]

At recent meetings of THUG we have been discussing aspects of the BASIC programming language. This article is part 1 of a repeat of some of the points on BASIC file handling for those who were unable to make it to the meeting.

Why Use Disk Data Files?

The use of disk files for data storage is going to increase the complexity of any programs written, and for some applications they are not justified and can be replaced with direct input each time the program is used or with READ and DATA statements. However, in most cases disk files will improve efficiency and accuracy, by requiring data to be keyed only once, and will allow a much greater amount of data to be used as the computer's memory is no longer a limit. Finally, use of disk files makes your data independent of the program so that the same data is readily available for other programs to use. [Sydtrug Editor's Note: Also the program will be able to use other data without changing the data statements.]

Types of Files

Most versions of BASIC offer at least two main types of files, namely sequential and random access files. The first of these types, sequential, is a type of data file where data is stored in the order it is entered, and can only be accessed in this way. It can be simply visualised as a long sheet of paper on which all data is printed as it is input, and which must then be read back only by going to the beginning and reading all data in the same order. On the other hand, a random file is best visualised as a series of cards, all the same size, where each separate data record is written to a new card and

any card can be retrieved at any time. (These will be covered in detail in a future article.)

Sequential Files

As stated above, sequential files can be visualised as a long printed list of the data which is typed back automatically in the same order as it is printed. In BASIC, this analogy can be carried further as the statements used to put data in a sequential file and get it back out are the same as those used at your terminal, namely PRINT, INPUT and LINEINPUT STATEMENTS. When used with data files, these statements retain the same formatting requirements and traps as when used with the screen or keyboard.

First, consider the PRINT statement. The following statement will print three numbers and two strings on the one line as shown:

```
PRINT A,X$,F4,Q1,B$
```

```
100 HE LLO 25.1 200 TH,ERE
```

but if we were to type the same data from the keyboard in response to an INPUT statement we would type the following:

```
?100,"HE LLO",25.1,200,"TH,ERE"
```

If the commas were missing, or if the strings did not have quotes at each end, or if the data was input in a different order we would get an error of some sort. With data files, the same constraints apply except that one or more spaces will be accepted as separators between data items instead of commas. Therefore, some care must be taken when printing to and inputting from files to ensure the data is stored and read back correctly. To use a sequential file, the first thing that must be done is to open it. This is done by use of the OPEN statement. Before issue the OPEN command, two things must be decided upon - the name of the file we are going to open and which file buffer to use. The file name can be any legal file name and can include a disk identifier, but care must be taken that it is not the same as one already in use, as opening it for use as a sequential file will erase all data in it. The file buffer is a number from 1 to 15 which specifies to BASIC which area of memory it should use to store data temporarily before writing it to the disk and after reading it in. These areas are set aside by BASIC when it is first entered, but normally not all 15 are set up. The default for most BASIC versions is to set up only three such areas unless it is told a different number when first run. If three areas are set up, only numbers from 1 to 3 can be used as the buffer number. Once these points are decided, the file can be opened as follows:

```
OPEN "O",n,"filename"
```

where n is the buffer number.

Returning to the analogy of the file as a list on paper, this command is the same as moving to the top of a new blank sheet of paper ready to write the data, or erasing all data from an existing sheet and returning to the top. If the file does not already exist it is created by this command, or if it does exist all data is erased from it.

To put data onto the file a new type of print statement is used:

```
PRINT#n,data
```

where n is the buffer number previously used to open the file and data is any list of variables or expressions that would normally be used with the print statement. For example:

```
PRINT#1,A,X$,F4,Q1,B$
```

would print the data previously listed onto a file opened using buffer 1. Data can be printed to the file using any combination of PRINT# statements and will be formatted in the file in the same manner as it would be formatted on the screen. Thus a comma between items printed will separate it into 16 character spaces, and a semicolon will separate items by only one space (or none for strings). Also, data stored in an array in memory can be printed to the file by use of a simple loop:

```
FOR I = 1 TO 15
PRINT#1,X(I)
NEXT I
```

It is important to remember that, when retrieved, the data will come in the same order as it is put on to the file.

Once all data is printed to the file there is a final command that must be issued to ensure that it is saved and will be able to be accessed in future, namely the CLOSE command. This tells BASIC

that a file is no longer in use and causes any data still in the buff to be written to the disk, and a mark to be written to the file show where it ends. It also causes the directory to be updated with details of the file size and storage location on the disk. The CLOSE command can be issued as a command to close one file by specifying the buffer number (e.g. CLOSE 1) or to close all open files by putting no number (e.g. CLOSE).

