disk USER

BBC MICRO
MODEL B
MODEL B+
MASTER 128

ALL DFS FILING SYSTEMS

In the Magazine

UTILITIES REVIEWED

DISCUSSION

CHOLO SOLUTION

On the Disk

LIFE SAVER – BACKING-UP IN THE BACKGROUND

FULL SCIENTIFIC CALCULATOR

INCREDIBLE SHRINKING SOFTWARE – THE ULTIMATE COMPACTER

BLAST YOUR WAY OUT IN TAZMAN

LEARNING MATHS MADE FUN

CHOLO EDITOR

DOUBLE CATALOGUE

FROM THE PUBLISHERS OF A&B COMPUTING

disk USER

D was a Drummer and beat out the bass

IN THE MAGAZINE

DISK DOCUMENTATION

DISK INSTRUCTION

DISK INSTRUCTIONS

To get the best from your copy of *Disk User*, please carefully read the instructions below. We have made *Disk User* able to run on a very wide range of systems.

One point to note is that **we strongly recommend copying the disk on to a blank formatted disk before you use it.** You should use this copy as your working copy, and keep the original as the back-up. Many of the programs require to write to the disk, and doing this will diminish the usefulness of the original.

40 Track Drive Systems

Disk User is supplied on a 40 track disk so will work on any 40 track BBC micro system (at least, any that we know of!) straight away. Remember to make a working copy before use.

40/80 Switchable Drives

If you have this sort of drive, you can use *Disk User* straight away with the drive switched to the 40 track setting; don't forget to make a copy for normal use. However, you may wish to copy the disk on to 80 track format, in which case, with a single drive, you should follow the instructions for 80 track systems.

With two switchable drives, or one switchable drive set to 40 track and an 80 track drive (or even a 40 track drive and an 80 track drive), you can easily copy *Disk User* on to 80 tracks; put *Disk User* into drive 0 (40 tracks)

and a blank formatted 80 track disk into drive 1 (80 tracks) and type:
COPY 0 1.* <RETURN>
Here <RETURN> means hitting the return key. You can set the boot option to drive one by typing:
*DRIVE 1 <RETURN>
*OPT 4 3 <RETURN>

80 Track Drives

Because *Disk User* is supplied as a 40 track disk, 80 track disk drives have to double-step through the disk. Probably the most convenient thing to do is to copy *Disk User* on to 80 track format. This can be done in two ways.

If your filing system allows double-stepping, we recomend using the system's own command. As a general rule, built-in 40-to-80 track converters should be used where available; the documentation for your filing system or utility ROM will give full instructions, and we give suggestions for some better-known systems further on).

Not all filing systems have facilities for double-stepping; Acorn's DFS is one such system. To overcome this, a program called CHANGE is supplied on the *Disk User* disk in a section which can be accessed by 80 track drives.

Using CHANGE

Insert *Disk User* into an 80-track drive (or 40/80 switched to 80-track) and type:
*CHANGE <RETURN>
The program will prompt you to insert a pre-formatted blank 80 track disk when it is ready to write to it (you will have to swap back and forward between the two disks several times if you are using only one drive). Once this is completed, you can use the newly created 80-track version of *Disk User* and keep the original as the back-up.

Our suggestions on how to use *Disk User* on some popular DFSs now follow.

Master 128

This Acorn DFS has a software double stepping mode for a 80 track drive. Set it with the command

*DRIVE 0 40 <RETURN>
and then hit <BREAK>
Disk User will then work without any need for conversion. However this may not allow writing to the disk in 40 track mode; in any case, you should make a working copy, so copy to a 80 track disk.

DFS on Master Compact

The DFS is supplied as an image on some versions of the Master Compact Welcome disk (or is available from Acorn on disk) and this may be used in conjunction with a 5¼ inch 40 track disk drive to run Disk User. Please note that we **cannot** at present supply *Disk User* on a 3½ inch disk (if there is ufficient demand, we may be able to in the future).

Opus DDOS/Challenger 3

If you are using the Opus DDOS disk filing system or Challenger 1.0/DDOS then issue the command
*4080 AUTO <RETURN>
or
*ENABLE 40/80 <RETURN>
and Disk User will work without any need for conversion.

Challenger 3

If you have the later ROM version Challenger 1.1 then issue the command
*OPT 8,1 <RETURN>
to achieve the same result. Disk User will work effectively from the RAM disk. Use
*COPY 0 4 *.*

*CONFIG 4=0
*OPT 4 3
to run from RAM disk

Solidisk DFS

With the Solidisk DFS 2.1 and 2.0 you can set a software double stepping mode for a 80 track drive with the command
*ENABLE 80 <RETURN>
Disk User will then work without any need for conversion.

Watford DFS

The Watford DFSs also have a software double stepping mode for an 80 track drive. Consult your manual for the appropriate FX call or command. Disk User will then work without any need for conversion.

Disk failure

If for any reason your copy of Disk User will not work on your system then please carefully re-read the instructions given above.
If you still experience problems then:
1. If you are a subscriber, return it to:
Disk User, Infonet, Times House, 179 The Marlowes, Hemel Hempstead;
2. If you bought it from a newsagents, return it to **Disk User Replacements, Discopy Labs, 20 Osyth Close, Brack Mills, Northampton NN4 0DY** before 1st January 1988. You can make telephone enquiries about *Disk User* on 01 437 0626 (please ask for *Disk User* Editorial). Enquiries in writing will be dealt with as promptly as possible and replacement disks sent out immediately. Please use appropriate packaging, cardboard stiffener at least, when returning a disk. Do not send back your copy of the magazine. Only the disk please.

