

NEW
8096 and MMF9000
PETS IN EDUCATION

MAY 1981

COMMODORE CLUB NEWS



Volume 3 Issue 4

NEW HARDWARE

VIC, 8 inch disk drives,
colour PET, 8096 and MMF9000

D.M.S.

Review of Compsoft's
Data Management System

DISK DRIVES

Start of a special beginner's course

BASIC 'vs' MACHINE CODE

Same program in two
different languages

EDUCATION

*SPECIAL PULL-OUT
SECTION*

Contents

Editorial — A new editor, and the future expansion of Commodore Club News	2
The PET Show — It's coming soon.	2
Approved Products — A new man in charge.	4
New Products — 8096, 8061, 8062, colour 8032, VIC, MMF9000.	4
Petpack & General Disk Library — New Releases, plus a review of Cosmiads and Arcadia One.	6
Life — Corrections to manual, plus a review.	8
Marcopoulos Shell footnote — The saga continues.	8
Business User's Column — Barry Miles on Security.	10
Review Section — Luxor Copyholder, various add-ons, plus a review of Compssoft's Data Management System	13
Disk Use for Beginners — A beginners guide to using disk drives. First in a series.	16
Machine Code Arithmetic on the PET — A guide.	18
BASIC 'vs' MACHINE CODE — The same program written in both, and comparing the two execution times.	19
EDUCATIONAL SUPPLEMENT	
Editorial — What we're doing in Education.	1
NCC — A report on microcomputers in schools and colleges.	1
Chelsea College — A project's description.	4
Taylor Wilson on MUPET.	4
Public Domain Software — The philosophy behind Commodore's public domain software releases.	5
Trevor Lusty , on the setting up of a school's computer club.	5
Educational Workshops — A list of the current educational workshops.	7

Next Month

The Next Newsletter.

The main feature will of course be the PET Show. We'll be having a special pull-out feature on it, telling you everything you need to know to get the most out of it. And, of course, the coupon for free entry to the Show for all newsletter subscribers.

There will be an extensive article about the PET and the IEEE, written by David Muir of Napier College, something of an expert on the subject.

We'll be continuing our series on the use of the disk drive for beginners, and for the more advanced user an article on file access in machine code.

Two articles for relative newcomers to BASIC, on string equations and the use of arrays in BASIC.

As well as this there'll be a number of contributions from the educational field that we couldn't fit in this time around, plus the usual miscellany of facts and figures on the PET.

Old tricks for new Pets...

COMMAND-O is a FOUR KILOBYTE Rom for the 4000/8000 Basic 4 Pets with all the "Toolkit" commands RENUMBER (improved), AUTO, DUMP, DELETE, FIND (improved), HELP, TRACE (improved & includes STEP), and OFF - plus PRINT USING - plus four extra disk commands INITIALIZE, MERGE, EXECUTE, and SEND - plus extra editing commands SCROLL, MOVE, OUT, BEEP, and KILL - plus SET user-definable soft key, 190 characters - plus program scroll up and down - plus 8032 control characters on key. Ask for Model CO-80N for the 8032 or CO-40N for the 4016/4032. £50.00 plus Vat

New tricks for old Pets...

DISK-O-PRO is a FOUR KILOBYTE Rom that upgrades 2000/3000 Pets, but lets you keep all your old software - including Toolkit. As well as REPEAT KEYS and PRINT USING, you get all the Basic 4 disk commands CONCAT, DOPEN, DCLOSE, RECORD, HEADER, COLLECT, BACKUP, COPY, APPEND, DSAVE, DLOAD, CATALOG, RENAME, SCRATCH and DIRECTORY - plus extra disk commands INITIALIZE, MERGE, EXECUTE and SEND - plus extra editing commands SCROLL, MOVE, OUT, BEEP and KILL - plus SET user definable soft-key, 80 characters - plus program scroll-up and scroll-down. We recommend the 4040 disk or upgraded 3040 for full benefit of disk commands. Ask for Model DOP-16N for new Pets 2001-3032, and 2001-8 with retrofit Roms & TK160P Toolkit. £50.00 plus Vat, other models available.

PRONTO-PET hard/soft reset switch for the 3000/4000 Pets. We don't think you'll "crash" your Pet using our software, but if you do the Pronto-Pet will get you out! Also clears the Pet for the next job, without that nasty off/on power surge. £9.99 + Vat

and no tricks missed!

KRAM Keyed Random Access Method. Kid your Pet it's an IBM! VSAM disk handling for 3032/4032/8032 Pets with 3040/4040/8050 disks means you retrieve your data FAST, by NAME - no tracks, sectors or blocks to worry about. Over 2,500 users worldwide have joined the "Klub"! Now you can too, at the 1981 price, £75.00 plus Vat.

SPACEMAKER All our Rom products are compatible with each other, but should you want, say, Wordpro with Kram, or Disk-o-pro with Visicalc, then SPACEMAKER will allow both Roms to address one Rom socket, with just the flip of a switch, for £22.50 plus Vat.

We are sole UK distributors for all these fine products. If your CBN dealer is out of stock, they are available by mail from us, by cheque/Access/Barclaycard (UK post paid) or send for details.

Calco Software

Lakeside House Kingstons Hill Surrey KT27QT Tel 01-548-7258

Editorial

Introduction

Since you received your last newsletter, David Middleton, the then editor, has moved on to Commodore's Technical Support department. So, allow me to introduce myself as your new editor, and at the same time offer my thanks to Dave Middleton for all the help he's given me. My name is Pete Gerrard — you may have noticed my name attached to articles in previous newsletters.

First of all, let me apologise for the time you've all had to wait for this newsletter — the transfer of various personnel has meant inevitable delays in reaching publication dates. However, Commodore Club News will be appearing with increasing regularity from now on, and in the not too distant future we will be accelerating to become a monthly publication. Without any increase in subscription and advertising charges, you'll be pleased to note! As well as becoming a monthly publication, we will be extending the size and content of the Commodore Club News in future issues. In addition we will be incorporating a number of articles that would previously have appeared in "Microcomputers in Schools and Colleges", to give the broadest possible coverage of the PET and its many uses. It is hoped to include a special pull-out with every edition of the newsletter. Future ones planned include a guide to the PET Show in June, and an article about the exhaustive testing a piece of Business Software goes through before being released as part of the Commodore Software Library. Having said that, I'd like to say a few things about content.

Content

There has been a debate in previous newsletters about the content of the magazine. Is Commodore forgetting the beginner? Does Commodore think that every 80 column SuperPET user is purely a businessman and nothing else? Are pages and pages of machine code (In whatever form of output) desirable? Obviously everybody is going to have



Editor Peter Gerrard checks the clock — opening time is pending.

their own opinions of what should go in a newsletter, and ideally we should aim to be somewhere in-between all those opinions. I would appreciate any comments you may have on the presentation of this, and any future, newsletters. I'll try and strike a happy medium (never did like laughing witches).

On a personal note, I do think more attention ought to be paid to the beginner, and I hope that this is reflected in this issue. Machine code addicts will not be forgotten, so don't despair. There appears to be a big demand for articles on the use of disk drives. Again, we've started a beginner's guide in this issue, and this will grow into a very useful series of articles. Finally, being the COMMODORE Club News, I think you ought to know a little bit more about Commodore and the people in it. Consequently, we'll be telling you more about the people behind Commodore as the months go by.

I hope you enjoy the magazine, and at the same time find it useful and informative. That's what we're here to be. If you disagree on anything, dislike anything, have any grumbles (or even praise!), contact me, the editor, at the address at the end of this introduction. Also, if you've written an article and would like it considered for publication, again you get in touch with me. The article can be submitted in any shape or form, as long as it's decipherable. Volume 3 issue 5 is on the way!

Address to send articles and comments to:—

The Editor
Commodore Business Machines
818 Leigh Road
Trading Estate
Slough
Berks.

To renew your subscriptions, contact the same address, but mark your envelope 'CCN Subscriptions'.

The PET Show

Many of you will have attended the PET Show at the Café Royal last year, where the best in software and hardware for the PET was on show. Not only Commodore, but many of our dealers and Approved Products people were there as well. It was an ideal opportunity to see just how much support is given to the PET — anybody who saw the massive array of talent available can't fail to have been impressed.

In fact, I only heard one complaint about the PET Show — it wasn't big enough. Well, this year your prayers are answered, because on the 18th June at the West Centre Hotel in SW6, the second International PET Show will get under way.

With 14,000 square feet at our

disposal, over 100 exhibitors from all over the world will be demonstrating applications from industrial to commercial, medical to educational, and basically covering everything possible on a PET. Whatever your profession is, by going to the PET Show you'll be able to see how you can do it more cheaply and efficiently. After all, over 100,000 European PET users can't be wrong.

Commodore's Software Department will be demonstrating all their Business Software; many of the Approved Products range will be there; a special programmers corner; specific user-interest seminars; educational workshops; and much, much more. The next issue of the newsletter will have a special pull-out feature on the Show, containing much more detailed information about the Show itself, and telling you how to get there, where to stay, where to eat and drink, and the precise dates and times of opening.

And, of course, subscribers to the newsletter get in for free!! Next issue will have a special cut out form enabling you to see the Show for nothing. At the Show, as well as seeing the massive array of software and hardware, you'll get the opportunity to meet many other User Club members in informal surroundings. As Editor of the newsletter I'll be there as well, so if you want to come along and meet me feel free to say hello, and we can have a chat about whatever. If there are any questions I can't answer, I'm sure I can point you in the right direction! Hopefully this will be a chance to get to know both sides of the fence, as it were, better, and any comments you might like to make on the newsletter will be given a proper airing. As I say, as well as myself there'll be plenty of other User Club members present, so who knows what might happen?! See you there.

Approved Products

As PET sales continue to grow (over 1900 units shipped in March) so the need for a wide range of software and hardware add-ons grows with it. To help supply this need was the original idea behind the Approved Products range, and already a wide variety of products are available. In the last Approved Products catalogue there were twenty pages describing items available. But, as the market expands

so must everything associated with it, and this is why there has been a growing emphasis on the Approved Products range over the last few months — you may have seen the article by Andrew Goltz in Volume 3 Issue 2 of the newsletter about how a product goes about getting approved; a testing process. Obviously it is to everybody's benefit that any prospective product goes through an intensive grilling before being able to bear the stamp of "Officially Approved by Commodore". If it gets that far, it must be good.

Clive Booth

To continue this emphasis, and to enhance and improve the existing range of Approved Products, Commodore have recently appointed Clive Booth as 'the man in charge'. One of Clive's many roles is overseer of current and possible future additions to the available range, and so anyone considering "Commodore Approval" should contact him at the usual address. It is his declared intention to promote Approved Products as much as possible, and to set the ball rolling the first catalogue from Clive will be appearing shortly. There will also be a very large number of Approved Products suppliers exhibiting at the forthcoming

John Collins, bodily going where no man has gone before.

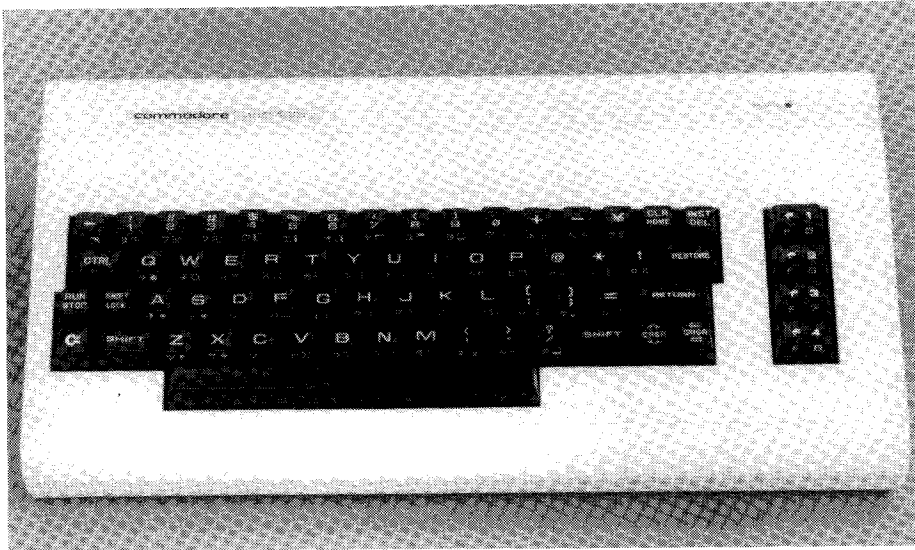
PET Show in mid June, and these will be given the maximum amount of available publicity. Next month's issue of the newsletter will contain a lot more details about exhibitors. Of course, newsletter subscribers will receive a copy of the new catalogue as soon as it is available. Format and layout of this new catalogue will be very much up-market from previous publications, and this, along with the PET Show, should ensure an even greater market penetration for an already successful area.

One field in particular that Clive is keen to explore is the 'linking' between programs, particularly in the Business Systems area. An excellent example of this is the programm DMS from Compsoft Ltd., which provides linking facilities between itself and Wordpro/Wordcraft, and Visicalc as well. There is a review of this package elsewhere in this newsletter. We would hope to increase the number of products available that show these features, as in this way we can help to ensure an 'across the board' approach to Approved Products.

So, if you have a product that has so far not come to our attention, and which you think is worthy of greater publicity than it is so far getting, contact Clive Booth. You'll be taking the first step on a mutually profitable path.



New Products



Before saying anything else it must be mentioned that none of the hardware items I'll be talking about are expected to appear before the end of the year, with the one exception of the VIC.

VIC Delivery

The latest news we have on VIC delivery is that they'll be launched via the dealer network at the end of May/early June. So, by the time of the PET Show in mid June you'll be able to place orders on the stands with the dealers. All of the dealers who'll be selling VICs are getting extensive training on the machine, so that we'll be able to continue with the excellent support the dealer network already gives. In the next issue we'll be able to give more details.

As a subscript to the VIC story, did you know that the VIC is being marketed in Germany as the VC? And the reason? Well, when it was launched in Germany named as the VIC, it caused a bit of a stir. Apparently, in Germany, VIC is the word used to describe what two people get up to when there's a power failure.

Two other products that have been mentioned in the past were the 8096, the 8061 and 8062 eight inch disk drives. The two disk drives have very little news on them, other than being shown at the recent Hanover Fair. However, I think we're a long way from seeing any in the UK for a while.

On the 8096, the 96K 80 column PET, there is rather more news on it's configuration, but again it will be some time before they're available for release. So don't pester anybody! We'll let you know. Meanwhile, to be going on with, here are some specs. on the 8096. (Who'll be the first magazine to think of a new name for it, rather like the 8032 soon became the SuperPET? MegaPET perhaps?).