A short program to store a list of names in a disk file using the commands listed above might be written as follows:

```
10 OPEN "O",1,"NAMES/DAT:0"
20 PRINT"TYPE THE NEXT NAME OR END TO FINISH"
30 INPUT NA$
40 IF NA$="END" THEN GOTO 70
50 PRINT#1,NA$
60 GOTO 20
70 CLOSE 1
80 END
```

Now that we have stored our data in a file, we need some way to get it back out so it can be used. We cannot use the same OPEN "O" command for this as that command erases all data from the file as opens it. The command in BASIC to open an existing file so that the data can be read is OPEN "I". This command opens an existing file and positions BASIC to the beginning of it to read back the data stored there. If the file does not exist BASIC will give an error when this command is executed. The format of the command is the same as that of the OPEN "O" command, namely:

```
OPEN "I",n,"filename"
```

Once the file is open, we use a variation of the INPUT statement to read the data back into variables so that the program can use it. For example:

```
INPUT#1,A,X$,F4,Q1,B$
```

This procedure works fine so long as we have only stored numbers in the file, but when strings have been stored there are some additional problems that must be watched for. For example, a comma is recognised as a separator between items of data so a string containing embedded commas will be read back as two parts.

```
A$="HI, THERE"
PRINT#1,A$
INPUT#1,X$,Y$
PRINT X$,Y$
```

will result in the following output:

```
HI THERE
```

Other problems with strings include cases where the string contains any leading or trailing spaces, single quotes or double quotes and also cases where two or more strings are printed to the file by the one PRINT statement and get read back as only one string. Thus when using files to store strings, care must be taken with the contents and with the input and output of them to the file. One solution to this is to always print strings on separate lines in the file by using separate PRINT statements for each string, and then get them back from the file with the LINE INPUT# statement. This statement works the same as an INPUT# statement, except that it only works for string variables, and only for one at a time. However, when inputting using LINE INPUT#, only a carriage return (or ASCII character 13) is taken as a data separator and all other characters are treated as part of the string.

As with file output, once all input from the file is completed the file should be closed with a CLOSE statement so that the directory and end of file marks are correctly maintained.

Using these statements, we can now write a program that will read back the list of names saved in the previous example and print them for us.

```
10 OPEN "I",1,"NAMES/DAT:0"
20 LINE INPUT#1,A$
30 PRINT A$
40 GOTO 20
50 END
```

The obvious problem with this program is that it will keep reading past the last name in the file and will get an error. This doesn't matter with this simple program, but would be bad in the middle of a larger one, so we need some method of finding when we reach the end of the file. For the examples given a simple change to the original program by changing the order of two lines, and it

insertion of extra lines in the second program will make this possible as follows:

```

10 OPEN "O",1,"NAMES/DAT:0"
20 PRINT"TYPE THE NEXT NAME OR END TO FINISH"
30 INPUT NA$
-->40 PRINT#1,NA$
-->50 IF NA$="END" THEN GOTO 70
60 GOTO 20
70 CLOSE 1
80 END

10 OPEN "I",1,"NAMES/DAT:0"
20 LINE INPUT#1,A$
-->25 IF A$="END" THEN GOTO 45
30 PRINT A$
40 GOTO 20
-->45 CLOSE 1
50 END

```

This simple solution will not always work however, as it may not be possible to pick a value to mark the end, which will never occur in the normal data and thus cause a premature end to the input. BASIC includes a command that allows for this, namely the EOF() function. This function is equal to 0 for a file opened in "I" mode if the last item of the data has not yet been input, and becomes -1 when the end of file is reached. Thus our input program could be modified as follows to work with the unmodified output program:

```

10 OPEN "I",1,"NAMES/DAT:0"
20 LINE INPUT#1,A$
30 PRINT A$
-->35 IF EOF(1)=-1 THEN GOTO 45
40 GOTO 20
-->45 CLOSE 1
50 END

```

The number placed in the () for the EOF() function is the buffer number of the file we wish to test.