Disk Hints

You can speed up your disk access, speeds by issuing the software command: ***FX255,15** <RETURN>
The effect lasts until <CTRL><BREAK> is pressed. This extra speed is only possible with the latest disk drives - the half-height and slimline drive types.
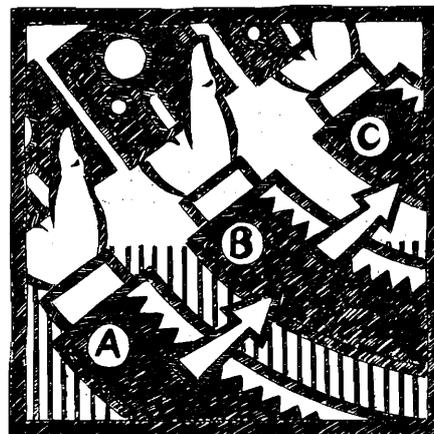
Disk User files

All change — 40 track to 80 track convertor
files:
CHANGE machine code file. *RUN

Disk User — Disk magazine title page animation
author: Abbas
files:
P.RUNDISC BASIC file

A.DISC machine code file
files:
MENU BASIC file
Autosave - Automatic program saver.
author: Andrew M. Page
files:
AUTOSAV BASIC file

Cholo Game File Editor - Customise the robots etc. in Firebird's Cholo.
author: Philip Colmer
files:
DISCLST BASIC file
DISC data file

Kwik Kopy - Fast file copying routine.
author: Andrew B. Heptonstall
files:
QUICK BASIC file
NOTE: This program fails to work properly on the Master Series and with the OPUS Challenger!

Dualcatalogue - Have 60 files each disc side.
author: Micheal Spalter
files:
DUALCAT BASIC file

Tazman - Arcade style shoot 'em up.
author: Rob Anderson
files:
TAZMAN BASIC file
O.OBJECT machine code file
O.CODE2 machine code file
O.CODE3 machine code file
O.CODE4 machine code file

Count With Teddy - Educational counting game for young children.
author: Richard Oldman
files:

TEDDY BASIC file
TEDDY2 BASIC file

Break Message - Utility to display message on BREAK.
author: Micheal Spalter
files:
BRKMESS BASIC file

Line Compacter - BASIC program line compacter.
author: Wouter Kolkman
files:
LINCOMA BASIC file - assembler source code

Program Shrinker - BASIC program shrinker.
author: Wouter Kolkman
files:
SHRNKaA BASIC file - assembler source code

SHRNKaB BASIC file - assembler source code
Calculator - Pop up calculator with full scientific functions.
author: F. Munkert
files:
CALC1 BASIC file
CALC screen graphics data file

Animation - Letter D is visited by Abbas.
author: Abbas
files:
P.RUND BASIC file
P.ALPHA machine code file

Note that Disk User almost fills a 40 track disk. Any software that may need extra disk space to save information must be copied onto a blank disk. e.g. Dualcatalogue, Line Compacter, Program Shrinker and Autosave.

DISCUSSION
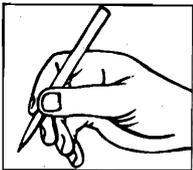
Dear Sir
I bought from my newsagent the first issue of Disk User, read it with interest and will be looking out for the next issue.

There are several magazines devoted to the BBC Micro, all of which I have seen at various times. In all of them I have found items of interest to me but in all of them I soon find that I get lost in the maze of technicalities which liberally sprinkle amost any article. Therefore I would offer a most fervent plea to you to remember those people using BBC Micros to whom references to PAGE, &E00, OSHWM and similar esoteric incantations might well be extracted from 'The Wizard's Book of Advanced Spells' for all they mean to us.

We (and I am assuming that I am not alone) belong to the group of people who only want to use an item of equipment, not understand how and why it works. We are the people who talk by telephone to somebody in Los Angeles without being interested in whether our words are transmitted by satellite or undersea cable. We certainly have no idea how the telephone exchange works nor do we have any desire to know.

We use our computers in a similar way. We only want to use the machine to run a piece of software, not delve into the way in which it works or how to make it work differently. The many people who do understand the various spells one can cast on a computer to make it do all sorts of wonderful things have my admiration. But I have neither the desire nor the time to join them. I would no more think of writing my own programme than I would of building my own telephone. I do not even understand the language which comes with Wordwise Plus.

Please remember me and those like me when choosing articles and programmes for inclusion in Disk User. By all means include spells for those inclined to use them and good luck to them. I would be lost without such people who understand it all, to write the software and make the computer in the first place. But please think sometimes of the dunderheads who have no idea of what makes it all work and are happy in their ignorance.

As to items I would like to see in future issues of Disk User, the following list very probably contains things which are already available but which I cannot understand or have failed to recognise. Be that as it may, here are a few things which I would like to see:
1. Some means of storing more than 31 files on one side of a disk from within a word processor without having to fit a new DFS. (I am on dangerous ground here, using one of the magic incantations, DFS. I hope I have used it in context).
2. A simple means of printing just what I see on my screen at any time.
3. A HELP service for dunderheads where I might find out what stupid thing I have done which prevents something from doing what it should. It could also provide a service to translate into understandable English some of the worst excesses of certain instruction manuals which appear to have been translated from Sanskrit by somebody who speaks only latin.

My thoughts about this first issue of Disk User are extremely favourable. For once I could understand most of what I read in a computer magazine (I did get lost in the second two columns of the article on the Disk Examiner utility but as that only explains how the thing works, I think, it did not really matter).
Yours sincerely

H.F.C. Wheeler, Herts *

Thank you for your flattering comments on Disk User — we like it too! We do try to make the magazine understandable to as wide a range of readers as possible. There are obviously items which have to assume a certain level of knowledge in a reader, or they would take up far more space in the magazine than we could possibly justify. However, we think you will find a number of items at a reasonably accessible level in this issue.

As regards your more specific requests, we will be looking at the possibility of doing just those sorts of item in future issues of Disk User.
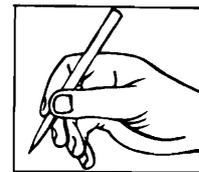
Dear Sir
Firstly, well done on an excellent publication. I am sure that it will get even better. Full marks.

Now to my point. In the second issue (August/September '87) you stated, on page 5, that the Selective Copy utility does not work on the Opus Challenger. Whilst true in its original form there are just three changes that need to be made to the 'SELCOPY' BASIC program to make it work. I have made these changes, below, and it works perfectly on my Opus Challenger.