CBM 8096 (64K add on RAM)

The 64K add-in memory adds 64K-bytes of RAM to the CBM 8032 providing a total of 96K-bytes of RAM. The add-in memory is mapped into main memory in 32K-byte blocks. Only one 32K-byte add-in block can reside in main memory at one time.

Control of the add-in memory is through a register on the add-in board that provides selection of two 16K-byte blocks, write protection, and enabling of the add-in memory. Additional bits in the register allow the programmer to PEEK at I/O and screen memory.

The add-in memory was designed to be used by programmers that are familiar with the internal software operations of the CBM 8032.

The add-in memory board is mounted on the CBM 8032 console by four mounting brackets, and is connected to the computer by two power cables and one logic cable.

This board will enable PET to be used as a fully "Soft" machine. Basic Languages or operating system changes will thus cause no problems as they can be boot loaded from disk. The 16K paging system is simple to use, thus making it easy to convert existing software for larger memory capacity.

It is essentially an 80-column Super PET with an Add-In board fitting inside the case and giving it 64K bytes of memory. There are three interconnecting ribbon cables with headers, positioned as follows:—

1. 40 pin dip connector between main logic board (UB14) and Add-In board (CJ3).
2. 6 pin power/control cable between main logic board (J10) and Add-In board (J2).
3. 6 pin power/control cable between main logic board (J11) and Add-In board (J2).

Description:—

i) The Add-In board provides 64K bytes of RAM in 16K byte blocks.

ii) It is controlled by the contents of a control register (write only) located as \$FFF0 (65520) which cannot be read directly. However, a copy is kept somewhere at address space \$0000-\$7FFF (0-32767). The program to change the contents of this register should also be located in the lower 32K of the original machine.

iii) The bit structure of the control register is as follows:—

Bit 7 = Enable the Add-In expansion.

Bit 6 = Allow I/O PEEK through at \$E800 (59392).

Bit 5 = Allow screen PEEK through at \$8000-\$9000 (32768-36846).

Bit 4 = Board select (always set=0).

Bit 3 = Select one of the two 48-64K blocks.

Bit 2 = Select one of the two 32-48K blocks.

Bit 1 = Write protect the 48-64K blocks.

Bit 0 = Write protect the 32-48K blocks.

iv) Memory map structure for CBM 8032 with Add-In is given on the next page.

v) If Add-In control register bit 0 = 1, writing into blocks 0 and 1 is inhibited by hardware. If Add-In control register bit 1 = 1 then writing blocks 2 and 3 will be inhibited by hardware.

vi) At any time two blocks of 16K bytes can be enabled and mapped into 32-38K or 48-64K byte address space. With AICRB7 = 1 and AICRB4 = 0 (enables Add-In and selects board).

AICRB3	AICRB2	Blocks Enabled
0	0	0 and 2
0	1	1 and 2
1	0	0 and 3
1	1	1 and 3

vii) On power up Add-In board RAM is not selected. The memory map of CBM is unchanged. RAM is located in the lower 32K bytes of address space and RAM, I/O and screen and screen RAM in the upper 32K. Programs written to function on CBM will run without modification, the only exception being that the program must not write into location \$FFF0 (65520). This location is reserved for the write only control register whose bits select the function of Add-In memory board.

I/O Handling:

I/O devices in the CBM are located as follows:

6520's at \$E810 (59408) and \$E820 (59424)

6522 at \$E840 (59465)

CRT controller at \$E880 (59520)

Screen memory at \$8000 (32768)

These devices can be accessed in two ways:—

i) Switch all Add-In memory out and restore normal memory map.

ii) Use I/O PEEK through feature of bit 6. This is necessary when a RAM loaded application program accesses I/O. Screen memory follows the same recommendations.

PEEK through feature:—

If Add-In control register bit 5 = 1, accesses to locations \$8000-\$9000 (32768-36846) will not be taken from Add-In memory blocks 0 or 1 but from normal screen memory. Similarly if Add-In control register bit 6 = 1, accesses to locations \$E800-\$F000 (59392-61440) will not be taken from Add-In memory (blocks 2 or 3) but from normal I/O locations on the board.

Interrupt processing:—

When the contents of the control register are being changed, it is necessary to execute SEI instruction

to avoid the possibility of being interrupted. It must be remembered that the interrupt vector in the CBM point to routines in ROM which are not accessible when Add-In memory is selected. Therefore the memory manager (6502) must set vectors to a point at a routine that will switch modes (in/out Add-In memory), call interrupt service and then restore the Add-In memory board.

8032 Colour PET

A prototype of a colour 8032 was also on display at Hanover, but it is still very much a prototype. Basically it was done to prove that it could be done, and market tests are still going on to see if there is sufficient demand for the machine. Looking further ahead, there are plans afoot to produce a Commodore hand-held computer by the spring of 1982, but as yet there are no details available on it.

MMF9000

Finally, on to the MMF9000. The what?! MMF stands for micro-mainframe, and the following comes from an American press-release.

Mr Irving Gould, chairman of the board of Commodore International Limited, on April 2nd said that Commodore has just announced a "new generation computer called the Commodore micro-mainframe at the Hanover Fair in Hanover, Germany."

According to Mr Gould, "this brand new computer combines the power of mainframe computing with the low-cost of microcomputers." Mr Gould went on to note that "the first of the three principle features of the Commodore micro-mainframe is its ability to accept any one of five mainframe languages, including Fortran, APL, Basic, Pascal or Assembler, while Cobol will shortly be available. In addition, the Commodore micro-mainframe has full screen editing capabilities. The second principle feature is its mainframe communications capabilities, including high speed RS232 interface and terminal emulation ability. Finally the Commodore micro-mainframe has built into the system a 96K RAM virtual memory system and 132K memory including 36K ROM."

In addition the Commodore micro-mainframe incorporate both 6502 and

6809 microprocessors and an intelligent operating system which gives full Commodore software and peripheral equipment compatibility.

Mr Gould went on to say that "the exact price of the Commodore micro-mainframe will be announced at the National Computer Conference in Chicago beginning on May 4th (ed. note — you'll be reading this after; we go to press before!), but that it will be priced 50% to 75% below the lowest-priced competing mainframe with comparable results.

Deliveries of the Commodore micro-mainframe in the UK are expected at the end of 1981.

Breakthrough

Mr Gould further noted that "this breakthrough in value and technology is the result of close collaboration between Commodore computer systems division and one of Canada's foremost technical universities, the University of Waterloo, located in Waterloo, Ontario, which has gained an international reputation in the field of computer studies. The usual high level of documentation associated with Waterloo software will be available."

Professor Wesley Graham of the University of Waterloo commented that "the Commodore micro-mainframe offers the user a comfortable entry into large scale computing. One of the main features of the micro-mainframe is that the high level languages specifically developed and run on a stand alone micro-mainframe, and then later transferred to a traditional mainframe, should a user's future growth require it. This puts Commodore firmly into the mainstream of both the mini and mainframe computer industry with a very significant price advantage for the end user."

Mr Gould concluded by noting that "Commodore systems division thus continues its policy of close cooperation with distinguished universities and with independent systems houses such as BMB Compuscience of Milton, Ontario, a significant contributor to this project."

That's the end of the press release, so this is me again. Once again I must stress that none of the above (apart from the VIC) are anywhere near delivery. We just like to keep you informed of what's going on in the world of Commodore.

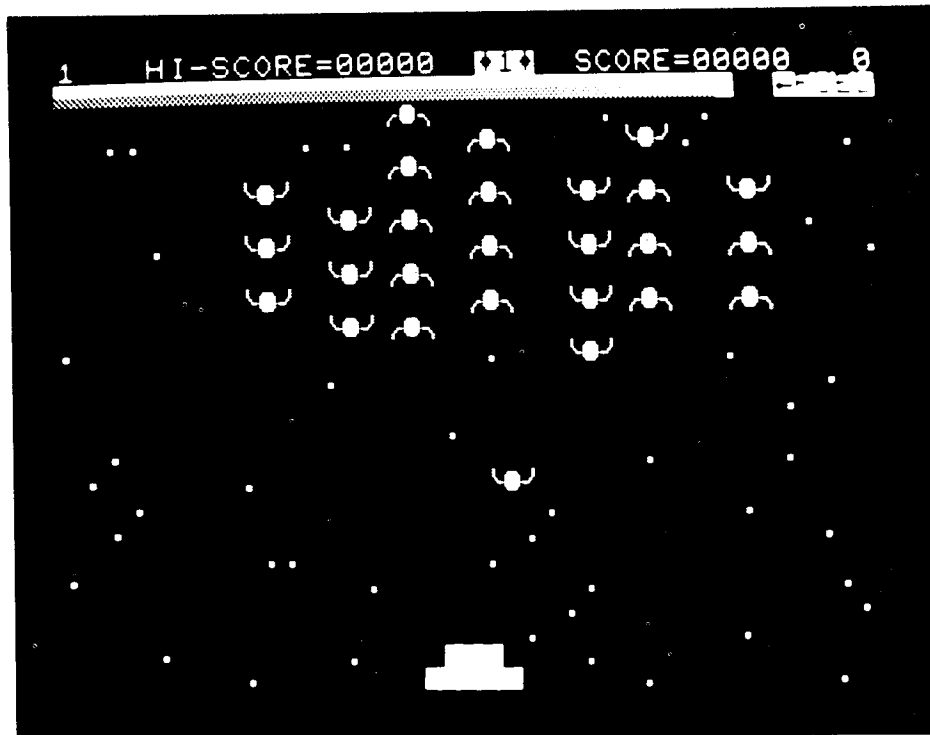
Petpack and General Disk Library

Three new programs have been released this time around. These are:—

MP106 — Cosmiads — £10.

An innovation in arcade game writing. This is the tenth game in the very popular arcade series, and although I'm continually being proven wrong on saying this (and quite happily so!) it is the best so far.

The game is played throughout against a constantly changing background of stars, and as in most of the 'space wars' type of arcade game you are controlling a spaceship which can move along the bottom of the screen, and which can launch lasers at the aliens poised above. When you load and run the program you will initially be given a screen of instructions, which basically tell you which keys to press to move and fire. 'S' starts the game, and at any time whilst the instructions are being displayed you can change the level of the game. There are an incredible ten levels of play for the game, ranging from 0 (the lowest) to 9 (the highest, most difficult and for masochists only!). It is pre-set at level 5, which, as you might surmise, is in the middle of the range of difficulty. When you're satisfied with the level of difficulty, press 'S' and you're into the game itself. Now is the time, if you've got sound, to turn it on.

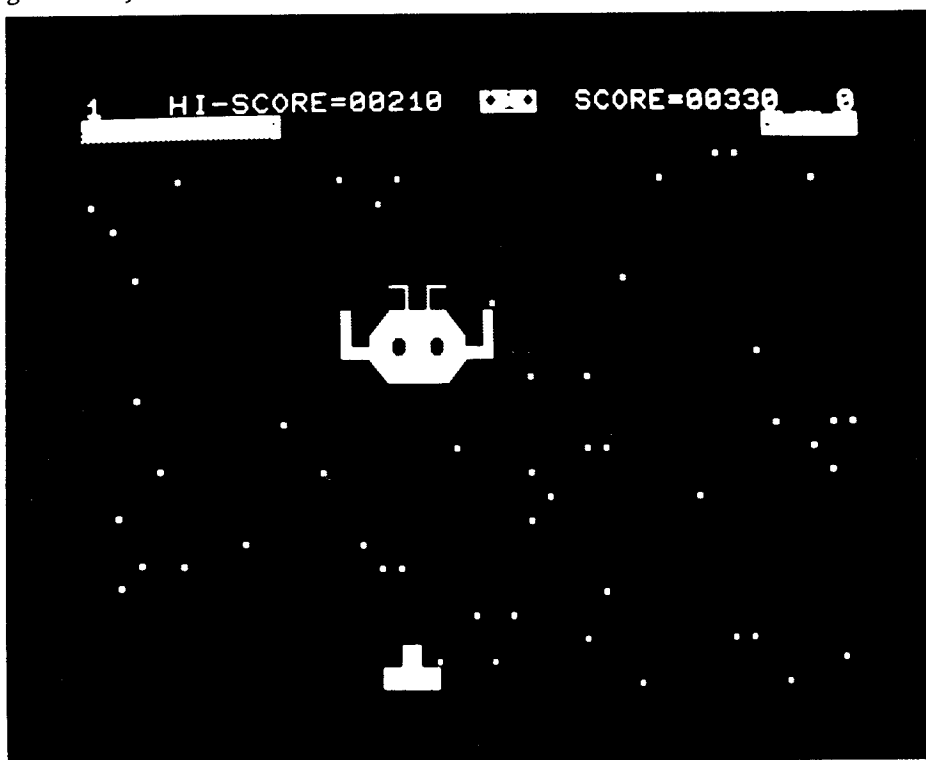


The screen will change to show, at the top, a fuel gauge filling up, and a screen filling up with stars, which constantly flicker throughout the rest of the game. If at any time during the game you run out of fuel, as this is constantly being depleted during the course of play, you are destroyed and the game is over and, as at the end of every game, you're returned to the instruction page. You can, of course, quickly get back to the start of another game by pressing 'S'. At the

top of the screen you're shown how many lives you've got left, what the highest score is, what screen level you're on (more of that in a minute) what your current score is, and what level of difficulty you're on.

Screen level is yet another example of how excellent this game is. For each level of difficulty you select (0 to 9), there are nine further sub-levels of difficulty, which are the screen levels mentioned earlier. So in other words you have ninety different levels of play!

Onto the game itself. Your mission is to destroy the swarms of invaders which appear on the screen above you. These beings sway back and forth across the screen, but don't fire at you. You might think, therefore, that since you can fire at them, and they're not firing at you, it might be easy to destroy them and win the game. However, at frequent intervals one of them will emerge from the swarm and come dive-bombing down the screen at you, firing as he does so. They are also extremely intelligent, so staying in a corner to avoid them is not a good idea as they'll simply crash into you and you lose one of your three lives. Incidentally, you get a bonus life by scoring 1,500 points, and a further bonus at every further 15,00 points you manage to get. Points are gained for shooting the aliens, as you might imagine.



Still, if you manage to avoid the Kamikaze aliens, don't run out of fuel, don't get hit by their fire and manage to kill them all off, you are faced with the prospect of a Zylon. This is a very large alien being, which speeds across the screen from left to right and back again, getting lower as he does so. To kill this off you have to shoot it in the mouth, which is continually opening and closing. If you

do you get a bonus score, but as he dies he shoots at you, so watch out! If you don't get him you get no bonus, but don't lose a life.

Once you've been through this, the smaller cosmiads re-appear again, and you once more go through the process of trying to kill them all off. This time though the action is more furious, with two of them coming down at once, and firing more bullets!