Using OPEN "O" and OPEN "I" we can create new files and read old ones, but how do we add to an existing one? One way is to open the existing file as "I" in one buffer, and open a new file as "O" in another buffer, and then read all existing data in and copy it to the new file after which the "I" file is closed and any new data is also written to the "O" file.

```

10 OPEN "I",1,"OLDFILE/DAT"
20 OPEN "O",2,"NEWFILE/DAT"
30 LINE INPUT#1,A$
40 PRINT#2,A$
50 IF EOF(1)=0 THEN GOTO 30
60 CLOSE 1
70 PRINT"TYPE NEW DATA OR END"
80 LINE INPUT A$
90 IF A$="END" THEN GOTO 120
100 PRINT#2,A$
110 GOTO 70
120 CLOSE
130 END

```

However, BASIC provides a much easier way to extend existing files. This is done using the OPEN "E" command, which is similar to the "O" command, but adds data to the end of an existing file without deleting any data already stored there. If the file opened in the mode "E" does not exist BASIC will generate an error. (NOTE: not all versions of BASIC support this mode.)

One additional point that must be watched for when using disk files, is that there are lots of new errors that can result when using them, which can cause your program to crash. For example, forgetting to close the drive door or leaving a write protect tab on a disk can stop your program and may result in the loss of data that has taken some time to enter. It is therefore a good idea to include an ON ERROR GOTO trap in your program that looks for disk related errors and gives a warning message and a second chance to the user.

In the next article on this topic I will go into details on how to use random access files for data storage. If anyone wishes more information on the items covered in this article, or has any questions on BASIC that they would like answered or explained, please do not hesitate to write to me care of the Editor.

=====

LS-DOS Corner

by Gary K. BRYCE

The following article takes a slightly different "tack" to those I have previously presented, in that I am using a BASIC program to demonstrate the use of two SVC's (QGTMOD and QGTDCB) rather than an assembly language program.

Unlike the ROM based BASIC interpreters supplied with the Models 1 and III, Microsoft BASIC for the Model 4 does not allow ready access to the printed line count (or other print parameters). BASIC for the Model 4 relies on the user having filtered the printer with the FORMS filter prior to loading. As FORMS/FLT is relocatable you cannot simply PEEK a specific location (as can be done with the printer DCB on the Model 1 or III) to gain access to variables controlled by the FORMS filter.

The program presented below allows you to access any of the variables controlled by FORMS/FLT to test or modify if desired and but can only be used with LS-DOS 6.3. I have taken the opportunity to demonstrate the use of the SVC access from BASIC introduced with the 6.3 release. I will present a version for use with earlier versions of TRSDOS/LS-DOS 6.x next month.

As it stands the program is designed to become part of any of your existing BASIC programs where you want to be able to access or change any of the FORMS parameters. Before going into a detailed description of the program operation I will detail the two SVC's used.

QGTDCB -> SVC-82

This SVC is used to determine the location of the Device Control Block (DCB) for the specified device.

Registers Used: AF, HL, DE.

DE => Two character device name (1st char in B, 2nd char in D). If DE=0, then a pointer to the first unused DCB will be returned.
HL <= Address of the DCB.
Z <= Set if the DCB exists.

QGTMOD -> SVC-83

This SVC locates the entry address of a memory resident module if that module uses the correct header protocol.

Registers Used: AF, HL, DE.

DE => Pointer to the name of the module terminated with any character within the range of X'00 to X'1F.
HL <= Returned entry address of the module.
DE <= Pointer to the address of the first byte past the module name storage within the module header.
Z <= Set if the module was found in memory.
Now on to a description of the program itself on a line by line basis.

10 - Define variables beginning with J as INTEGERS. That is to say, values in the range of -32768 to 32767.
20 - Dimensions a one dimensional array J(x) with a total of 6 elements. This array is used by the new BASIC interface to SVC's, where each element is used to pass information that may be needed by the SVC by using the previously illegal USR11(VARPTR(ARRAY(0))) statement.

J(0) - SVC number on entry, Z flag on exit.
J(1) - Value passed to/from HL register.
J(2) - Value passed to/from DE register.
J(3) - Value passed to/from BC register.
J(4) - Value passed to/from IX register.
J(5) - Value passed to/from IY register.

30 - Defines a function FNINTG%, which is used to convert a SINGLE PRECISION variable to an INTEGER. This function is taken straight from Lewis Rosenfelder's excellent book "BASIC FASTER AND BETTER".