Let's assume that we want to copy from Drive 0 to Drive 2. As it stands the program would send the comand *COPY 02 〈filename〉. This is not correct for the Challenger, there must be a space in between the source drive and the destination drive else Challenger will give Bad Drive error. To effect this just change line 190

DISCUSSION

from STX drv+2 to STX drv+3. But in doing that the command sent will now be *COPY 0 2 〈filename〉, so the DFS would look for a file with a Z as the first character, (as there is no space in between the destination drive and the filename). To change this we need to change the STRING$(11, in line 1500 to STRING$(13, to allow for the length of the " 0 2 filename". To copy files on Directory other than $ just alter the default by typing *DIR 〈letter〉 before using the utility.

That is all that needs to be done to make the program work, but I have altered a couple of other lines as enhancements. E.g. The Challenger system has a RAM disk but the program won't let either source drive or destination drive be out of the range 0-3, wheras the RAM disk is drive 4. To allow drive 4 to be selected just change line 1430 from CPX #52 to CPX #53.

I have also added the following lines to the very start of the assembly routine. They perform a *OPT 7 3 to set the disk stepping speed to minimum, so speeding up disk access time. The lines ato add are:

```
152 LDA #139
154 LDX #7
156 LDY #3
158 JSR osbyte
```

Just RUN the BASIC program and the machine code will be saved automatically. Just type *scopy and away you go.

I hope I have been of some assistance to fellow 'Challengers'.

Finally, I would like to issue a plea for some compatibility with Challenger Double Density Format in your utilities, such as the excellent automatic disc cataloguer in issue 1 which will cataloge any single density discs I have but will not catalogue double density discs — which comprise 99% of my collection! Even hints or tips on how to convert programs would be of help.

Yours faithfully
M.J. Holmes, Northumberland.
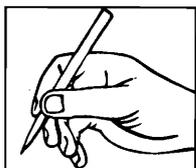
> Dear Sir
> You might like to know that this useful programme from issue two of your new magazine will run on Challenger.

Two small alterations are needed:
190 STX drv (Challenger will not accept the syntax 'COPY 02 Fred', which Acorn DFS will, but needs a space between drive numbers — 'COPY0 2 Fred' is accepted)
1430 CPX #54 (Allows use of drives 0 to 5 (ASCII value of max drive no + 1)).

With best wishes for the success of an interesting publication.

Yours faithfully
J.W. Yates, Surrey.

It just goes to show that there's more than one way to skin a rabbit, if one can say that sort of thing nowadays! We look forward to hearing from a reader who's come up with a single line alteration to get this program to run on the Challenger... no, that isn't a challenge we're throwing down.
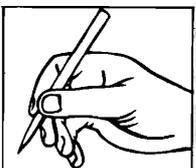
> Dear Sirs
> I bought a copy of your first issue a few weeks ago in Aberdeen.
> I have a few questions about a couple of the programs. Firstly though, I use a

Master 128 with twin 80 track disc drives.
1) In "Zoth" nothing happens when I get into any of the three chambers. How do I get the screen to "come to life"? My copy doesn't understand LOOK even though it is mentioned on the HELP page.
2) In "Barmy Butterflies" the screen fills with nonsence when I try the game play option, no matter what I've tried before that. Am I doing anything wrong?

Forgoing the above I really like the idea of Disk User and I am looking forward to further issues.

Yours Sincerely,
John J. Gilroy, Morayshire. *

If any other readers are experiencing difficulties with getting any part of Disk User to work, we would ask them first to check that there's nothing on their system which might conflict with the proper working of any of the software. For example, though it's difficult to generalise on this, but we have heard from a couple of readers who have found that spell-checker ROMs can cause problems, at least on their machines; however, we would add that in these cases, Disk User failed rather more dramatically! we will be following up all these sorts of instances we know of, and at the moment it's too early to say exactly what the fault was.

Once you've assured yourself that it's not your system that's causing the problem, and that you have read all the instructions, it's time to send the disk back to us, or rather to our agents — see Disk Instructions for details of where to send it (not that the address is different for subscribers and for those who buy their copy via a newsagent. You will get your disk replaced as promptly as we can.

Note: * indicated letters that have been shortened.

Imogen Hints

Below are some quick solutions but, first, how about these 'illegal' passwords discovered by Stuart Painting: NORMAL-MODE, REVIEW-MODE, TEST-MODE, DEBUG-MODE, GIMME, DUMP, EPILOGUE.

For the moment I'll leave you to discover what you can do with them (for instance, DEBUG plus the letter of the part will get you all objects, DUMP naturally dumps screen to printer, TEST-MODE then GIMME gets all spells, etc), but isn't it amazing that a supposedly debugged game should still have the programmer's own testing cheats in it?

Interestingly, Jahinder Sidhu from Birmingham reckons the minimum number of changes required to complete the game is 107, the maximum allowed being 150. Can you beat his figures. Section A, 9 changes; B, 7; C, 4; D, 8; E, 7; F, 8; G, 8; H, 5; I, 7; J, 3; K, 5; L, 5; M, 5; N, 5; O, 10; P, 10; and then one final change.

Our solutions come courtesy of Iestyn Day, David Booth and J H Drew.

Heroes all!

Section A: get bow, shoot balloon, table falls, jump over wall, shoot creature

Section B: fill bottle with water, water plant, climb plant, fill bottle with poison, tip poison into dog's bowl.

Section C: jump ono ball to push it onto see-saw, throw small ball at ball on pillar, collect sword, cut rope.

Section D: collect gun, shoot creature, shoot sign 'Gnu Climbing' off screen, climb rope, get axe, go right, shoot sign again, climb, chop down door.

Section E: whip dog, climb rope, get

Section F: get saxophone, stand on trap door, deafen creature to kill it. frog, make frog croak at 'Speak here' sign.

Section G: get bowl, spill TNT, throw bowl at TNT and pick up, get splint, explode door.

Section H: get banana, entice gorilla onto rope that holds up portcullis.

Section I: get small egg, throw at big egg, climb over duck, jump into duck, become duck and fly.

Section J: collect clanger, push bell into room with dog, hit bell, dog dribbles into umbrella which opens door.