And so the game progresses, getting faster and faster as it goes on, with more and more aliens zooming down the screen at you, until it reaches almost impossible level. Quite simply this game is superb — a game amongst games. It's by the same author as our current fastest selling arcade game Cosmic Jailbreak, with the promise of more to come. We'll keep you informed!

Arcadia One

January saw the rather low-key launch of what is possibly the best example of machine code programming in the games world, namely Arcadia One, GD1820, retail price £25 (plus VAT). For this we owe our thanks to Dave Pocock and Paul Higginbottom of Commodore, who between them did the extra programming required for this product. I say extra, because all of the games were already written — although they did have to be altered — but the major work came when they had to be put together in a package that was as easy to use as possible, and was as versatile as possible. The consequence of this is a product that works on Basic 2.0 or Basic 4.0 (although not the 80 column — yet!) and works with 2040, 3040 or 4040 disk drives. To start playing any game, you have to press just one key. And the whole product is presented in a very attractive and well documented package.

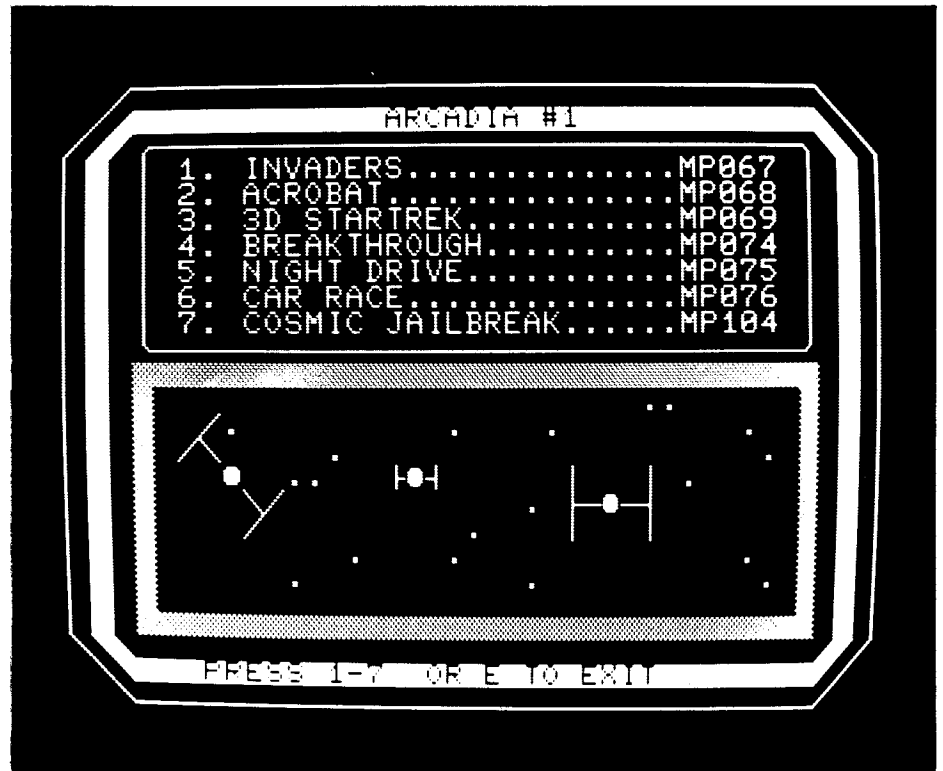
To be more specific, Arcadia One is a collection of the first six arcade games in the cassette library, namely Invaders, Acrobat, 3D Startrek, Breakthrough, Night Drive and Car Race. When you get the package, you put the disk in the right hand drive (drive 0), and type LOAD"*",8, followed by pressing the RETURN key. This will initialise the disk and load the first program from it, which is the menu for the entire set of games. The name of each game is displayed, preceded by a number from 1 to 6. To play the required game you simply press the relevant numeric key and the game is loaded and is ready to go. After each game, pressing the run/stop key takes you back to the menu, ready for any one of the other games to be played. It is also one key stroke to exit from the program. So all told it is a very versatile, simple to use product.

One word of warning — this is a

'protected' disk, so under no circumstances should you attempt to either make a copy of it or write anything to it. If you find this disturbing, don't worry. Our new software production department now replaces any faulty diskettes on a day to day basis, so, if ever anything should go wrong, you need never be without your favourite games for long! Also, you must not put a write protect tab on the disk. Why not? Buy it and see!

The request for highest scores on the arcade games has not produced the greatest of responses. Is this because you're so busy playing games that you don't have time to sit down and write a letter?! Anyway, here is an internal list of highest scores (some programs get really exhaustive testing!):-

Invaders	—	35,000
Acrobat	—	12,200
Breakthrough		1,340
Night Drive	—	1,420
Car Race	—	9,560
Crazy Balloon	—	8,540
Cosmic Jailbreak	—	80,900



But don't let these put you off (maybe they don't!). Write in anyway, and perhaps we'll draw up a Commodore 'vs' the rest of the world team — first prize, 20 pence for a game of Invaders!

MP107 — Basic Aid 4 — £10.

This is a Basic 4.0 40 column version of the popular Basic Aid. It will add a large number of extra commands to PET's Basic, including things like renumber, merge, trace, find, etc. The program can be relocated anywhere in your PET's memory, so that it can be used on any Basic 4.0 40 column PET. Also included is a Basic 4.0 version of DOS Support, which gives one key versions of the most common disk commands.

MP108 — Extramon 4 — £10.

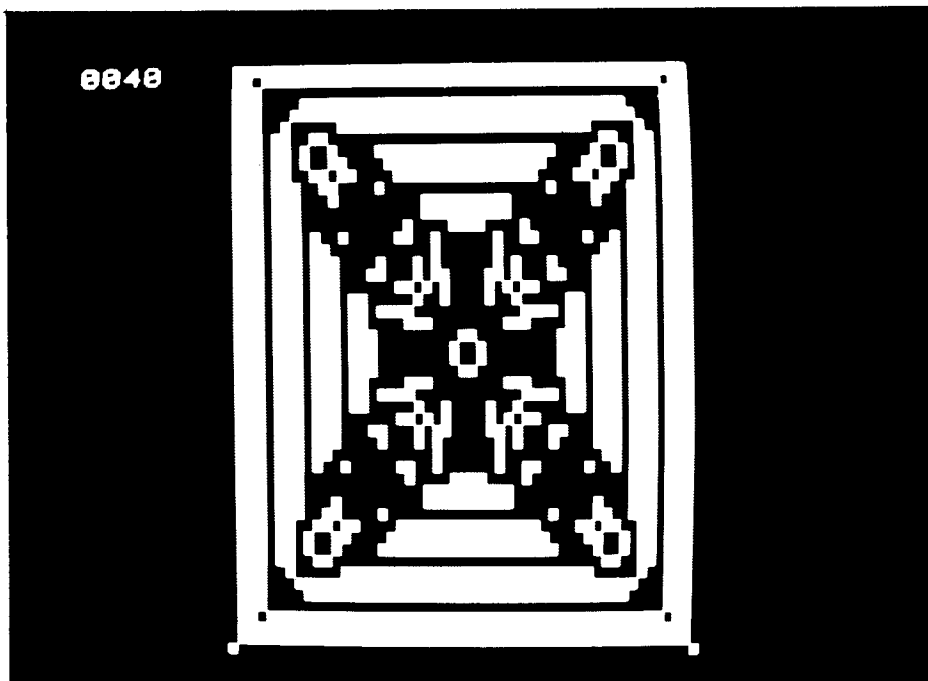
This is again a Basic 4.0 version of an existing program, in this case the original Extramon. The program adds a number of extra commands to the resident monitor.

Life — MP103

If you've already bought this package, there were one or two things happening on the screen that were not mentioned in the accompanying manual. I'd like to explain what those things were.

First of all, the generation counter in the top left hand corner of the screen. By pressing the 'X' key, this can be toggled off, i.e. if it's off and you press the 'X' key it will come on, and vice versa. Of course, it still continues counting when it's not displayed on the screen. Pressing the 'Z' key will set the counter to zero again. The other addition is the use of a single step generation display. In the manual it states that by pressing the 'H' key, the cells will stop generating. This is certainly true, but one thing that's not mentioned is that by repeatedly pressing the 'H' key, you can jump ahead just one generation at a time, and pressing 'G' will get you back to continual generation.

To those of you unfamiliar with the program, it is a PET version of Conway's famous game of Life. There have been versions available before, but this one goes far beyond anything



that's been on the market before. The first main thing you notice is that instead of restricting itself to the 40x25 size of the screen, it gives you a 120x120 display size, and the actual PET screen forms just a window of that display, as you scan up or down, left or right. You can use quarter square graphics displays instead of the more conventional 'blob' as one of the cells. Many more

features are present — the counter and single step as mentioned earlier, designing and storing for later recall your own 'standard' patterns, the ability to call up a number of pre-set, well known patterns, and so on.

It works on any PET with more than 16K RAM, and as an added bonus will even work on the 80 column PET. And all for seven pounds as well!

Marcopoulos-Shell Footnote

Nick Marcopoulos (CPUCN, Vol 3.1, 24) gives a version of the Shell sort tailored for the PET. On the 2001 series PET subscripts starting from 0 can only range up to 255 before wrapping round to 0 again. Are Marcopoulos's magic numbers (59,17,5,1) the best when N must be under 256? A few months of exploration show that (45,13,4,1) will sort random data on average about two percent faster. The magic numbers need not all be prime, but they definitely should be co-prime in pairs. The test programme was much as Marcopoulos's but with:—

```
3 W=RND(-SQR(TI)/PI)
```

```
6 I=K+M+K2:Z=T+TT+K1:INPUT"ITEMS";N=N-1;DIM X(N),A$(N),K9(3)
```

```
7 K9(0)=45:K9(1)=13:K9(2)=4:K9(3)=1
```

```
.....
```

```
50 K1=3+(N>9)+(N>34)+(N>106):IF N=0 THEN RETURN
```

```
53 FOR K2=K1 TO 3:K=K9(K2):FORM=0 TOK-1:FOR I=M+K TO N STEP K:
```

```
W=X(I)
```

```
56 FOR J=I-K TO M STEP -K:IFA$(X(J))>A$(W) THEN X(J+K)+X(J):NEXT
```

```
58 X(J+K)=W:NEXT I:NEXT K:RETURN
```

Here is a skeleton table of average sorting times in jiffies for different N:

N	T
10	39.2
25	148.1
50	387.8
75	656.7
100	964.7
150	1611.7
200	2315.2
256	3150.4

These average times are based on 200-500 sorts at each value of N. Some puzzles remain: for example (41,13,4,1) was the second best set of magic numbers and (43,13,4,1) inexplicably turned out worse than both its neighbours.

The optimum set of magic numbers in the best worst-case sense would presumably have more and smaller magic numbers: best worst-case is not the same criterion as fastest average on random data. Equally, if the data will always be nearly in sort fewer and larger magic numbers will do better.

All this effort for a mere two percent? Yes, but this is the difference between professional and amateur. The professional, when he has got his factor-twenty improvement cannot afford to spend his expensive time on further tiny improvements. The amateur's time is not costed, so he can afford the luxury of gratifying his curiosity. This amateur would like to know if the best magic numbers for (say) N=1000 are (X,45,13,4,1) or a completely different set: can anyone with a more modern PET tell him?

This time we will turn our attention to the question of security. This is a very important matter, and was examined to some extent last time, insofar as the question of internal control procedures, for security of data against error and/or fraud were concerned. This is far from being the whole story, however.

We must remember the extreme mobility of disks, and the consequent danger of the loss of confidential information by their removal, also there is the question of accidental or malicious damage to our business as a result of the concentration of so much that is important to us in so small a space.

There is a possibility of our not appreciating the revolutionary nature of the computer: like all revolutionary phenomena, it can be a force for evil as well as for good. It is true that the ability to store over 170,000 characters of information on a single disk, (or three times that amount in the case of the 8050 disk unit), is extremely advantageous, but it is equally true that the same information can easily be accidentally destroyed, or stolen, since the disks are so very portable, and in certain regards vulnerable.

So we must seriously address the question of disk security, program security and data security, seeking to eliminate the problem before it occurs: a form of preventative maintenance!

Diskette Security

First: How vulnerable are the media with which we must work? This is extraordinarily difficult to decide. Rumours as to the suitability of the various makes of disks are rife. I have heard of the entire stock of a certain manufacturer's disks being withdrawn from a supplier's hands, and replaced free of charge, with no explanation. No-one seems to know the truth of the matter. The only warning I have seen from a software manufacturer is to the effect that the life of the disk is 40 hours in contact with the read/write head of the unit. This does not tell us a great deal, since we cannot expect to keep a log



of the times when contact is occurring. We merely have a guide to the effect that random access file usage will cause the data disks to wear out very much faster than will using the same media to store programs, which are loaded into the machine once, whereupon the disk is put back into its cover and stored away.

Does the fact that disks are now available with reinforced hubs and in thicker packets imply that the others are too vulnerable, or merely more vulnerable? Who knows?

In any case, we can lay down some basic rules for disk-handling which, if obeyed, will reduce any problems to manageable proportions.

Firstly, make sure that everyone who handles disks realises that the parts where the oxide coating is exposed should never be touched, since the layer of grease which covers our skin will reduce the reading head of our disk unit to an illiterate hulk!

It is perhaps unfortunate that the name "Floppy Disk" has fallen into common usage, since it encourages people to "see how far they will bend". This is a very good way of making sure that the surface is distorted, and illegible. It might be thought that "only a small amount of

data will be lost", but unfortunately the disk unit relies totally on the directory track to obtain access to the data on the disk, and if *that* track is corrupted you have effectively lost all your data. It is true that Harry Broomhall, one of the most expert PET users in the country has developed a "Lazarus" program to raise disks from the dead, but the charges for the service are far from trivial.

Dangers

What other dangers lurk for the unwary? Well firstly, the fact that we are dealing with *magnetic* media should not be forgotten. This means that any magnet, whether permanent or electro, placed close to the disk, is likely to destroy its contents. Thus we should keep any device powered by an electric motor away from our disks, and this includes the motor part of a vacuum cleaner. Furthermore, we should avoid like the plague those rather smart metal stands which are designed to hold copy conveniently for typing by means of a magnet.

The attractions (pardon the pun) of magnetic clips for papers to be attached to filing cabinets temporarily

Educational Supplement

Introduction

As you can see we have a special section in this newsletter, covering the educational area. This is to replace and enhance Microcomputers in Schools and Colleges, and will become a bi-monthly feature. Obviously we will have educational items in other newsletters, but this special section will be totally devoted to the PET in Education. This way you'll be getting a regular flow of information on PET-related activities in schools and colleges, and at the same time an opportunity to see what is going on in the rest of the PET world. Similarly, you'll be getting the chance to show other PET users what's happening in education. We'll be covering future conferences, public domain software, educational workshops, and anything and everything to do with PETS and education.