40 - Defines a function FNVECT%, which stores a vector in a variable specified when the function is used. It will convert two bytes at a given address +1 to an integer. More details can be found in the description for line 100 and 130.

50 - Stores the SVC number for the QGTDCB SVC in the first element of the J(x) array used in the SVC interface.

- 60 - Places the device name PR into the third element of the array to be passed into the DE register.

```

i.e.          HEX
ASC("P")      =  X'50
ASC("R")*256   =  X'5200
              -----
              X'5250
              -----

```

So that the D reg will contain X'52 (R) and the E reg will contain X'50 (P) on execution of the USR11.

- 70 - Is the USR11 call to the new SVC interface so that the @GTDVCB SVC is executed to find the address of the Device Control Block for the PR device.
- 80 - The address of the DCB is returned in the HL register and as such is accessed from the second element of the array J(1).
- 90 - The first byte of a DCB contains the TYPE field of the DCB. A value of X'06 means that the printer is not linked to any other device (i.e., not filtered), while a value of X'47 means that it is linked. More details are available in the Technical Reference manual or in the Programmers Guide to LDOS/TRSDOS 6.
- 100 - The FNVECT% function is used to perform a 2 byte PEEK operation on the VECTOR field of the DCB (DCB+1 and DCB+2) to find the address of the printer driver.
- 110 - FORMS/FLT has an inbuilt name of \$FF within the header of the module. The FORMS\$ string is used to point to the module name for use in the @GTMOD SVC.
- 120 - Stores the SVC number for the @GTMOD SVC in the first element of the J(x) array used in the SVC interface.
- 130 - Uses the VARPTR and FNVECT% functions to create a pointer to the contents of the FORMS\$ string to be passed to the SVC in the third element of the J array and so into the DE reg as required by @GTDVCB.
- 140 - Executes the @GTDVCB SVC to find the start address of the FORMS/FLT module in memory (if present).
- 150 - Tests J(0) for the Z flag. FLAG will be 64 if the Z flag was set (\$FF was found).
- 160 - The start address (in HL) of the \$FF module is passed to the MODADD variable.
- 170 - The program is terminated if the \$FF module was not found or if the address of the printer driver and the beginning of the \$FF module do not match. (This may occur even if the *PR is filtered with FORMS/FLT and subsequently filtered or linked to another device).
- 180 - Call to a subroutine which stores the current settings of the FORMS/FLT into the FVAL array.

The main body of your program would be placed after line 180. Of course, you can change the program as presented to reduce the number of lines by using multiple statements per line as well as including any additional dimensioning or declarations that your program requires.

Lines 20000 to 20030 can be called at any time from within your program so that the current parameters can be read. You could also access these same locations to change any of the current settings using the statement :-

```
POKE MODADD+DISP,VALUE
```

Where DISP is a value of 12 to 21 and VALUE is 0 to 255. DISP is the displacement from the start of the FORMS/FLT module to where the FORMS parameters are stored. The parameters of accessed by MODADD+DISP are as follows :-

| DISP | DISP |
|-------------------------|----------------------------|
| 12 - Page size in lines | 13 - Line Count |
| 14 - Lines/page | 15 - Chars printed on line |
| 16 - Translate from | 17 - Translate to |
| 18 - Indent | 19 - ADDLF/TAB expand |
| 20 - Max chars/line | 21 - Left margin |

As mentioned previously, there is one floppy drive available for FINANCIAL and Honorary members to purchase. A form is provided below, which you may enter your bid. All bids will be placed in a hat (or other suitable receptacle), and the lucky one will be drawn out. The momentous happening will eventuate at Sefton on 10th December. So if you are interested this is positively your last big chance, (we think)!

SYDTRUG

Floppy Drive Bid

Name_____

Address_____

Phone_____

COST \$100+freight

```

10 DEFINT J
20 DIM J(5)
30 DEF FNINTG%(A!)=-(A!>32767)*(A!-65536))-((A!<32768)*A!
40 DEF FNVECT%(A!)=FNINTG%(PEEK(A!+1)+PEEK(A!+2)*256)
50 J(0)=82
60 J(2)=ASC("P")+ASC("R")*256)
70 X=USR11(VARPTR(J(0)))
80 DCBADD=J(1)
90 TYPE=PEEK(DCBADD)
100 VECTOR=FNVECT%(DCBADD)
110 FORMS$="$FF"+CHR$(3)
120 J(0)=83
130 J(2)=FNVECT%(VARPTR(FORMS$))
140 X=USR11(VARPTR(J(0)))
150 FLAG=(J(0) AND 64)
160 MODADD=J(1)
170 IF FLAG<>64 OR TYPE=6 OR VECTOR<>MODADD THEN PRINT
    *PR has not been filtered with FORMS/FLT:END
180 GOSUB 20000