Section K: use magnet to get weight from seal, throw weight onto rope, jump and repeat.

Section L: throw nut at first monkey, get another nut quickly, climb rope, throw nut at door.

Section M: get apple, place under rope and climb.

Section N: collect tulip bulb, crush hamster under log, plant bulb in dead hamster, climb tulip.

Section O: get torch, burn table, burn rope with parrot on.

Section P: get egg-timer, climb rope by weight, use pendulum to jump pit, collect bird, make pendulum swing, throw bird at broken clock, climb over boulder.

Of course, this has to be presented as shorthand solutions and if this is all too hard for you then just use this short bit of deviousness from 'the Hackers of Warwick School' - load in a saved file and change the two numbers at the start of the dump. These represent your power numbers to change these numbers (backwards) - just change these numbers to your own choice.

Or you could use this short cheat from Michael Mooney of Bolton for unlimited transformations on the disc version:

```
'LOAD "G" /
?&2DCA=&AD /
CALL&3C06
```

Naturally Mark Gidley also has a disc cheat for infinite lives - first SAVE the game to disc, then *LOAD l.filename Enter

```
SAVE l.filename
3000+&5 0 9EA.
```

Then

```
0%=?&3005:&$3
002="000":?&830
05=0%.
```

Enter 3000.

Passwords first:
A - BALLOONACY B - DRIPPING-STUFF, C - SEE-SWORD, D - GNU-PROBLEM, E - WHIP-IT, F - SAXOPHOBIA, G - DOWN-AND-OUT, H - FOLLOW-ME, I - DUCK-EGG-JAM, J - PAVLOV-WAS-HERE, K - ...BLUES, L - TENDER-HOOKS, M - BABBOONACY, N - HAMSTER-APPLESOURCE, O - FIRE-WORKS, P - TIME-FLIES.

COME HOME TO A REAL LIVE WIRE

Inject the power of a mainframe into your micro. MICRONET is the only product that provides you with:-

- Real-time interactive chatlines
- Free and discounted telesoftware
- M.U.G.'s inc. the UK's most popular SHADESTM
- Daily computer news and reviews
- Free PRESTELTM electronic mail
- Teleshopping — holidays, hotels etc.
- 70,000 PRESTEL users 56,000 Telecom Gold users
- National and International Telex
- Technical Support and Help Desk

PLUS: Access to PRESTEL and Telecom Gold, and there's much more — all for one low cost subscription.

Take a look for yourself: Flip the disc over and press shift/break to run the demonstration taken directly from Micronet. If you have a modem the disc has details of a free demonstration database. If you don't yet have a modem then pay your year's membership in advance (£66.00) and we will give you one FREE!

ARCHIMEDES

Archimedes - question of compatibility

Apologies for not including this piece in the last issue of Disk User. It was forced out by our full disk documentation. If you haven't already heard about Archimedes we've given a brief account of the launch and computer specification, followed by details on the ADFS filing system and how you might go about converting to the new, and exciting, RISC machine.

Date: June 16th
Time: 10.30 am
Place: Piccadilly, London

The launch of Archimedes, the Acorn RISC machine (ARM) brought oohs and aahs from assembled journalists as Acorn's Steve White, radio microphone in hand, demonstrated the graphics and sound capabilities of a sensational new computer.

Herman Hauser, ex Acorn, now head of Olivetti research, had already compared the advent of RISC with the advent of the jet engine, a simpler mechanism than propeller (conventional processors) blasting data out the other end at vastly increased speeds. The demonstrations made his remarks seem entirely appropriate.

The videos, live demonstrations and expert verdicts however hardly touched upon the vital facts about software compatibility. After all the Archimedes 305 and 310 are "the next BBC Micros". The Master Series remains but the Archimedes 305 (512K) and 310 (1Mb) BBC Micros still have the red keys to prove their pedigree. From the point of view of disk users, Archimedes presents remarkable, though by no means total, compatibility with what has gone before.

Firstly the disk filing system is ADFS, an enhanced version but with no problems reading old ADFS disks, if they are on 3.5" format that is. Acorn have introduced a new ADFS standard (800K) to Archimedes ADFS but that is restricted to the new machine. A second 3.5" drive can be introduced by simply plugging in a cable. Alternatively the 300 series can carry a Winchester hard disk controller on a Podule (as Acorn have christened the Archimedes peripheral interface modules) and 20Mb hard disk. A backplane has to be fitted before podules can be accomodated.

For the majority of DFS users there are therefore two hurdles to Archimedes compatibility:
1. move the software onto ADFS
2. swap 5.25" for 3.5" media

Solidisk Technology have already introduced a 5.25" disk drive with in-build power supply for the Archimedes. This acts as drive 1 under the ADFS and draws a small amount of power from the machine. There are no modifications involved at the Archimedes end although the drive needs minor 'tweaking'.

Software compatibility is potentially good with Arthur (the BBC lookalike operating system) ADFS and BASIC V now exploiting the new 32 bit RISC processor and its family of Acorn designed memory, video and I/O chips. Of course Archimedes comes with a window management system and desk top applications in ROM (total 512K) but the fancy 'front end' does not detract from these lynchpins of compatibility. The Archimedes BBC Micros 305 and 310 are available from September with the Acorn badged 440 available from November.

Superb seven position eight channel sound, 256 (from 4096) colours on screen at high resolution and the badge 'fastest microcomputer in the world' are tempting features. More so when your home-brew BASIC will need few adjustments and many 'legal' applications such as the View family will run on the 6502 emulation while you assess the new software.