If there's anything you'd like to see in this section, can you contact the editor of the magazine. Articles can be in any shape or form, typed, handwritten or drawn in blood — as long as I can read it!

Special Offer

The growth of the educational market is shown by the number of PETs that have been sold in this area — approximately 10,000 in the last

three years. To give extra help to this market Commodore appointed Nick Green to be in sole charge of the PET and Education. With Nick's support there have been an increasing number of seminars, offers, etc. to aid the large number of schools and colleges using our equipment. Two of the latest offers will be the public domain issue of Comal and the free issue of a disc containing Pascal, Assembler and Lisp for either Basic 2 or Basic 4 for any educational establishment purchasing a 4032 or a 4040 from the start of April to the end of June. For the purposes of this offer an educational establishment is defined to be one of the following: a state controlled primary or secondary school, a private preparatory or secondary school, a college of further or higher education, a polytechnic or a university. The offer is limited to one "language pack" disk per establishment irrespective of the number of units purchased. This is purely a disk based system, which is why we're including educational establishments who purchase a 4040 disk drive. This software, worth £281.75, will significantly increase the value of any system using it, giving as it does a multi-language capability on the PET. Elsewhere in this educational supplement you can read an article describing A-level teaching with Pascal, and what a useful aid this is. We're offering not only Pascal, but Assembler and Lisp

as well, giving you the possibility of straight away using your PET in any one of four different computer languages. Please discuss any details of this offer with your local dealer.

Educational Workshops

Published for the first time in this newsletter is a list of educational software workshops — around 140 in all. What IS an educational workshop? Basically, it is any educational establishment that uses PETs, and is willing to act as a base for testing, receiving and distributing educational software, and in particular the public domain software that Commodore have been releasing, and will continue to release in the future, with your support, of course. With Commodore's continuing commitment to education, these workshops provide a valuable means of support to existing and future PET-in-Education Users. If there's anyone we've missed out, can you write to Nick Green at the usual Commodore address in Slough, and we'll get your names in print as soon as possible. There are a number of reasons for printing these names and addresses, but perhaps the major one is the growing number of educational PET users who need help and advice in starting up. We hope that the workshops will be able to assist them in this, and help keep the PET the UK's leader in education.

The NCC, Micros and Training

Graham Beech, The National Computing Centre

The Author

Graham Beech is NCC's CBT specialist — he was previously at Wolverhampton Polytechnic, where some readers may remember him as Director of the Central Program Exchange. Although he is no stranger to computers in education, he has found quite a few differences in the industrial and commercial world; some are shown in Table 1 overleaf.

The NCC, Micros and Training

The National Computing Centre, (NCC) is the focus for many computer activities in the UK — including microcomputer applications and computer training. A new development for NCC is computer based training (CBT) — a natural fusion of the microcomputer and training.

Most readers are aware of — maybe even users of — computer assisted learning (CAL) packages. These are either obtained from a centralised source (e.g. the Schools Council Scheme) as a package or they are produced locally in an ad hoc fashion, often deficient in documentation and even in reliability! Neither of these approaches is prevalent in training. A few packages are available (e.g. for touch typing as in Adaptive Keyboard Trainer, £15.00, MP095,

and for fault finding) but the majority of CAL in training is produced to professional standards by teams of people, including educational designers and programming experts. So far, they have used mainframe systems such as PLATO or, more commonly, IBM's Interactive Instructional System (IIS). The importance of these systems is that they provide a pre-structured framework into which the user enters his training material. They provide support for such important aspects as:

- Text and graphics
- Answer matching
- Routing within and between lessons
- Question construction (e.g. multiple choice)
- Allocation of resources (e.g. videotapes)
- Student records.

So, there is more to training than meet the eye — hence the term “computer based training” to distinguish it from straightforward CAL.

The reason for the popularity of these powerful systems is that they dramatically reduce the time spent on CAL development. Any of you who have worked with CAL will know that it takes about 200 hours of design, coding and debugging to produce just one hour of instruction if

you use BASIC. The point is that BASIC is not a suitable language for CAL, unless your time is not chargeable! Using IIS, PLATO and other systems, the development ratio can be reduced to 50:1, or even better. Since this is about the same ratio as for videotape or tape-slide production, CBT is now a serious contender as a presentation medium. What makes it so attractive are three very important points:

Attractive Features

- 1) Uniquely, it requires the trainer to **input** the training content: this forces the training course to be examined in detail.
- 2) Again, uniquely, the trainee is forced to **interact** with the delivery medium.
- 3) Once produced, CBT can be delivered at the time and place required, so in the long term, it is **cheaper** than competitive media.

Furthermore, it is “inflation proof” — fewer human trainers are needed, and modifications are made easily and inexpensively.

These features characterise CBT. At the NCC, we are trying to provide these same advantages on microcomputers — thus opening up CBT to a

much greater audience. Our first achievement has been to produce a “frame-generator” package, which interacts with the trainer **via** a series of prompts. Learning frames are linked together to reflect, essentially, a branched learning package. Each lesson, in the current version, consists of up to 80 frames, and up to 10 lessons can be stored on one 5¼" floppy diskette. The user is not aware that he is using a large BASIC program!

The development work has been done on a Tandy microcomputer, but versions are being produced for the PET and other micros. We believe that frame generators — and other easy authoring aids — could give a big boost to CBT and to CAL.

Apart from software development, NCC is also offering:

- **Workshops on CBT:** for trainers and teachers who need a good awareness of the technique.
- **Consultancy:** NCC will visit individual organisations and advise on specific requirements, quite impartially.

For further details, or just to put your point of view, write to:

Graham Beech
National Computing Centre Limited
Oxford Road
Manchester M1 7ED.
Tel: 061-228 6333 Extn. 224

	Education	Training
Style	Mainly CAL — mathematical simulations are very common.	CML is more important — much training is decentralised.
Applications	Spreading through the curriculum — French grammar to Computer Science.	Mainly used in procedure training — e.g. form filling.
Costs	Sensitive — so micros are essential.	Up to £20/hr including development costs is tolerated so long as . . .
Effectiveness	Not a great issue — students can not leave before the end of term!	. . . the course is more effective, preferably shorter than before.
Equipment	In schools — mainly micros, sometimes a general purpose mainframe.	Usually customer's own mainframe — some bureau services — a few micros.
Skills	Most teachers learn and use BASIC, PILOT, etc.	Few trainers wish to learn how to program, so structured packages are used.

Table 1: Some differences between Education and Training

A-Level Teaching with Pascal

It will not have escaped anyone's attention that a great deal of fuss has been made recently about Pascal. Its supporters claim it as the language of

the future, while its detractors call it a false idol, and point out the many advantages of BASIC with machine code subroutines as a standard pro-

gramming language for microcomputers. Whatever the rights and wrongs of this argument, some A-level examining boards have

recently been expressing dissatisfaction with the way that candidate's knowledge is usually restricted to BASIC, and Pascal has been mentioned at examiners meetings with teachers. With the release by Commodore of TCL Pascal (a superset of UCSD Pascal) A-level teachers using PETs now have the chance to find out about Pascal for themselves.

TCL Pascal is a disk-based system, with the following features:

1. An editor for Pascal source-programs. This gives the full PET screen edit, and a number of extra features, such as upper/lower case commands, automatic line numbering, find, change and delete commands, and decimal/hexadecimal conversion.
2. A resident compiler, which occupies most of RAM, but leaves about 5000 bytes free for short source programs. Using this, short routines can be compiled into a standard code called p-code without any use of the disks. Working like this, a few special features of Pascal are not available. Nevertheless, it is a very useful mode to use when teaching the language.
3. A disk compiler, allowing long Pascal programs to be compiled into p-code. All features of Pascal are supported, together with a number of special PET features such as hexadecimal constants, PEEK, POKE, GET and PLOT facilities, random number generator and clock. Disk data files can be used in this mode, and it is also possible to chain different Pascal source-code files from the disks.
4. A p-code interpreter, allowing compiled p-code object files to be run. This runs very much faster than a BASIC interpreter.

Obviously, not all these are resident in PET RAM at the same time. TCL Pascal in fact consists of a short core machine code program which is always resident at the top of RAM, together with five sequential files, containing machine code which can be loaded into RAM and run as required.

So what has Pascal got to offer? The main point is that many A-level students are likely to go on in the world of mainframes, and it is therefore undesirable that their knowledge of programming should be entirely restricted to BASIC. The

question as to whether or not Pascal is in any sense a "better" language than BASIC is therefore irrelevant — the important thing is that it is different. The following features are probably particularly important.

Important Features

1. The philosophy of Pascal is very different from BASIC, as it is descended from ALGOL rather than from FORTRAN, and is therefore part of a different tradition.
2. Like most languages in use in large installations, Pascal is a language which has to be compiled before it is executed. This allows students to be introduced to a number of important ideas, such as the difference between compile-time and run-time errors. As TCL Pascal uses a single-pass compiler, error trapping is not very efficient, and tracking down an error from a large number of different error messages pulled off a file on a floppy disk can be very challenging.
3. Pascal is a language designed around the concepts of "structured programming". Roughly speaking, this means that programs are built up out of self-contained blocks, each with just one entrance and one exit. Variables in these blocks are independent although they can be easily passed, and as up to six character variable-names can be used, their names can be self explanatory. This is essential in a commercial environment, where most programming is done as team projects, and where it may be necessary for one programmer to take over another's work at very short notice.

Something approaching structured programming is of course possible in BASIC, by avoiding GOTOs and making full use of sub-routines. Most people however would agree that there isn't much point to going very far with this, and that too rigid an insistence on structured programming in BASIC is unnecessarily restrictive.

Structured Programming

Pascal on the other hand is a language in which structured programming is completely natural.

a. A Pascal program is built up out of procedures with independent variables, which can be called by name. Using the chaining facility of the disk mode of TCL Pascal, it is possible to combine procedures written by different people into a single program. This would make team projects at A-level a very real possibility.

b. There are a number of different and powerful ways of looping in Pascal, being FOR...DO, WHILE...DO, and REPEAT UNTIL.

c. The standard conditional in Pascal is an IF...THEN...ELSE construction.

d. Because of b and c above, it is completely unnecessary to use GOTO in Pascal. (GOTO is possible and not hard, but it is made deliberately awkward and ugly).

4. Pascal is a language designed to force clear thought. With BASIC it is possible to start a program from scratch on the keyboard, and fiddle about with it until it works, which is a very good thing for a student just beginning programming. In Pascal such an approach is not possible, which is a very good thing for a student starting A-level. Two reasons for this are the compiled and structured features already mentioned. Another is the necessity to declare all variables at the start of a procedure, and to be very careful about their types — integer, floating point, boolean, etc. Additional variable types can also be defined by the user.
5. Pascal supports some very powerful COBOL-type data structures such as arrays, sets and records. These are potentially very useful in A-level projects.
6. Pascal is a recursive language, meaning that a procedure can call itself. (This is possible in a stack-based BASIC such as the one used on PET, but not to a very useful extent). Recursion allows very elegant solutions to programming problems such as the eight queens or the Tower of

Hanoi, and is used quite a lot in advanced mathematical programming. It is also on most A-level syllabuses

Conclusion

Pascal is an excellent language for use in A-level computer science courses. With BASIC and TCL Pascal for

high-level, and the Commodore Assembler to illustrate low-level programming, the PET rivals any other machine for A-level teaching. The TCL version of Pascal shows all the important features of the language plus some extras, is a well-designed system, and is very reasonably priced.

Michael Bolley

(Michael Bolley teaches mathematics and computing science at Modern Tutorial College. He has himself recently implemented a new language for PET by writing a machine-code interpreter for an extended version of common PILOT).

Chelsea College Projects

Further to some recent publicity to the contrary, we have pleasure in announcing that a range of quality software in the field of CAL (Computer Assisted Learning) is now available for the PET. The materials have been developed by the Schools Council 'Computers in the Curriculum' Project based at Chelsea College and in their current format, are PET adapted versions of the Project's first phase of the work. Primarily written for use on mainframe-teletype systems, these programs present output in the form of alphanumeric text and graphs. New editions of the science materials which will take advantage of the graphic capability of the PET are now being developed with the support of the new DES grant to the Project. These new versions will be available at the end of next year. Most software is available in both disk and cassette format (OLD and NEW ROM) but is obtainable by a somewhat unusual method.

Unlike virtually every other item of

micro-software, the Chelsea materials are only released on purchase of the teachers' and students supporting notes. These are published by the Schools Council and distributed by Edward Arnold (Publishers) Ltd. Disks and cassettes are obtainable from Chelsea College on receipt of the completed form distributed with these packs. Documentation is also provided for all programs.

The subject areas covered are Biology, Chemistry, Physics, Geography and Economics together with the multi-disciplinary Home Heating. The latter requires random access to a large amount of data in a number of files and so can only be used with disk systems. Each other pack covers about seven topics, details of which can be obtained from Chelsea College or Edward Arnold.

The pack price includes the cost of software regardless of whether the cassette or disk version is required. The prices are listed below:

Biology	£20.50
Chemistry	£16.60
Physics	£18.40
Geography	£20.80
Economics	£22.00
Home Heating	£ 9.20

The materials are generally aimed at pre-CSE/O-level through to advanced A-level pupils with a larger topic spread in some subjects than others.

Materials are also available for the PET from the Chelsea Science Simulation Project which covers topics aimed at sixth form and first year undergraduates. These programs are currently published as separate units by Edward Arnold. Software is available by the same means as the Computers in the Curriculum Project. Pack prices are between £6 and £9 for each topic.

Addresses:

Chelsea College
Computers in the Curriculum
Project
Friese Greene House
Chelsea Manor Street
London SW3 6LX

Edward Arnold (Publishers) Ltd
Woodlands Park Avenue
Woodlands Park
Maidenhead
Berks.

Taylor Wilson Multi-PET System

The Taylor-Wilson Multi PET System generated a tremendous amount of interest at the Which Computer show at the National Exhibition Centre in November. Prospective educational users were quick to appreciate its potential to provide up to eight PET users with access to a shared disk and printer thus enabling them to provide the maximum number of keyboards with full processing facilities for much less than the cost of providing disks and printers dedicated to each PET.

Among the many existing educational users of the Multi-PET system are a number of commerce departments in colleges. The Multi-PET system is used with up to eight of the new 80 column, 12" screen PETs sharing a disk and daisy wheel printer to provide multi-station word processing facilities for a fraction of the cost of dedicated word processing systems. WORDCRAFT is now available with utilities built in to support up to eight stations with both local and shared printers with one or more disk drives.