```

**** Main program goes here ****

```

19999 END
20000 FOR DISP=12 TO 21
20010 FVAL(DISP-12)=PEEK(MODADD+DISP)
20020 NEXT DISP
20030 RETURN

```

Help Wanted Department

Here we print requests by individual members for assistance.

If you have a problem and need help, let us know and we will publish your request here. Some reader may know just what you need. However, if you do receive help, please let us know so that we can publish the answers too.

This month David SUTTON is in need of a copy of the public domain communications program KERMIT which will run on his System 80. any one can help, he can be contacted on (02) 771-5482.

Worth Repeating

Nothing succeeds like reputation.

-- John Huston

What we really need is more young people who will carry to their job the same enthusiasm for getting ahead that they display in traffic.

-- "Changing Time"

An authority on ageing says that men are smartest at 50 - which precisely when there is nobody around the house who will listen to them.

-- Bill Vaughn

Every baseball team could use a man who plays every position superbly, never strikes out and never makes an error - but there's a way to make him lay down his hot dog and come out of the grandstand.

-- "The Gour"

When a resolute young fellow steps up to that great bully, the world and takes him boldly by the beard, he is often surprised to find that the beard comes off in his hand, that it was only tied on to scare away timid adventurers.

-- Oliver Wendell Holmes

SYDNEY TRS-80 USERS' GROUP

Floppy Drive Bid

To the Sydtrug Committee,

I wish to bid for the Floppy Drive which has become surplus to the group's requirements.

The group believes it to be in working order

BUT it is being sold AS-IS. Cost \$100+freight.

What's This BOOLEAN Business? Part 3

by Max MORRIS

[Reprinted from "Adelaide Micro User News", G.P.O. Box 214, ADELAIDE S.A. August 1987]

This month we will look at some programming uses of logical AND and the advantages over other alternatives. Some years ago "80 Microcomputing" published a series of "one liners". they were games which could be written in one line of BASIC. That is, the whole game was less than 255 characters long. My favorite was called "SNAKEAROUND" by Bill Peters. I have printed it here in thirteen lines for ease of reading, but it can be written entirely in one line.

It sets pixels randomly about the screen and the player uses the four arrow keys to move or "snake" between them. The game ends if the snake hits a pixel. It achieves all this and keeps score by clever use of BOOLEAN algebra.

```
10 N=128:F=16:X=64:Y=24:CLS
20 FOR T=1 TO N:N=SET(RND(127),RND(47))
30 A=(PEEK(14400)AND 120)/4
40 F=-(A=0)*F+A
50 D=F AND 6
60 C=(F AND 24)/4
70 X=X+C*(C>0)
80 Y=Y+D+3*(D>0)
90 X=X+N*(X=N)-(X<0)
100 Y=Y+48*(Y>47)-(Y<0)
110 IF POINT(X,Y) CLS:PRINT "SCORE: " P:END ELSE SET(X,Y)
130 P=P+1
140 NEXT T
150 REM SNAKEAROUND BY BILL PETERS FROM "80 MICRO"
```

Some of our members are so intelligent that they can read a line of BASIC and know what it does, but remember, I am the one who had trouble with the program PRINT "FRED" in Pascal. This means I have to work through lines with a pencil and paper to see how they work.

Let's start at line 30. The program has to decide which arrow key is being pressed and then move the snake accordingly. When an arrow is pressed a value is stored in memory at location 14400. The arrow keys are up, down, left and right and the values stored are 8, 16, 62, and 64 respectively (but in binary form). PEEK(14400) will give one of these values. In ordinary programming we could PEEK(14400) and then send the program to one of four subroutines to move the snake, return, check the keyboard etc. Very simple to write, but not very economical. Line 30 is more elegant.

The number 120 in binary is 0111 1000 so that $120 = 64+32+16+8$. If we and any of the values given by pressing an arrow key with 120, the value is returned unchanged. Any other values are masked out. Another advantage is that it can respond if two keys are pressed simultaneously so that the snake can move diagonally.