AND DISKS

Archimedes - full specification

ARM CPU at 4/8 Mhz 0.5 to 4Mb RAM depending on model
Separate CMOS battery backed RAM 240 bytes, plus 16 for battery backed real time clock

512K ROM containing machine operating system, BBC BASIC V, ADFS, ANFS, BASIC editor, desk top manager, character sets (ISO 8859) Latin, Greek - Arabic version under development
Sound - 2 channel stereo with 7 stereo positions and 8 voices. one internal loudspeaker, 3.5mm stereo jack plug
Disk drives - 1Mbyte unformatted 3.5"
Serial interface - RS423, 75 to 19200 baud software selectable, 9 pin D-type plug
Parallel 8 bit Centronics interface, 25 pin D-type socket
Keyboard - 103 enhanced PC style. A 3 button mouse plugs into the keyboard viaa 9 pin circular connector. Adjustable function key card holder
Expansion (300 series) - 64 way DIN 41612 connector on main PCB for 2 socket backplane card (for Podules)

Econet - standard Master 128 module (although Acorn are working on a new version)
Power input - 198 to 264V AC (50Hz)
Software on Welcome disk - Painting Program, Font Designer, Music Editor, tutorials, utilities, 6502 emulator, floating point emulator, range of fonts
Documentation included - Welcome guide (for newcomers), User Guide, mainly on BASIC V and Arthur

Archimedes Display

18 display modes
European standard rate (non interlaced)
text graphics resolution number of colours
20x32 160x256 4,16,256
40x32 320x256 2,4,16,256
80x32 640x256 2,4,16,256
132x32 text only 16
40x25 text only 2
40x25 Teletext 16
80x25 text only 2,4,16
132x25 text only 16
High scan rate (for use with monitors such as NEC multisync) 3 screen modes text graphics resolution number of colours
80x64 640x512 2,4,16

Archimedes - prices (including VAT)

Archimedes 305 £918.85
with mono monitor £976.35
with colour monitor £1148.85
Archimedes 310 £1006.25
with mono monitor £1063.75
with colour monitor £1236.25
The 400 series starts at £1608.85 and pushes up to £2873.85 for the top of the range 4Mb RAM with colour monitor and 20Mb hard disk.
Econet module £49.99
0.5Mb RAM upgrade £102.35
Floppy disk (3.5") £143.75
20Mb hard disk and Podule £573.85
Podule backplane £40.25
A Series (confusing isn't it, all these different titles) manual £19.95 ROM Podule and I/O Podule prices to be announced. These are designed to bring Archimedes up to the BBC Micro specification!!! A MIDI interface to add on to the I/O module will be available, price unknown MS-DOS 10Mhz 80186 co-processor unit, availability/price unknown Third party podule releases are known to include modem, IEEE, video frame grabber (Watford Electronics) and high performance A to D interface.

Technicalities

Problems in porting DFS based software to Archimedes ADFS may be encountered for a number of reasons, most of which software writers have already encountered with the Master Series:

● assumptions have been made about DFS workspace (directly reading memory locations which no longer hold the required information)

● assuming DFS format when using OSGBPB (&FFD1), eg expecting filenames to be at intervals of 8 bytes when reading the names in the current directory

● encountering the hierachical structure of the ADFS eg unable to jump directly between directories with *DIR. ADFS requires the full pathname to be specified

● OSWORD calls &7D, &7E and &7F are not supported by the ADFS. Nor does the ADFS any longer support &70, &71, &72, &73 for double density formats.

Andrew M Page

AUTO SAVE

It doesn't take long to find out that a system
crash can lose a lot of programming work.
This utility will save much of the grief.

This routine is designed to save a copy of any BASIC program under development every few minutes. The routine uses the interval timer to interrupt BASIC when the set time duration has elapsed; the BASIC program is saved onto disc under a pre-designated filename.

The routine is compatable with both versions of BASIC and is, obviously, designed for disc-based machines.

Setting The Timer

To set the timer, run the source program AUTOS. You will be asked to enter the time delay in minutes. The routine will then be saved to disc under the filename AUTO (though this can be any other seven letter filename). Time delays of less than one minute will be rejected.

Using the routine

Before using the routine it is important to load the program that you are working on into the machine. Having done this the routine can be enabled by typing:
*AUTO PROGRAM
This loads and executes the routine, using PROGRAM as the filename for your BASIC program. You will hear a reassuring beep as your program is being saved.

The filename can be made to change every time the event occurs by simply adding a star (*) at the end of the filename. For example, typing:
AUTO PROG
would save the program as PROGA the first time it was backed-up, then PROGB the next time, then PROGC, and so on.

.If at any time you wish to disable the routine simply type:
*AUTO <return>.

If the program that you are working on uses files, you are advised to disable autosave before running the program, as the machine could 'hang' due to the interrupt routine being stalled.

The machine code is assembled at &900; if this location poses a problem as it could with certain ROMS fitted (DISC DOCTOR uses some locations in page 10), the location can be altered by changing LOC on line 110 to the required address.

If you decide to assemble the machine code at PAGE (&1900), **do not forget** to execute **PAGE=PAGE+512** before loading your own program to run with the routine.

The routine uses four zero page locations which are: &70, used as a store; &71, used as a store for incrementing filenames; &72, used to store the position of the last character in the filename; and &73, which is used to store the low byte of TOP for later comparison.

How The Routine Works

The routine uses the interval timer to interrupt BASIC, and save the BASIC program at the pre-designated time interval. When this happens the interval timer is reset to the previously entered time delay.

If there is no program in memory, or its length has not changed since the last event then the routine returns to BASIC without saving any code. However if the program length has changed between events then the code between PAGE and TOP will be SAVEd to disc.

The values of PAGE and TOP are stored by both BASIC1 and BASIC2 in the first page of memory. The high byte of PAGE is stored in location &18. The low byte of TOP is stored in location &12 and the high byte in location &13. The routine uses these values to save the program to disk via the operating system command line interpreter (OSCLI).

The program is well REMed so those readers who are interested in finding out how the routine works should have little difficulty following the listing in the source file AUTOS.