Important features of the Multi-PET system are:

Support any combination of PETs with BASIC 2.0 or BASIC

4.0 ROMs.

Any combination of peripherals in local or shared mode.

British made — supplied with two year guarantee.

Has been successfully used with PETs up to 100 metres from disk.

Supplied complete with utilities and tutorial for ease of use.

Choice of manual or program control, complete security of files being transferred.

Additional stations may be added to existing installations.

Enquiries to Peter New at Taylor-Wilson Systems (tel: 056 45 6192) or any PET dealer.

CBM brings Software into the Public Domain

Commodore Business Machines' investment in the education market started with a series of regionally organised free one-day seminars on the use of microcomputers in schools, specifically aimed at teachers who were interested in implementing microtechnology in the classroom. Due to the very successful response to the first seminar, CBM published and distributed to a large number of educational establishments their "Microcomputers in Schools and Colleges" newsletter. Commodore have now further extended their educational programme by releasing a series of free educational software programs for use by teachers in schools and colleges.

The software comprises CAL and simulation programs aimed at age ranges, 3 years to 1st year University standard and the subject matter ranges from arithmetic problems to political science.

Known as Public Domain software in the United States, the idea of distributing free software was originally started by that well-known PET figure, Jim Butterfield when he wrote the 6502 programs for the KIM I Microprocessor. The KIM I subsequently became one of the most popular hobby microcomputers on the US market, and when the original 8K PET was introduced, Jim Butterfield continued to distribute his partly finished programs free of charge by now mainly to teachers and hobby groups. The interest shown in the software escalated and CBM Canada took up the distribution, having collected programs from North America and Canada.

CBM UK hopes that teachers will take full advantage of this freely available Public Domain software — especially those teaches learning or participating in workshop groups. These workshops are an ever-growing number of self-help groups which have spontaneously been set up around the country to familiarise or train teachers and schools in the use of classroom microtechnology. In some cases, software development groups have also been formed.

Need for Feedback

CBM hope that with the release of this software there will be feedback and contributions of programs to add to the public domain series, and in particular contributions from teachers in non-numerical subjects using their PET in the classroom. It is hoped that the programs in this first release will come back extended, modified and improved.

Nick Green, who runs CBM's educational programme said: "We are the only microcomputer company in the UK who offer such support to the educational establishment. We cannot ignore the fact that there are currently 10,000 PETs in use in education and the more feedback and information we have on the use of PETs in classrooms, the more we can offer by way of practical help to both teachers and students by extending our educational programme." Nick added "There was a clear call from our second London Conference for more curriculum oriented software. We urge subject matter experts to produce topics that will guide teacher-programmers. We will gladly assist and encourage any proposals in this area."

The first person to take advantage of the release of this software was Mr Trevor Lusty from Telford School, Telford Newtown, who recently spoke at a "Microcomputers in Education" seminar organised by Commodore in Birmingham about his own classroom experience with the PET. Mr Lusty has organised the first PET workshop in his area and plans to use the software for familiarising himself and other teachers with new techniques.

Any teachers and workshop group organisers who wish to obtain any releases in the Workshop series should contact Mr Nick Green at the Leigh Road address.

Trevor Lusty writes:

In 1977 it would not have been unkind to have described Shropshire as a computing desert. To be fair there were a few cases of enlightenment, with about six schools doing some form of computing, mainly through the good offices of the local technical colleges or the facilities of

friendly universities.

Each year, around the beginning of July, most of Shropshire's 53 secondary schools close down for the day and their teachers meet on what is affectionately known as CSE day. Here subject teachers discuss the latest examination papers, elect new panel representatives and chat about teaching in general. It was at the Computer Studies meeting where the usual fifteen or so were moaning about the lack of facilities, non co-operation from colleagues and other educational matters; when one member said, "Let's form an Association". The usual doubts were expressed but in the main it was felt that it was a positive step. The first task of the Association would be to prepare a report on computing in the County and make recommendations to our administrators stating what we (the teachers) felt the County should be doing to encourage the development of computer education. The chap with the bright idea found himself the Chairman, someone was bullied into being the secretary and a working party was formed to prepare the report. The Shropshire Association for Computing in Education (S.A.C.E.) was formed.

The first meeting was held a month or so later, and was very well attended. There was a lot of enthusiasm, much skill and a great deal of interest in all aspects of computer education within the County, a resource which could be used to good effect. The proposed report was completed and forwarded to all schools and the administrators at the Education Office.

It was about this time, Autumn 1978, realising the need for good communications throughout the County, that the Association published the first edition of its journal 'Printout'. Although a modest affair, the journal proved to be a major step forward for the Association. The aims of the teachers within the County were drawn to the attention of many people of influence, as well as dispelling the feeling of isolation that many teachers felt.

One of the problems of the Association was how to convince non-computing colleagues that the computer could and should be used in many areas of the curriculum. "Oh, these computers, they are only a gimmick, they will go the same way as the overhead projector." was a comment

heard at one meeting! Many of us in the County made a positive effort to develop the use of computers across the curriculum, encouraging maximum pupil usage. The efforts of the Association, the active support of the Mathematics Adviser and key members of the Education Office enabled us to convince doubting Thomases that we had the expertise and the will to develop computing in Shropshire. The County responded and has put a 32K Commodore PET and printer into most secondary schools, and this at a time of economic cut-back.

Software Development

The Shropshire team is a group of highly skilled and carefully selected

teachers — ie. all those who could be conned into joining! It meets at least once every half term to discuss policy, share ideas and drink coffee.

The group is lucky in having a professional systems analyst as one of its members and he has helped the rest to develop an insight into the problems involved in developing software. The main theme has been that good programs take a long time to write and even longer to de-bug, and that with the best will in the world, it is unlikely that all teachers will develop the necessary expertise to write first class software.

In the present shortage situation, one aim of the group has been to avoid duplication of work, and much discussion has been about the transportability of subroutines. We have looked at the MUSE proposals, but have found them a trifle unwieldy for our situation. We are only writing

PET software and are more concerned with how easy it is for another PETter to use a routine than how the program might be adapted to a different machine.

As far as possible the work is shared with each member of the group looking after a different area of the curriculum. The first aim has been to find and catalogue already available software in each area. However, this has not stopped people writing their own programs and collaboration has paid dividends. One case was when we wanted to plot large numbers on the screen, three members of the group each wrote a subroutine to plot characters of a different size. Line numbers and variable parameters were agreed and we now have three routines from which to choose. Each subroutine prints a large number in XZ on line LZ and in position PZ. Here are two of the programs:—

```

10000 REM**SET NUMBERS**
10010 LZ$="[HMS][23CRD]"
10020 NZ$(1)="[RVS][ ][CRD][CRL][ ][CRD][CRL][RVF][<]"
10030 NZ$(2)="[RVS][ ][ ][CRD][2CRL][ ][ ][CRD][2CRL][2][RVF]"
10040 NZ$(3)="[RVS][ ][ ][RVF][CRD][2CRL][<][RVS][ ][CRD][2CRL][2][RVF]"
10050 NZ$(4)="[ ][CRD][CRL][RVS][2<][RVF][CRD][2CRL][>]"
10060 NZ$(5)="[RVS][ ][ ][CRD][2CRL][ ][ ][CRD][2CRL][2][RVF]"
10070 NZ$(6)="[ ][CRD][CRL][RVS][ ][ ][CRD][2CRL][2][RVF]"
10080 NZ$(7)="[RVS][ ][ ][CRD][CRL][ ][ ][CRD][CRL][RVF][<]"
10090 NZ$(8)="[RVS][ ][ ][CRD][2CRL][ ][ ][CRD][2CRL][2][RVF]"
10100 NZ$(9)="[RVS][ ][ ][CRD][2CRL][ ][ ][RVF][CRD][CRL][<]"
10110 NZ$(0)="[RVS][ ][ ][CRD][2CRL][RVF][ ][RVS][ ][CRD][2CRL][2][RVF]"
10120 CZ$="[CRD][2CRL][CRD][2CRL] "
REM**CLEAR NUM**
10130 RETURN

10500 REM *** PLOTTING ROUTINE ***
10510 PZ=4*NZ
10520 PRINT LEFT$(LZ$,LZ+1);TAB(PZ);CZ$
10530 PRINT LEFT$(LZ$,LZ+1);TAB(PZ);NZ$(XZ):RETURN

```

Note that as the routines use a lot of graphics the listings have been decoded. Cursor control characters and graphics characters are shown in

square brackets, eg. 5CRD is five cursor downs and " means shift quote. Be careful to include all embedded spaces as these are impor-

tant, but often appear insignificant in the listings.

```

10000 REM *** NUMBER PLOTTING ROUTINE ***
10010
10020 LZ$="[20CRD]"
10030 DIM AZ$(20):FOR IZ=1 TO 15:READ AZ$(IZ):NEXT IZ
10040 DIM AZ(9,4):FOR IZ=0 TO 9:FOR JZ=1 TO 4
10050 READ AZ(IZ,JZ):NEXT JZ:NEXT IZ:RETURN
10060 DATA "[3"]","[RVS][ ][ ][CRD][2CRL][ ][ ][RVF][<]"
10070 DATA "[RVS][ ][ ][RVF][CRD][2CRL][2][RVF]"
10080 DATA "[RVS][ ][ ][CRD][2CRL][ ][ ][RVF][CRD][CRL][<]"
10090 DATA "[RVS][ ][ ][CRD][2CRL][2][RVF]"

```



```

10080 DATA "[RVS][ ]"[ ]:", "[RVS][<][RVF][ ]"[RVS][>]",
" [ ] ", "[RVS][ ]"[RVF] ", "[ ] [RVS][ ]"
10090 DATA 1,15,15,12,13,14,14,14,1,2,3,4,1,2,6,5
10100 DATA 7,8,9,14,1,8,10,5,7,8,11,12,1,2,2,2
10110 DATA 1,15,11,12,1,15,10,2

10500 REM *** PLOTTING ROUTINE ***
10520 PRINT "[CHMS]";LEFT$(LZ$,LZ);:PZ=4*NZ
10530 FOR JZ=1 TO 4:PRINT TAB(PZ);AZ$(AZ$(XZ,JZ));:
NEXT JZ:RETURN

```

As an example of how the above PET into a clock. This idea can obviously be developed by, for example, making a

chess clock. The main use however, will be education software for classroom demonstration.

```

1000 REM *** CLOCK ROUTINE ***
1010
1020 LZ=10:FOR I=1 TO 6:T(I)=-1:NEXT I
1030 PRINT "[CLS][2CRD] [RVS]
PLEASE ENTER THE CORRECT TIME[2CRD][RVF]"
1040 INPUT "HOURS = .[3CRL]";H$:
H$=RIGHT$("00"+MID$(STR$(VAL(H$)),2),2)
1050 INPUT "MINS = .[3CRL]";M$:
M$=RIGHT$("00"+MID$(STR$(VAL(M$)),2),2)
1060 INPUT "SECS = .[3CRL]";S$:
S$=RIGHT$("00"+MID$(STR$(VAL(S$)),2),2)
1070 TI$=H$+M$+S$
1080 PRINT "[CLS][8CRD] [37&]"
1090 FOR I=1 TO 6
1100 PRINT " [&]";TAB(37);" [&]"
1110 NEXT I
1120 PRINT " [37&]"
1130 PRINT "[CHMS][11CRD]";TAB(13);"[RVS] ";TAB(25);
"[RVS] "
1140 PRINT "[CRD]";TAB(13);"[RVS] ";TAB(25);"[RVS] "
1150 GOSUB 10000
1160 FOR I=1 TO 6:XZ=VAL(MID$(TI$,I,1)):NZ=I
1170 IF XZ=T(I) THEN 1190
1180 T(I)=XZ:NZ=I+INT((I-1)/2):GOSUB 10500
1190 NEXT I
1200 GOTO 1160

```

The Way Forward

The Association has achieved its primary aim, so what of the future? The need for good communication is even more important. We have just published the fifth edition of our journal, which incidentally is funded by the Authority and published free of charge to all interested parties in Shropshire. The Association is setting up a software exchange, the soft-

ware writing group is growing, and the Authority is to sponsor in-service training at various centres throughout the County. Meetings will continue to be held at regular intervals to examine topics of general interest. The way ahead is clear, to encourage the development of good educational software.

The interesting and important thing about the Association is that it is teacher inspired and teacher led.

The Association has a good working relationship and co-operates fully with the advisory service, but the main impetus comes from the teachers of the County who are making a positive contribution to the development of the service they provide.

Peter Massey
Trevor Lusty

The Shropshire Association for
Computing in Education (S.A.C.E.)

Educational Workshops

College Name and Address	No. of PETs	Courses for Teachers
LONDON		
J. Bennett, Computer Services, Middlesex Polytechnic, Bounds Green Road, London, N11	3	Yes
P.J. Bibb, M.A., L.B. Haringey, Chief Education Officer, 48/82 Station Road, London, N22.	6	Yes
Robin Bradbeer, Polytechnic of North London, Holloway Road, London, N7	20	Yes
A.T. Ley, Moat Mount School, Worcester Crescent, Mill Hill, London, NW7	1	Yes

College Name and Address	No. of PETS	Courses for Teachers
Dr. P. Birkett, Science Department, Bolton Institute of Technology, Deane Road, Bolton.	4	Yes
D. Brown/A. Rose, Rose Bridge School, Holt Street, Ince, Wigan.	1	Yes
Mrs. J. Fiddie, The Deane High School, New York, Junction Road, Bolton, BL3 4NG.	1	—
Alan Goodall, Xaverian College, Lower Park Road, Manchester 14.	3	Evens. esp. Wed.
John G. Harris, Salford Resources Centre, Half Edge Lane, Eccles, Manchester.	2	Yes
Miss A. Howlett, Leigh College, Marshall Street, Leigh, Greater Manchester.	2	No
M. Jasper, Church of England High School, Northfleet Road, Peel Green, Eccles, Manchester.	1	Internally
R.K. Johnson, Science Inspector, Education Offices (1), Crown Square, Manchester, M60 3BB.	50	Many
E. Leonard, Loreto College, Chichester Road, Manchester, M15 5PB.	1	Yes
Paul Murphy, Computer Resource Centre, Manchester Teachers' Centre, Barlow Moor Road, Manchester.	3	Yes
Bernard Phillips, St. Albans Catholic High School, Holmcroft Road, Gorton, Manchester 18.	1	Yes
B.E. Rant, Wythenshawe College, Moor Road, Manchester. M23 9BQ.	1	Yes
T. Robertshaw, Hope High School, Eccles Old Road, Salford, M6 8BJ.	1	—
B.H. Stead, Education Office, Gateway House, Standishgate, Wigan, WN1 1XL.	25	Yes
D.R. Stretch, Chorlton High School, Corkland Road, Manchester, M21 2XD.	1	No
MERSEYSIDE		
G. Catty, Formby High School, Freshfield Road, Formby, Liverpool.	4	Adult Ed. Class
R.P. Stevens, Knowsley Hey Comprehensive School, Seel Road, Huyton, Liverpool.	1	No
Mike Trotter, Microelectronics Advisor, 10 Hatchmere Close, Birkenhead, Merseyside.	1	Runs courses
J.K. Williams, Science Department, Anfield Comprehensive School, Breckside Park, Liverpool.	1	Not Yet
LANCASHIRE AND CHESHIRE		
K.S. Clark, Palatine High School, St. Annes Road, Blackpool, Lancashire.	1	Yes
I.G. Fanning, Holy Cross College, Manchester Road, Bury, Lancashire.	1	Yes
Mrs. E.M. Heseltine, Leigh CE High School, Leigh Road, Leigh, Lancashire.	2	Inservice to Staff
Paul Martin, St. Nicholas RC School, Greenbank Lane, Hartford, Northwich.	1	No
Miss M. Mills, Computer Services Department, Preston Polytechnic, Corporation Street, Preston.	21	Yes
E.J. Mullineux, Wellington Boys School, Moss Lane, Timperley Altrincham, Cheshire.	1	Internal
P.D. Sudlow, Arnold School, Blackpool, Lancashire.	1	No
Gordon Thomas, Maths Department, Chester College, Cheyney Road, Chester, CH1 4BY.	6	Yes
T.L. Thurogood, Blackpool College, Electronics Department, Blackpool, Lancashire.	15	Yes
David Webb, Edge Hill College of H.E., St. Helens Road, Ormskirk, Lancashire, L39 4QP.	10	—
A.M. Woods, Mount St. Joseph School, Hawthorne Road, Bolton, Lancashire.	1	Yes

should also be ignored.