Let us assume that no key is pressed and work out the values in each line. Line 30 $A=0$ (no key pressed). Line 40 tests if $A=0$, as it does the value -1 for TRUE is placed in the brackets so we can evaluate $F=-(-1)*16+0$ so $F=16$. Line 50 $D=16 \text{ AND } 6=0$. Line 60 $C=(16 \text{ AND } 24)/4$ therefore $C=16/4=4$. Line 70 $X=64+4*3*(-1)=65$. $(C>0)$ is TRUE so is replaced with -1 in the brackets. Line 80 $Y=24+0+3*(0)=24$. $(D>0)$ is FALSE so 0 is returned in the brackets. Lines 90 and 100 prevent the snake moving off the screen by testing if $X>128$ and $Y>48$. Line 90 $X=65+128*((0)-(0))=65$ (both tests false). Line 100 $Y=24+48*((0)-(0))=24$ (both tests false). Line 110 tests if the snake has hit a SET pixel (POINT (X,Y)). If it hasn't its position is SET and the loop continues. If no key is pressed yet, X enters the loop with a value of 65, which calculation will show

becomes 66 in line 70 but Y remains at 24 so the snake just moves across the screen to the right until X becomes 128 and then one logical test in 90 becomes TRUE. Minus one replaces a zero and $X=0$ and the snake reappears at the left side of the screen.

It would take far too long to calculate each line again for each arrow key when PEEK(14400) gives 8, 16, 32 or 64 but I suggest you do it yourself for at least one value, perhaps 8 or 16 (up or do arrows) and see how the program works.

Also check the values in line 100 when the snake reaches the top or bottom of the screen when $Y=0$ or 48. You don't have to draw the bit patterns in lines 50 and 60 Just try out the values in BASIC as you did last month. $16 \text{ AND } 6 = 0$ and $16 \text{ AND } 24 = 16$. This program achieves its purpose by using BOOLEAN algebra in a very economical way. There are a number of clever techniques worth learning, particularly if you like writing games. Once you get it running, keep the kids away or they won't do their homework unless you have a machine which has the arrow keys all on one side of the keyboard not spaced for two-hand use like the Model 1.

Signed Numbers

by Bill ALLEN

[Reprinted from "BITS & BYTES", newsletter of the TRS-80 SYSTEM 80 Computer Group, 41 Montclair Street, ASPLEY QLD., August 1988]

Sydtrug Editor's Note: This short item was a sort of postscript to article on BOOLEAN algebra which will explain the seeming unconnectedness of the opening sentence.]

Some readers may have long since been wondering why, early in this article, the computer started off returning a value of -1 for true for the truth testing ($A<15$), while the individual bit testing returns a value of 1. In BIT arithmetic, there are ONLY TWO values -- 1 or 0 (i.e., there is no provision for signed numbers at this level). The reason for the minus is provided by the fact that the first process deals only with INTEGER values. If you have been studying your binary, you would know that two bytes (16 bits) can represent up to 65535 (1111 1111 1111 1111 in binary) or FFFF in Hexadecimal. However, the term INTEGER has an additional meaning in this case. It not only means that the number is rounded off to a whole number (no fractions or decimal places), but is also a SIGNED number.

By SIGNED number it is meant that there is an INDICATOR BIT which tells the computer whether the number is positive or negative (+ or -). The bit which is used for this sign is the Most Significant Bit (the one farthest left -- Bit 7 -- in the Most Significant Byte). So, in INTEGER functions, we have only fifteen bits available to represent the value in Binary -- hence the range of values -32768 to +32767. If the sign bit is 1, then the number is a negative value and if the sign bit is 0, the number is shown as positive. As the TRUE at bit level returns a 1, the sign bit will be a 1 and therefore the TRUE number naturally returned from a truth test will be negative.

In practice, on the screen there is a space instead of a + sign when a positive value is printed, because we always assume a value is positive unless we are told otherwise (-), so there's normally no real need to print the + sign in a result. Some computers do the conversion of the TRUE (-1) to a positive (1), but it doesn't really matter either way, as long as the programmer is aware what it will be. He/she will adjust the program arithmetic to suit when using truth testing functions, remembering the mathematical fact that $-1 * -5 = +5$ (or minus value times minus value equals positive value)!!