General Description Of Program Listing

Line numbers given in **bold** followed by action in normal type.
100-180 initialise the program

constants
250 disables events if they are already enabled,
340-380 check for a filename, if no filename is found then interrupts are disabled. However if a filename is found it is read in and stored in a buffer. If it contains a star then an 'A' is stored in location &71, all following interrupts will save the BASIC program under a different filename, ie starting with an 'A' as the right most character in the first filename, then a 'B' in the second, and so on
410 enables interrupts (*FX14,5)
440-460 output a message to the screen and wait for the space bar to be pressed
490 after interrupts have been enabled this is the entry point whenever an interrupt is generated
510 raises the bell pitch
530 resets the interval timer to the original time delay
550-560 read the high byte value of PAGE and TOP and call a subroutine which converts these numbers into ASCII form
610 reads the low byte of TOP. If it is found to have not changed since the last interrupt then the BASIC program in memory is not saved, as it is pointless duplicating the same file
620 stores the new value of the low byte of TOP for comparison at the next interrupt.
650 if the filename is to increment at every interrupt then the letter which is to become the right most character of the filename is read in from location &71 and stored in the file buffer. The contents of &71 are then incremented
660 saves the BASIC program via OSCLI
680 sounds a high pitched bell
690-700 in these lines the bell pitch is reset, the registers are pulled from the stack and the routine is exited.

Subroutines

730-780 are routines that basically store the values of PAGE and TOP in the file buffer as ASCII characters
790 is a routine to disable events
820-890 contain the text and storage buffer used by the routine.

DISK NEWS

Write for Disk User

Disk user is constantly on the lookout for original and well written articles and software for publication.

We are especially interested in disk utilities and interactive tutorial style programs. Graphics and music specialists may also appreciate the ability of Disk User to deliver their programs to a large and appreciative audience.

All submitted material must be in machine readable form. This applies both to programs (in any language) and to documentation, which should be prepared with a BBC Micro wordprocessor. 5¼" disk (DFS 40/80,ADFS large/small) equally acceptable. Please also include hardcopy and any suitable illustration, photographs and/or screen dumps.

It's sensible to give us a ring before going ahead with any major work.

All submissions will be acknowledged and material returned if not required. On acceptance the copyright in such works which will pass to Argus Specialist Publications Limited will be paid for at competitive rates. All work for consideration should be sent to the editor at the above adress.

Opening Doors

We've got news of software for those who find the door closed on them by different filing systems and incompatible disk media. There's nothing worse than knowing the data is on the disk but you can't get at it.

Baksoft supply a program for £19.50 (discounts for more than one off) which allow the BBC Micro to read and write a wide range of disks, including IBM and IBM compatibles such as Amstrad 1512 and CP/M machines. You must have a BBC B+, Master 128 or Compact or a Model B with the 1770 disk interface. The programs run on the BBC, and depending on the size of your disk drives, you can read and write many formats including: MS-DOS/PC-DOS, Torch CPN, BBC DFS and ADFS. Apple, Sirius IBM high density, Amiga and hard sectored disks cannot be read by the BBC.

The programs are menu-driven and allow you to COPY, TYPE, HEX dump, ERASE, RENAME files, to get a directory and (in some cases) to FORMAT disks. Access is also allowed to BBC * commands.

Pop Ups with Sideset

Sideset is a disk based switching system for the BBC. The BBC runs a single program at any one time and its operating system is unable to run more than one as is now possible with larger memory computers. However, as utilities such as REPLAY and GENIE have demonstrated, it is possible to stop one program, use another and then return to the first at precisely the point you left it. In practice this means saving the complete status of the computer ie the whole memory.

Sideset from Maze Technology actually offers the ability to freeze a number of programs in their 'current' state while running one program in a screen window. Programs can be suspended, reactivated or closed down. The windows which contain the programs are user defineable and are neatly nested on screen so that a number of frozen programs can be viewed at once.

Sideset is a single ROM chip which performs all the management of task switching. Sideset manages the freezing of the program, the saving and retrieving of the memory via the disk surface and the screen windowing.

Sideset comes with its own set of 'pop up' style utilities, calculator, notepad, address book and 1000 year diary.

Compatible with all BBC Micros, DFS and ADFS, Sideset is available at a launch price of 39.95 plus VAT. Details from Maze Technology, 11 Braemar Avenue, Neasden, London NW10.

New Verbatim Floppy System

Floppy disk storage has dominated data storage for microcomputers despite attempts to introduce fast tape systems and the downward spiralling costs of hard disks. Some new technology may boost the lifespan of floppy storage into the 90's.

Verbatim have introduced a new family of floppy-disk based subsystems. They are only available for IBM PCs at the moment but the add-ons - complete systems with controller, cabling and software - may find success on other microcomputers. The software makes "unlimited" storage available to most applications, languages and disk based operating systems.

The servo-embedded technology enables the drives to constantly - 102 times per second - monitor and realign head position, ensuring accurate data storage and retrieval. The systems break the current single floppy storage limit of 1.6 Mbytes. The half height drives are seen as competitors to hard disk drives, offering portability of media and ease of backup, the two main problems with hard disk media.

DISK NEWS

Three time winners

A disk version of the Ket Trilogy from Incentive Software is now available. Released at the beginning of June, the disk retails at £11.95. The adventure comprises three separate parts, Mountains of Ket, Temple of Vran and the The Final Mission. Each is a complete adventure in itself, and can be played independently of the other two. Details from Incentive, 2 Minerva House, Calleva Park, Aldermston, Berkshire, RG7 4QW. Telephone 07356 77288.

Adventure lovers can also invest in the £25.95 disk version of the Robico Rick Hanson Trilogy. The disk version costs only two pounds more than the cassette bundle. Details on 0443 227354.

Watford DDFS Mark 2

Watford Electronics has redesigned its DDFS board for the BBC Micro to make it faster, slimmer and Acorn ADFS ROM compatible. The original DDFS set new standards in disk filing system design, bringing double the storage to DFS disks. The completely redesigned version, the Mark II, is compatible with files created by its predecessor.

An official Acorn Tube host is built-in, allowing a second processor or co-pro adaptor to be used without the need to have an Acorn DNFS ROM installed. The automatic 40/80 track sensing almost does away with the need for switchable disk drives.

Extra speed can be generated through the option to switch off error check on BPUT and BGET operations. OSGBPB has been recoded, to make it even faster.

Reduced size has been made possible through state of the art technology. It fits with all third party ROM boards.

Watford claim that the Mark II has incredibly powerful 8271 emulation ensuring general software compatibility. The DDFS is capable of providing either the full 80 percent storage increase or a file the full size of the disk.