Underground trains have powerful motors, which have been known to destroy the entire contents of hard disks, containing millions of characters of information, so only the masochistic will venture forth with vital, uncopied data in their possession.

Sending disks through the mail obviously has a considerable in-built hazard. Not only must you use cardboard to prevent bending, but wrapping in aluminium foil is mandatory to give some protection against bomb-detecting X-rays. This precaution is said to offer adequate protection against the underground train danger, so I always do this myself when travelling that way.

What about user-errors which can destroy data?

Paradoxically, the operating system designed into the PET can cause disasters, especially for the people who have become so adept at using the system that many actions are automatic.

The users of the Computhink disk drives have a particularly nasty hazard to avoid at all costs. It arises because of the ability to use single letter abbreviations for disk operating commands. Clearly, one of the most frequently-used commands is calling up a directory to see what is on a particular disk. Reasonably enough, the command for this is abbreviated to "d". If you wish to format a disk, thereby destroying its present contents, and preparing it for re-use, you press "f". "d" and "f" are neighbours on the keyboard! — Need I say more?

Disasters

However, the Commodore disk-drive has its own lovable characteristics, which cannot be ignored safely.

One of these is particularly unpleasant in that in the very act of seeking to ensure the safety of your data, you can destroy it in an unguarded moment. Picture the scene. You are about to make a copy of a disk in order that if ill befalls the original, you will be unaffected. So you set about duplicating the disk on drive 0 onto drive 1. Unfortunately you get confused, and duplicate your blank disk onto the one containing your data! This is even easier to do if you have two machines in use, one of which operates on Basic 4.0 and one

on Basic 2.0, because the new command operates by copying from drive 0 onto drive 1, whereas the old command made the disk on drive 1 become equal to that on drive 0.

All is not lost, however. There is a way of avoiding the problem, simply by using a disk-copying program, which will guide you through the process of copying a disk in an utterly foolproof way. The program is provided free by Commodore, so everyone can use it. The danger is of thinking that you are above making such mistakes and becoming somewhat blasé, so that you use only half your mind to carry out a very familiar routine task. Another pitfall for the unwary, of potentially greater seriousness, is the problem of "save with replace". Rumour has it that if you seek to use this command, saving a program by replacing it with another, the results are unpredictable, and can be that the entire disk is unreadable thereafter. My best advice is "Don't use it ever."

The worst hazard I have come across is to ignore one of the warnings which is tucked away in the literature. You will no doubt be aware that it is possible to protect a disk from being written upon by putting a self-adhesive label over the small gap at the side of the disk. Any attempt to write to such a disk will fail, and the error report "write protect on" will be available, provided you notice that the error light is on, and you interrogate the disk to find out the nature of the error. Even the latest manual for the disk drive does not contain the following warning:

IF YOU HAVE ATTEMPTED TO WRITE TO A PROTECTED DISKETTE, IMMEDIATELY REMOVE DISKETTES FROM THE UNIT, TURN OFF THE POWER TO THE DISK UNIT, AND THEN TURN IT BACK ON AGAIN.

The result of failing to carry out this procedure can be devastating: somehow the disk unit remembers that it was trying to write to the disk, and when you try to read, or initialise another disk, it "thinks" that it must carry out the write, and does so, all over the disk, usually destroying the vital directory map, and causing the loss of all the data, or more business for Harry Broomhall, the resurrectionist. The danger is even greater than it seems, for if you try to read

another disk, and cannot, for the reasons indicated, the obvious thing to do is to try to read the "backup" copy. Attempting this will destroy that one too. You then try the "master" backup, congratulating yourself on keeping three copies...!!

There is at present absolutely no protection against this problem, and my view is that it is best never to use the protection tab, because it is likely to do much more harm than good.

What can we do?

So what can we do about disk security in general?

Firstly, it would be wise to make sure that anyone with access to disks knows how to handle them gently, to put them in the cardboard sleeves when not in use, and to put them into a well designed storage system when not in use. If disks are put in piles, allowed to lie about getting dusty, written on with ball-pens, they will not give the service one would like to obtain.

Secondly, be sure that all staff in contact with disks are aware that they are sensitive to magnetic fields.

Thirdly, make sure that the practice of taking two backup copies of disks is scrupulously observed, the rule being that backup copies are required as soon as the disk contains an amount of information which can only be replaced at material inconvenience.

Fourthly, make an absolute rule that the first action carried out when a new program is obtained, and has been shown to load satisfactorily, is to take a working copy of it, or better still two. The master copy is then stored in a different building, to avoid fire risks, together with the first backup copies of important data disks.

This may seem very tedious, but, by having a working backup readily to hand, and a master, in a safe place, you are avoiding the hair-tearing stage of data processing, which is unappealing.

How can we avoid the risk of malicious damage. If you have a disgruntled employee, he or she can do untold damage by passing a small pocket magnet over the disks. You may even find it worthwhile to keep an eye open for office romances which go wrong, since electronic revenge can be devastating.

Reviews

The Luxor Copyholder

This product brings the opportunity to have a sophisticated copyholder in the reach of the people who do not wish to spend a large sum, but would welcome the opportunity to ease the problem of where to put the material from which they are copy typing.

Cheap stands exist, but anyone who wishes to know where they have got to, (and this is particularly the case if you are not a real typist and must look at the keyboard to find the keys!) will find that an automatic linefinding ruler is a great help. The obvious use for this is in correcting typed drafts using a word processor, and in typing long programs into the machine.

Where the Luxor devices score is that the prices have been kept down by using a pneumatic system to move the ruler down the copy, rather than an electric motor. The range, supplied by D.N. Computer Services, West Croft Industrial Estate, Manchester Old Road, Middleton, Manchester, is comprehensive, but the main advantages are that the cursor moves down one line at a time, by foot pressure, the stand will take weights of up to 1 kilo, and the copy can be up to 20mm thick. The whole device is non reflecting, and on an "anglepoise" type of bracket, so that its location above or to one side of the screen may be adjusted to suit your convenience. There is a magnetic paper holder (watch out for disks!) and the larger models will take A3 size paper. Automatic models start at £39.95, and represent good value. Recommended.

JKL Screen Print

This program (£8.00 from Supersoft) carries out a very valuable task in the simplest way. It is a screen-print routine, at present for the 40 column 3032 or 4032 machine, although it is intended to bring out an 80 column version in due course.

The cassette loads a routine into the second cassette buffer. The routine, after initialisation, is called by another SYS command from your

program, whereupon the entire screen contents are printed onto a printer. Readers will know that some printers require different programs from others; this cassette contains versions of the program which cater for both the Commodore printers, and for ASCII printers.

At any time, including in the course of program operation, you may print the screen contents, by pressing keys "J K L", simultaneously. The "JKL" will not be printed. The most interesting ability given by the program is to permit the user to print a transient display, by hitting the three keys, which are of course adjacent. Operation will stop until the screenful of information is printed, and will then resume.

The only caution is that the program must be disabled, (by another SYS command) before the use of either cassette, because it operates through the keyboard interrupt.

An excellent piece of work. As the program loads into that buffer from data statements each time it is run, it is not in danger of being overwritten by BASIC 4 and can therefore be used on the new machines. I understand that it may be possible to specify how many lines will be printed, in a future version.

Label BASIC

Another program from Supersoft, priced at £6.00, this time enabling the use of labels wherever you would usually use line numbers. For example as the destination of GOTO, GOSUB, THEN, RUN, ON...GOTO, ON...GOSUB.

Labels are put between asterisks and are converted, whenever a runnable program is required, into appropriate line numbers. Obviously one should keep copies of the labeled program, to permit changes later. The program is in BASIC, although it loads some machine code into the second cassette buffer.

The only restrictions are that one must avoid using basic keywords like "print"; labels must be placed at the beginning of lines, preferably with a space after them to promote legibility in the final version of the program. If

the line-number created to replace a label is longer than the label, you may run out of space, in which case the program tells you. This may be avoided by choosing labels which are longer than potential line numbers. If you use a utility with search and replace to insert the labels in place of abbreviations, there will be no trouble in using long labels, and they tend to be more meaningful!

Very useful!

Block Renumber

Another very useful utility from Supersoft, priced at £12.00, this time enabling you to renumber part of your program. This one, however, is different. It will renumber part of your program, with you choosing the start and finish of the old lines to be renumbered, and the increment, but most importantly, the starting point of the renumbered lines.

The process must be handled with some care, but the program helps you to take the necessary steps. It is possible for the renumbered lines to cause an overlap of lines which have not been renumbered, and those which have. This causes the message "Beware overlap" to be displayed. The instructions tell you how to cure the situation. It is necessary to use the Templeton merge technique to load the program, which contains some overlength lines. Alternatively you may use a disk merge program. If there is insufficient room on a line for the line number, you are given a warning of this; similarly if there are references to line numbers which do not exist.

The program will be of particular interest if you like to commence all blocks of a program at separate points, and have certain conventions which you like to observe, such as menus at particular line numbers, and each subroutine to start at a 1000 increment.

BASIC Edit

This utility, costing £6.00 from Supersoft, is a really advantageous one, enabling you to retain the

variables and continue a program run after you change a program, even by adding up to 1K of new statements. It is true that many programs can be rerun without inconvenience, but if you have substantial data input, particularly from cassette files, the ability to edit the program and re-run from where you had reached is invaluable.

The second cassette buffer is again deployed, although the code is relocatable.

The precautions to be taken are spelt out carefully in the instructions, and I can thoroughly recommend the program as a timesaver.

Linker

This program, from Dovetail Computer Systems, priced at £35.00, is a

rather expensive way of injecting discipline into your programming, by enabling you to bring subroutines into your programs, by name, automatically. Thus modular programming is encouraged, by being made easier.

Subroutines are merged fully, by line number, and up to 3 subroutines can be handled simultaneously. A subroutine can call a sub-subroutine, and if you change a subroutine, you can revise programs containing that subroutine simply.

While writing a program, you insert a special REM, on a separate line, showing the name of the subroutine for incorporation, thus: "rem*include sub.printout".

In the subroutine, you include a range statement, thus: "rem*range 10200-10400". This enables the

Linker program to substitute future versions of the subroutine, quite automatically.

The program works well, and since it can build up 10 programs at once, three subroutines at a time, the prolific programmer, who is also sufficiently disciplined to keep detailed records of the line numbers of subroutines, and the variables they use, will be able to assemble the programs more quickly and efficiently. Whether this is for you will depend on the value you place on your time, and whether you think you will be better served by putting the money towards one of the proliferating programming aids which offers a less sophisticated disk merge as only one of a large range of capabilities, at a similar price, or in the case of Basic Aid (MP096 for Basic's 1 and 2, and MP107 for Basic 4), much less!

DMS

by Christopher Young
of Macmeat Ltd.

DMS (Data Management System) is specially designed to reduce the vast number of hours normally taken up with tedious clerical tasks. It allows any 32K PET (3000, 4000 and 8000 series) to act as an ultra-efficient filing cabinet cum calculator. Due to a co-operative effort amongst software houses it now links to Wordcraft, Wordpro and Visicalc.

Its many uses include:

Student records/exam results analysis/Attendance records

Stock records

Client/personnel/patient records

Sales figures/client records

Mailing lists

Agency records (dating, employment, estate agents, etc., etc.)

Job costings

Clubs: membership and subscription records.

My own particular application of the program is for client and stock records and to print my price lists.

Powerful in its own right, it also links into Worpro and Wordcraft, the PET word processing packages. Anyone who has names and addresses and wishes to select batches of people to write to can store these on DMS, then transfer them automatically onto Wordpro or Wordcraft letters. The main advantage of storing informa-

tion on DMS first is that all the selection facilities of DMS can be applied prior to producing the letters, eg. ask DMS to select all the clients who live in Yorkshire or Birmingham who didn't purchase between particular dates and/or have a credit limit in a range of a particular amount, and write only to those people who fulfill those criteria. I can appreciate Comsoft's claim that hours and hours of typing and machine time are saved by this process.

Back to the main core of the program: perhaps the easiest way of describing its features is to show a few examples of typical applications. Remember, DMS will hold any information you require:

In Colleges:
Student Name
Street
Town
County
Telephone Number
Date of birth
Employer
Classes
Exam results
Attendances
Results
Course tutor
Fees
Extras
Payment details
Comments,
Etc., etc., etc.

DMS will display on screen or

print any student's record in 2 seconds, or it can search on several parameters, eg. all those who study French, passed Latin, live in Yorkshire or have John Doe as course tutor. Using the wordprocessing link, names, addresses and any other information can be transferred into letters. Fees can be calculated and 'extras' added on. In cases where VAT is applicable, DMS will work this out on an individual basis, add it up to amount payable and even tell you what your quarterly VAT returns are.

It will delete any fields not relevant before printing, or sort the records on any criteria, eg. alphabetic order, fee order, etc.