DDFS Mark II costs £39. An upgrading deal for DDFSs of any manufacture is on offer at £30, provided the original DDFS and ROM board are returned. Acorn's ADFS ROMs are £25. Details on 0923 37774.

BeebDOS

If you work with both IBM and BBC microcomputers and you use a double density filing system (ADFS, DDFS, DDOS) then a new product called BeebDOS could prove invaluable.

BeebDOS from Micro Boss Ltd, runs on IBM PCs and compatibles (including the Amstrad 1512, and allows the sharing of both text and graphics data between BBC and PC. Even though the disk is in an IBM drive, it can be handled as a BBC disk, with similar * commands available.

DISK NEWS

Design on Disk

AMS have launched their computer aided design package for the BBC. Aimed at schools and colleges, AMX Design incorporates the user-friendly mouse interface which characterises AMX product.

Students are able to produce drawings from the simplest PCB Design up to the most intricate architectural plans, while drawing tools such as ellipses, circles, arcs and boxes can be selected. Advanced features such as full zoom macro options and multi-level drawing planes are included as standard.

AMX Design will output to any Epson dot matrix, Penman or Linear Graphics Plotmate plotters. Price is £69.99 with discounts on bulk educational purchases. Details on 0925 413502.
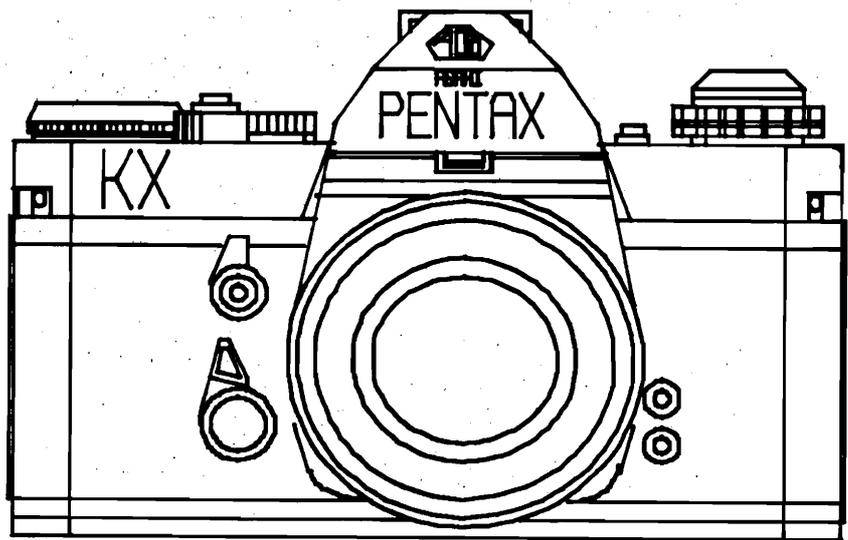
| DATE 19.6.87 | A DETACHED HOUSE | DRAWING NO. 1 |
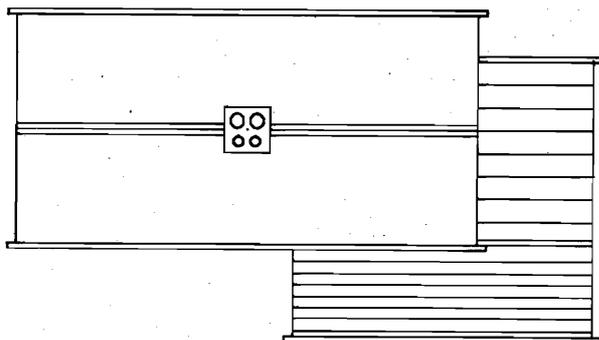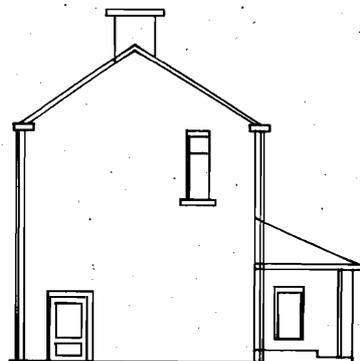
ORTHOGRAPHIC PROJECTION

DETACHED HOUSE.

DRAWN USING AMX DESIGN
PLOTTED USING PLOTMATE A3M
BY LINEAR GRAPHICS

DESIGN & TECHNOLOGY DEPT
WOOLWICH POLYTECHNIC SCH

R S Anderson

Tazman - try this tough arcade challenge. Send in your high scores to Disk User and we'll acknowledge your game-playing skills in our high score table in a forthcoming issue. Top score wins a dozen Disk User data disks. Best of luck!
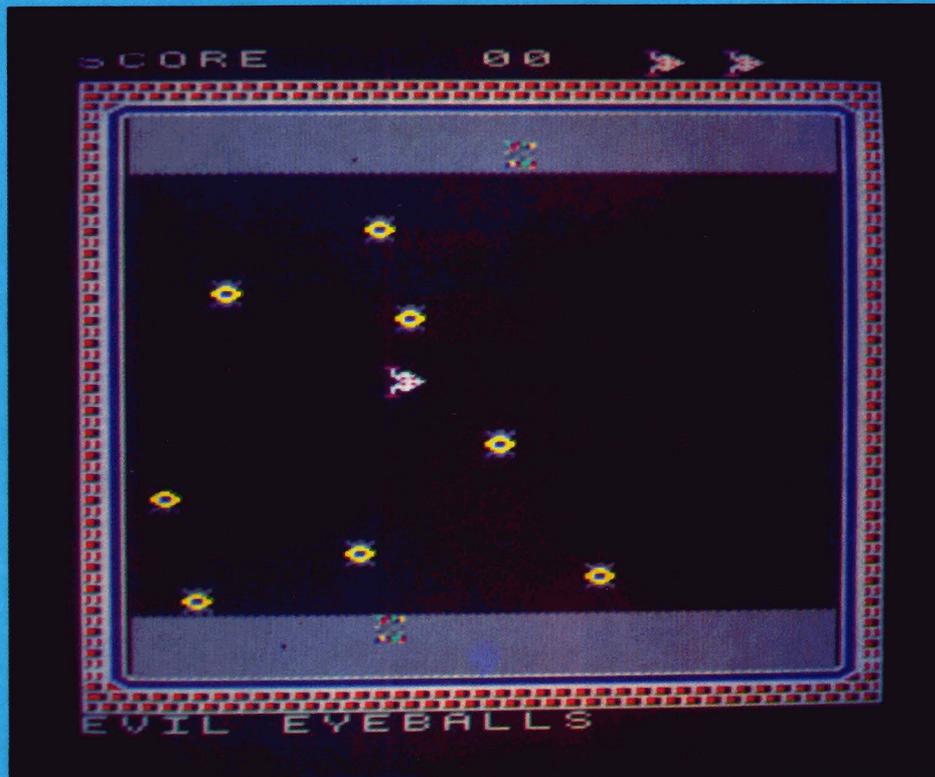
TAZMAN

```
INSTRUCTIONS

Keys:
            Z - left
            X - right
            : - up
            / - down
      RETURN - fire

Now press ESCAPE to play the
game, or press RETURN to see
the instructions again.
```