STOCK RECORDS

Stock number
Stock name
Stock description
Supplier
Alternative supplier
In stock
On order
Cost price
Selling price
VAT per unit
Price + VAT
Profit
Comments
Etc., etc., etc.

DMS can search on any single criteria or on several, eg. any product where the instock price is in a range of a certain amount or where the profit margin is less than 15% of cost

price, etc. DMS will work out automatically the VAT, price + VAT and profit lines, inventory value, etc., etc. It can be sorted prior to printing on any line of information, eg. value order.

CLIENT'S RECORDS

Client's name
Company
Street
Town
County
Contact
Our representative
Credit limit
Models purchased
Service call
Next call
Invoices paid
Outstanding
Comments
Etc., etc., etc.

SEARCHING

DMS can make detailed searches on combinations of information stored, then, if required, print the records out or transfer any information into word processed letters so that every letter is individually addressed and contains other individual information.

DMS can find any record in two seconds by searching on the key field; it can also search on any other field; it can also search on any other field or on several parameters at a time. It can search for any records where a number is less than, greater than, equal to, not equal to, in a range of; and will search on parts of fields or for specific information stored in a position in a field.

SORTING

DMS will sort (in machine code) all the records on file using any field as a base, eg. in numeric or alphabetical order. All the selection facilities can be applied prior to sorting if required.

A special screen scrolling feature can be used after sorting which allows the user to browse through records, moving back and forth through the file.

Labels can be sorted prior to printing.

PRINTING AND LINKS TO WORDCRAFT AND WORDPRO

DMS has a variety of print formats. Either the whole record may be printed, or selected fields or the information may be printed in columns, especially useful when processing

numeric information. Because my prices change rapidly, I cannot have a printer produce price lists for me. However, DMS can print lists of up to date prices which are then inserted into my own folder. Using the links with Wordcraft and Wordpro, the possibilities are limitless.

Any information stored in DMS can be transferred to the appropriate 'gaps' in Wordcraft and Wordpro. Imagine a list of people to whom you wish to write on a regular basis, eg. a client list: these names and addresses and contacts can be stored in DMS, then should you want to write to all the clients who haven't bought from you between certain dates, and/or who live in Bognor and/or owe money and/or ought to be stocking your latest product, DMS will select the relevant information including names, addresses, contact names, amounts owed or whatever, and transfer them *automatically* onto a fill-file which Wordcraft and Wordpro use as inserts for the letters you create. Using this facility means that typing hours are cut to a minimum. The benefits to anyone who ever wished for a mailing list are tremendous. If I have a special offer on one particular product, I can select those clients from my list who may be interested and write to them specially.

DMS has its own label printing program, allowing the user to load any label stationery and print sorted labels up to 5 wide. It can also print out all changes/transactions, thus providing a full audit trail.

NUMERIC FACILITIES AND LINKS TO VISICALC

Any of the following numeric processes can be applied to information.

- + × ÷ any fields with other fields or with constants; replace, move, delete and zeroise data. Results of the calculation can be stored either in fields reserved for the totals, or in any of 9 temporary work fields which can be printed out without affecting information on the disk. Up to eight calculations can be dealt with sequentially at one pass of the file, on either all the records on file, or on preselected records according to specific criteria. Sophisticated management figures can then be produced, eg. exchange rate calculations, financial projection etc., as well as routine sales figures, VAT fees etc. I can produce across the board price changes or have prices automatically

quoted to incorporate discounts, etc.

Compssoft have just produced a link program to transfer information into Visicalc for even more sophisticated numeric processing.

Thousands of records, either whole file, or preselected ones (eg. those who joined since a certain date/live in Yorkshire/get 5% discount etc.) are processed *automatically* at one time.

OTHER FEATURES

- 1 Password protection.
- 2 Transaction logging/audit trail.
- 3 Add or remove fields of information as required.
- 4 Merge information from different data files.
- 5 One program can do a number of jobs on the same PET, eg. at one college in Coventry one department keeps not only student records on DMS but also analyses exam results, keeps the film library records, does the departmental accounting, inventories and company visits files!
- 6 I found the manual very easy to read, especially the introductory chapters.
- 7 Menu driven.
- 8 Written in Basic with machine code sorts and searches.
- 9 Runs on 3000, 4000 and 8000 PETs.
- 10 Up to 5000 records on the 40 column PET and up to 13,600 on the 80 column PET.

Written by Compssoft of Guildford, a small PET-oriented software house, DMS already has over 500 end users. Although most Commodore dealers stock DMS, Heather Kearsley at Compssoft (0483 39665 or 0483 505918) is pleased to discuss your individual application in detail and will recommend a local stockist.

Overall, once the user has mastered the facilities available he or she has a tremendously powerful tool which can form the basis for a number of office procedures. It's certainly easy enough for anyone to use competently within an hour of starting, yet the hours of routine clerical time saved make it an invaluable aid.

Price £170, plus £30 for Visicalc or Wordpro/Wordcraft links from Compssoft, Great Tangleby Manor Farm, Womersley, Guildford, Surrey.

DMS is a Commodore Approved Product

For those of you who already possess a disk unit and know how to use it, I apologise from the start. This article is intended to show people who do not own a disk unit how easy it is to use one, and also for people who already have one but only know how to load their application software, and wish to know more.

Starting from Scratch

Once the disk unit is unpacked and sitting happily next to the PET, and connected via a single cable (this is known as the PET to IEEE cable), we are at the stage where both units can be switched on (make sure they're plugged in!). Switch the PET on first, and then the disk unit, making sure you have no diskettes in either of the drives. Depending on the disk unit you have, one of three things will happen:—

- 1 For either a 2040 or a 3040 disk drive, all three of the red LED's (that's the three lights on the front) will flash on and then go off again. As the only difference between a 2040 and a 3040 is the number on the front, in future I will refer to both by just using 3040.
- 2 For a 4040 disk drive the LEDs behave in the same manner, and the drive motors will run and the read/write head is positioned (a noisy process until you get used to it!).
- 3 For an 8050 disk drive, the two LEDs above the drives flash green and then go off. The central LED flashes red and green, and then remains on the screen.

Let us assume that the disk unit has powered up correctly. The first thing you'll want to do is to put a diskette in and see what happens. There is a diskette supplied with the disk unit, which contains several useful programs, so we'll take a look at this first and see what's on it.

For a 3040 or 4040 disk drive, carefully insert the diskette, after taking it out of its pocket, into the right hand drive (known as drive 0, the left hand one being drive 1) and type the following command:—

OPEN1,8,15,"I0"

Follow this by hitting the RETURN key, and as the motor starts running slowly close the door on the drive. For an 8050 this isn't necessary — simply push the diskette into the drive until it locks, and then gently press the drive door down until it locks into place. As you do this you can hear the drive motor spinning briefly.

Once the diskette has been inserted, and the above procedures implemented, we can have a look at the contents of the diskette. As you may have guessed by now the 8050 is a separate entity from the others, and as it is usually used with an 8032 PET it can use a different set of commands. I stress the 'can use' because everything in this article applies to all disk drives, and will work with any of them.

So, to see what is on the diskette, type the following command below:—

LOAD"\$0",8

Follow this by hitting RETURN, and then type:—

LIST

Again, hit the RETURN key. In subsequent articles I'll explain easier ways of looking at the DIRECTORY (which is a list of everything on the diskette, and is what you've just called up onto your screen) which do not destroy any programs that already exist in your PET — this method unfortunately does. Don't worry — your diskette is still safe!

If we now examine the LISTing we'll see what each of the various parts of the directory mean.

First line:—

0 "UTILITY DISK ",AB 2C

The left hand number is the number of the drive you're using, the name is the name of the diskette, which can be anything up to 16 characters long. If you've used less, the rest is filled up with spaces. The characters AB can be any two letters or numbers, and the final two refer to the DOS type. DOS stands for Disk Operating

System, and on a 3040 you'll see nothing there. On a 4040 you'll just see the number 2, and on an 8050 you'll see 2C. This simply means that the 8050 is the most advanced disk drive that Commodore has yet introduced. If your PET doesn't display exactly the above, don't worry. As long as it's something like the above you're okay.

Apart from the drive number, the rest of the line is in reverse field. For the second and subsequent lines, you'll get something like:—

12"NAME" PRG

The first number is the length of the program in BLOCKS. The name is the name of a file, and as in the diskette name this can be up to 16 characters long. The last three letters refer to the file type. A block is 256 characters, but two of these are used by the disk to link blocks together, so the maximum number of characters in a file is 254 times the number of blocks. The file type can be one of the following:—

DEL DELETED
PRG PROGRAM
SEQ SEQUENTIAL
USR USER
REL RELATIVE

The DEL one only appears in a directory on rare occasions, and usually indicates that you, or a program, has done something incorrectly! The REL one only appears on either a 4040 or an 8050. I'll say more about these, and the others, in later articles.

A new diskette

Remove your utility disk from the disk unit and return it to its pocket. RULE 1: A diskette must be in the disk unit, or in its protective pocket.

Now take a completely new (blank)

The last line of a directory will look something like:—

400 BLOCKS FREE

This line tells how much space is left on the diskette, in this case, 400 blocks.

diskette and insert it into drive 0. If you are using a 3040 or a 4040, do not close the disk door yet.

OPEN1,8,15

PRINT#1,"N0:NAME,AB"

Follow both lines by RETURN of courses, and when the disk motor starts running then you can close the door. NAME can be up to 16 characters long, and will become the name of the diskette. AB can be any two letters or numbers, and is referred to as the ID of the diskette (for Identity). Try to keep some order to the identification codes used, as it is important to have a unique code for each diskette used, wherever possible.

As the motor starts running, the red LED above the drive will come on (or on an 8050 the green LED). This process will take a couple of minutes, and when it's finished the light will go off again, and the motor will stop. If the central LED comes on (or goes red on an 8050) something has gone wrong. Retype the PRINT#1 line and try again (there's no need to open the

drive doors to do this). If the same thing goes wrong, try another diskette, and if this still fails it's worthwhile calling your dealer and ask him to check your disk unit.

We'll assume that nothing went wrong, and have a look at the diskette you've just 'made'. Load the directory as before (LOAD"\$0",8) and LIST it. What you will see is something like the first and last lines of the directory from the utility diskette, with no PRG as this is a brand new diskette, and has had nothing done to it other than being named. Consequently the number of blocks free (amount of space on the diskette) will show you how much information you can get on a whole diskette. This number, as you might surmise, depends on the disk unit. For a 3040 it will be 670, for a 4040 it's 664 and for an 8050 it's 2052. This means that on an 8050 you can store 2052 times 254 characters, or in other words, 521,208. So, if you were to use both drives, you could store over a million characters of information!

Now to check that both drives of

the disk unit are aligned properly. Remove the diskette from drive 0 and place it in drive 1 (the left hand one). Type the following sequence of commands, hitting RETURN after each one of course.

NEW

OPEN1,8,15,"I1"

LOAD"\$1",8

LIST

What we're doing here is first of all removing the old directory from PET's memory (NEW), initialising the diskette in drive 1 (don't forget to close the door as the motor starts running if you're using a 3040 or a 4040), LOADING the directory from drive 1 and then LISTing it to see what it looks like. With the exception of the drive number as the top left hand entry it should look exactly the same as from drive 0. If it isn't then you'd be well advised to consult your dealer.

That's all for now. More next time!

PETALECT. An all-round computer service.

PETALECT COMPUTERS of Woking, Surrey have the experience and expert capability in all aspects of today's micro-computer and word processor systems to provide users, first time or otherwise, with the Service and After Sales support they need.

COMPUTER REPAIRS AND SERVICE

If you're located within 50 miles of Surrey, PETALECT can offer FAST, RELIABLE Servicing with their own team of highly qualified engineers.

24 hour maintenance contracts available. Our service contracts start at around only 10% of your hardware cost per annum for on-site, or if you bring it to us at our own service dept., it costs only £25 plus parts. Representing real value for money.

MICRO COMPUTER SUPPLIES

PETALECT can supply the great majority of essential microcomputer-related products promptly and at really competitive prices. Such items as:-

TAPES●PAPER●FLOPPY DISKS●PROGRAMMES FOR BUSINESS●SCIENTIFIC OR RECREATIONAL APPLICATIONS●MANUALS●COMPUTER TABLES●DUST COVERS RIBBONS●TOOL KITS●PRINTERS●ELECTRONIC INTERFACES WHICH ARE PETALECT'S SPECIALITY.

If you want to find out more about what we can and would like to do for you, why not give us a ring on Woking 69032/21776.

SHOWROOM

32, Chertsey Road, Woking, Surrey

We're worth getting in touch with.

PETALECT
COMPUTERS

SERVICE DEPT.

33/35 Portugal Road, Woking, Surrey

Machine Code Arithmetic in the PET

Michael J. Smythe University of Edinburgh

Plenty of articles in CPUCN and elsewhere have shown how to use machine code to modify the PET's operating system or to speed up graphics or communication with peripheral devices. The Commodore User Manuals tell how to use USR to put a number into a floating-point accumulator...but what then? This is an introduction to linking BASIC with the ROM subroutines for floating-point arithmetic. Speed gains here are useful but not dramatic, because microcomputers have to do floating-point arithmetic by software, unlike big computers.

A few preliminaries. All memory locations quoted refer to "new ROMs" or BASIC 2; if you still have "old ROMs" or BASIC 1, refer to published listings of equivalents, eg. in 'The PET Revealed'. All numbers quoted in machine code are hexadecimal, so I have not bothered to put a \$ in front of them each time. Indirect addressing is frequently used: (AB) denotes the number whose address LO, HI bytes are held in zero-page locations AB and AB + 1.

The PET stores floating-point (non-integer) numbers in 5-byte normalised floating binary format...no need to worry about the details now, but they're in the User Manuals and more fully in 'The PET Revealed'. Scalar (non-array) variables are preceded by 2 bytes containing the 2-character variable name. The elements of an array are preceded by a header of 7 or more bytes. All we need to operate the ROM arithmetic routines are the addresses of the *first* of the 5 bytes of each number concerned.

PET handles floating-point operations in two 6-byte registers, floating accumulator #1 (FAC #1) starting at 5E and floating accumulator #2 (FAC #2) at 66; there is a third accumulator for temporary storage at 54. The subroutines take care of the expansion of 5-byte numbers into 6-byte accumulator form. Operations that transfer 5-byte numbers from memory to the accumulators require the (LO, HI) address of the first byte to be loaded into the (A, Y) registers. So, in order to operate on variables defined in BASIC, we need to be able

to find their storage locations.

PET organises its memory with BASIC programs starting at (28), scalar variables at (2A), and array variables at (2C). So to locate a particular scalar variable we set a 2-byte pointer to (2A) and move it up the variables area in steps of 7 until the wanted character(s) (variable name) is found; then we move the pointer 2 bytes so that it points to the start of the wanted 5-byte number.

Similarly, array elements are found by starting a pointer at (2C) and checking the first two characters of the array header. If it is the wanted array, we can get the number of elements from bytes 6 (HI) and 7 (LO) of the header, move the pointer 7 bytes to the start of the first array element, then by successive 5-byte steps to the remaining elements. If the first array encountered is *not* the wanted array, then bytes 3, 4 of its header will tell us the (LO, HI) start address of the next array to be inspected, and so on. If multidimensional arrays are involved, we can get the number of dimensions N from the 5th byte of the header, which is then $5 + 2*N$ bytes long. See 'The PET Revealed', because the Commodore Users Manuals don't mention this possibility!

Having got the pointer to the start of the wanted number, we load the (LO, HI) bytes into (A, Y) and are ready to use the ROM subroutines. DAAE loads FAC#1 from memory D998 loads FAC#2 from memory DB08 copies FAC#2 into FAC#1 DB18 copies FAC#1 into FAC#2 DADC stores FAC#1 into memory: (LO, HI) bytes of location in (46, 47).

Also D26D transforms a 2-byte integer, whose (LO, HI) bytes are in (Y, A), to a floating-point number in FAC#1.

An alternative way of finding names variables in memory is to let PET's subroutines do the work! Place the ASCII values of the characters representing the variable, for example A(1) would be 41 28 31 29, into memory starting at some location NAMELO, NAMEHI. Dump the existing bytes in 77, 78 into temporary storage, and load NAMELO

into 77, NAMEHI into 78. Then JSR CF6D will find the named variable and place the LO, HI bytes of its starting address in 5C, 5D, and these values could be stored for future use; or JSR CE0F will place the value of the variable in FAC#1. In either case the original values from 77, 78 (pointer to next BASIC character) must then be re-loaded. When array variables are concerned, the quickest procedure is to find the start of the header by search subroutine CF6D, move a pointer 7 steps to the first element of the array, and by 5 steps to successive elements. Note the previous warning about multidimensional arrays.

Using these subroutines we can get BASIC variables into the floating point accumulators, operate on them, and transfer the result back to memory. The result is normally found in FAC#1, and there is not subroutine to store in FAC#2 back to memory because the contents of the second accumulator may have been mangled by the preceding operation.

The following subroutines operate on the contents of FAC#1 and leave the result in FAC#1:

```
D725 add .5 DEA1 change sign
D8F6 LOG DEDA EXP
D9EE multiply by 10 DF7F RND
DA0A divide by 10 DFD8 COS
DB45 SGN
DB64 ABS DFDF SIN
DBD8 INT E028 TAN
DE5E SQR E08C ATN
```

The following subroutines require one number in FAC#1 and the second either in FAC#2 or in memory (LO, HI in A, Y), leaving the result in FAC#1:

```
D773 add FAC#1 to memory
D733 subtract FAC#1 from memory
D937 multiply FAC#2 by FAC#1
DA1E divide FAC#2 by FAC#1
DE68 power: (FAC#2) | (FAC#1)
```

Finally, two subroutines leave the result of a comparison in the accumulator A:

```
DB37 A = 01, 00, FF according as
FAC#1 , =, 0
DB67 A = 01, 00, FF according as
FAC#1 , =, memory.
```

Continued on page 20.

A Simple Program in BASIC or Machine Code

How fast is Machine Code?

I often get asked how much faster than BASIC a machine code program works. The answer varies between 300 and 5000 times as fast depending on what you are asking the program to do.

To bring home this vast increase in speed, this article describes a simple program, "Counting on the screen." First the method or 'Algorithm' is described, then the program is coded in BASIC and finally in Assembler code. The BASIC program shown uses PEEK and POKE rather than print to allow a direct comparison to be made with the assembler code version.

The various routines in each version perform identical operations, so apart from illustrating the speed difference, the comparison between the BASIC and Assembler text will help some of you who are just starting on Assembler code programming.

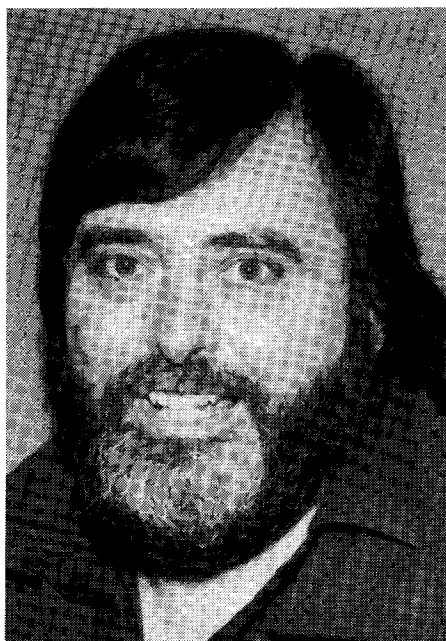
As it stands at the moment the program is written for an 80 column PET, although the necessary changes for 40 column machines are contained in the listings. It will work on any version of BASIC.

A counting Algorithm

- Set up 6 zeros on the screen
- 'Pick up' the left hand digit and add one to it.
- If the result is less than '10', put it back on the screen and go to step b.
- Replace the '10' by zero, put it on the screen.
- 'Pick up' the next digit, if all done then quit
- Add carry (one) to digit, go to step c.

The BASIC Program

The BASIC program is laid out in blocks, each headed by a REM title that corresponds to the algorithm steps. The blocks are separated in the



listing by the method I described in a previous issue of CPUCN. That accounts for the REM" at the end of each block.

In block 1000, SC is the rightmost digit position on the screen and N is the number of digits minus one. So SC+N will be the position of the left most digit. You can vary the number of digits by changing the value assigned to N in line 1040. In the listing shown N = 5, ie the program will count to a million, as does the Assembler version. But be warned, the BASIC program will take over seven hours to complete the count. If you change line 1040 to read N = 2, then the program will count to a thousand, which is more convenient to time.

In block 2000, successive positions on the screen are POKEd with the PET screen code for zero, which is 48.

Block 2100, by setting D to the same value as N, picks up the left hand digit and adds one to it.

Block 2200 checks to see if the screen code for the digit is less than 58. The screen code for 9 is 57 so if the current value is 58, then the count has gone too far.

Block 2300 adjusts a count that has gone too far, by setting the digit to zero.

In Block 2400, by setting D to one less than it's current value, SC + D will now refer the next digit to the right. If D = 0 then the right most

digit will be dealt with. If D is less than zero, then the count is finished.

In Block 2600, the carry value of one is added. By going back to block 2300, the program uses existing code to check out whether this digit has now 'gone over the top'.

The Assembler Code Program

For those of you not too familiar with Assembler code, use the BASIC program as a map. You will see that REM statements in BASIC are replaced by comments following a semi-colon in Assembler. I've made these comments pretty much the same as the BASIC REM statements and divided the modules of Assembler code in much the same way as the BASIC modules.

As with the BASIC program, some preliminary work is done, setting up the value of the right most screen digit location as SCREEN and the number of digits to be counted as NUMBER. The BASIC program clears the screen. I could have done this in assembly by using the ROM subroutines, but have left it out to keep things as simple as possible.

Routine SETUP puts 6 zeros on the screen. It uses the X register as a modifier of the location SCREEN which is the right most digit.

Routine ADDONE picks up the leftmost digit and adds one to it.

Routine CHKOVR tests the value after adding one to see if it has gone beyond 57, the PET screen code value for 9, by comparing it with the next screen code value 58. If it is 58, the program jumps to the ADJUST routine. If it isn't the new digit value is stored on the screen and the AD-DONE routine obeyed again.

Routine ADJUST replaces the value of 58 by 48, the PET screen code for zero.

Routine NXTDG decreases the X register by one. Because it is being used as a modifier to the location of the right most digit, decreasing X causes the program to process the next digit to the right of the one it's just been working on. When X is zero, then the digit location is not modified and so the rightmost digit is processed. If X

A Simple Program in BASIC or Machine Code: continued from p.19

is less than zero, ie. minus, after being decreased by one, then all digits have been processed and the count is over. Otherwise NXTDGI picks up the next digit for processing.

Routine ADDCAR adds one, the carry from the previous column, to the digit just picked up. The current value of the digit must now be checked to see if it is greater than 9, so the program jumps back to the check routine, CHKOVR.

This program will count to one million, displaying the count on the screen, in just under 30 seconds! You can check that yourself by the following little BASIC programme.

```
10 T = TI
20 SYS 826
30 PRINT "THE COUNT
TOOK"; (TI - T)/60;
"SECONDS"
```

Conclusion

In this particular instance, machine code proves to be around one thousand times faster than the equivalent BASIC.

It is a useful technique to write out the steps involved in every day English, the algorithm. Once you do that, as you can see, you can program without too much trouble in BASIC, Assembler or even Pascal.

Teachers in those schools who have taken up Commodore's Assembler, Pascal and Lisp offer might like to produce equivalent programs in all three languages and time them. Let me know how you get on.

Mike Gross Niklaus

Machine Code Arithmetic: continued from p.19

Armed with these subroutines, we can translate any of the arithmetic operations of BASIC into machine language, and hand the results back to BASIC. As an example, a program to find the mean of a large array is given below.

In order to illustrate both methods of finding where variables are stored in memory, the variable 'M' (the mean) is found by subroutine CF6D, assuming that M=0 has been included as a statement in the BASIC pro-

```
COUNT.....PAGE 0001
LINE# LOC CODE LINE
0001 0000 ;SIMPLE COUNTING PROGRAM IN ASSEMBLER
0002 0000 ;MIKE GROSS NIKLAUS
0003 0000 ;15.4.81
0004 0000 ;FOR COMPARISION WITH BASIC VERSION
0005 0000
0006 0000 SCREEN = 33764 ;33264 FOR 40 COLUMN
0007 0000 NUMBER = 5 ;NUMBER OF DIGITS IN COUNT
0008 0000 ;
0009 0000 * - 826 ;PUT PROG IN 2ND CASSETTE BUFFER
0010 033A ;
0011 033A A2 05 SETUP LDA #NUMBER ;SET UP 6 ZEROS ON THE SCREEN
0012 033C A9 30 LDA #48
0013 033E 3D E4 83 SETUP1 STA SCREEN,X
0014 0341 CA DEM
0015 0342 10 FA BPL SETUP1
0016 0344 ;
0017 0344 ;
0018 0344 A2 05 ADDONE LDA #NUMBER ;PICK UP LEFTMOST DIGIT
0019 0346 FD E4 83 LDA SCREEN,X
0020 0349 18 CLC ;ADD ONE TO IT
0021 034A 69 01 ADC #1
0022 034C ;
0023 034C C9 3A CHKOVR CMP #58 ;COMPARE WITH '10'
0024 034E F0 06 BEQ ADJUST ;IT IS? GO AND ADJUST
0025 0350 3D E4 83 STA SCREEN,X ;IT ISNT! SO STORE DIGIT
0026 0353 4C 44 83 JMP ADDONE ;ON SCREEN AND GO ROUND AGAIN
0027 0356 ;
0028 0356 A9 30 ADJUST LDA #48 ;REPLACE '10' BY '0'
0029 0358 3D E4 83 STA SCREEN,X
0030 035B ;
0031 035B CA NXTDG DEB ;MOVE ON TO NEXT DIGIT
0032 035C 10 01 BPL NXTDGI ;MORE DIGITS, YES ...
0033 035E 58 RTS ;NO! SO QUIT
0034 035F 3D E4 83 NXTDGI LDA SCREEN,X
0035 0362 ;
0036 0362 18 ADDCAR CLC ;ADD ONE AS CARRY
0037 0363 69 01 ADC #1
0038 0365 4C 44 83 JMP CHKOVR ;GO AND CHECK NEW VALUE OF DIGIT
0039 0368 ;
0040 0368 .END
```

ERRORS = 0000

SYMBOL TABLE

SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE
ADDCAR	0362	ADDONE	0344	ADJUST	0356	CHKOVR	034C
NUMBER	0005	NXTDG	035B	NXTDGI	035F	SCREEN	83E4
SETUP	033A	SETUP1	033E				

END OF ASSEMBLY

```
10 REM SIMPLE COUNTING PROGRAM IN BASIC
20 REM MIKE GROSS NIKLAUS
30 REM 15.4.81
40 REM FOR COMPARISION WITH M/C VERSION
99 REM "
```

The Basic Program

```
1000 REM PRELIMINARIES
1010 PRINT "C" : REM CLEAR THE SCREEN
1020 SC = 33764 : REM POSITION OF RIGHTMOST DIGIT ON 80 COLUMN SCREENS
1030 REM IT WILL BE 33264 FOR 40 COLUMN SCREENS
1040 N = 5 : REM NUMBER OF DIGITS - 1
1099 REM "

2000 REM SET UP SIX ZEROS ON THE SCREEN
2010 FOR I = SC TO SC + N
2020 POKE I, 48 : REM PET SCREEN CODE FOR ZERO
2030 NEXT I
2099 REM "

2100 REM PICK UP LEFTHAND DIGIT, ADD ONE TO IT
2110 D = N
2120 A = PEEK(SC+D) : A = A + 1
2199 REM "

2200 REM CHECK LESS THAN 10, IE PET SCREEN CODE 58
2210 IF A < 58 THEN POKE SC+D,A : GOTO 2100
2299 REM "

2300 REM REPLACE BY ZERO, PUT IT ON SCREEN
2310 POKE SC+D, 48 : REM ASCII CODE FOR ZERO
2399 REM "

2400 REM PICK UP NEXT DIGIT, IF ALL DONE THEN QUIT
2410 D = D - 1
2420 IF D = -1 THEN END
2430 A = PEEK(SC+D)
2499 REM "

2500 REM ADD CARRY TO NEXT DIGIT, GO AND CHECK RESULT
2510 A = A + 1
2520 GOTO 2200
2599 REM "
```

READY.

gram; while the start of the array is found by previously arranging that it is the first array dimensioned, so that its first element starts at (2C) + 7.

This program, residing in the second cassette buffer and called with SYS826, takes about 0.6 sec to find the mean of a 1000-element array, compared with nearly 6 sec in BASIC. The speed gain is less spectacular than in some other machine-language operations, because of the slow software floating-point routines, but it could be useful in interactive and graphics programs involving a lot of arithmetic.

I hope that this introduction to PET's arithmetic may "break the ice" for users who, like myself, have at first hesitated to explore the mysteries of the floating point accumulators.

FOOTNOTE: While browsing through the subroutines listed in 'The PET Revealed', you may come across the "joke" at D745. If your sense of humour expires at this point, try:

```
POKE6502,76:POKE6503,69
POKE6504,215
SYS6502
```