FLOPPY

Cholo has turned out to be one of the most ambitious, fascinating and complex of BBC games, mainly achieved by using disk storage. Find out more about the game with our editor on disk and read on for a solution...

Cholo solved!

Cholo is one of the most exciting games to be released for the BBC Micro so far. It offers 3D wire frame graphics and logical puzzles while including an element of shoot-em up games.

Since *Cholo* is mainly an adventure type game, the authors have included the facility to save and load positions from the game — a very useful by-product of the game's release on disk, which was essential given the amount of data that is loaded in throughout the game.

This month we're pleased to bring you a *Cholo* game editor by Philip Colmer, together with his hints, tips and thoughts about the game. If these aren't enough to satisfy you, then we also have a complete solution.

It's your choice to decide how much help you think you need!

The Editor

The program is written entirely in BASIC and should work with all generations of BBC Micro. It uses Mode 3 and should work with shadow mode enabled if memory is tight (such as on the Electron). However, it has only been tested on a BBC Model B and a BBC Model B+.

It works around a series of menus, where each option is selected by pressing the appropriate letter. Pressing ESCAPE at any time will return you to the main menu which is displayed on starting the editor:

A Load a Cholo game file
B Save a Cholo game file
C Edit a robot
D Quit

Load a Cholo game file

If "A" is pressed at the main menu, a prompt will appear for the name of the game file. The editor keeps track of the last file name used. This will be displayed inside square brackets. To use the default file name, just press RETURN on its own.

If the file cannot be found, a beep will be heard and the prompt will reappear, otherwise the game file will
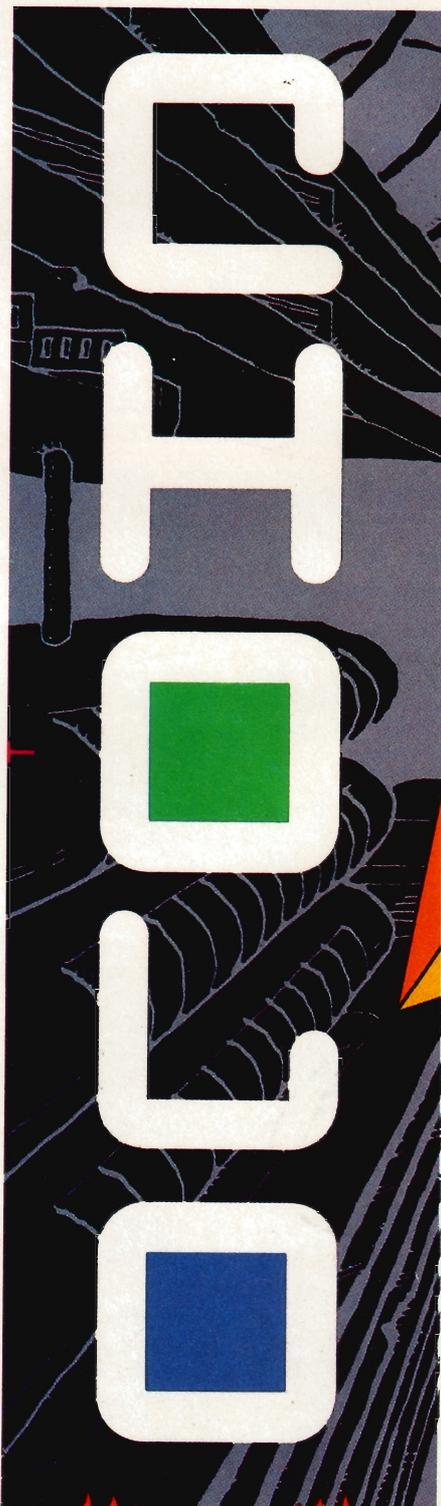
be loaded.

If the text entered begins with an asterisk (*), the program will pass it to the operating system for execution. Any output produced as a result will be displayed within a window, with page mode on. Pressing the space bar at the end of the command will return you to the prompt for the file name.

Save a Cholo game file

If "B" is pressed at the main menu, a prompt will appear for the name of the game file. The default file name will be displayed inside square brackets. After the file name has been entered, the game file will be saved.

As with loading a game file, operating system commands may be entered by prefixing them with an asterisk (*).

Edit a robot

This is the main section of the editor and is reached by pressing "C" at the main menu. A list of the available robots will be presented. The required robot can be selected by pressing the appropriate letter. Alternatively, you can return to the main menu by pressing "O".

Once a robot has been selected, two columns of information will be displayed. The left hand column displays the details of the selected robot while the right hand column displays the details of a reference robot. This is useful where you wish to make one robot similar to another robot or you wish to move a robot near to another robot. The left hand column will look something like this:

A Name (RIZZO)
B Type (RAT)
C Image (RAT)
D Direction (N)
E X location (6270)
F Y location (CF6C)
G Z location (0)
H Health (0)
I Controlled (Y)
J Password (REBELS)
K Modify RAM PAKs
L Change displayed robot
M Previous menu

FUN!

The values/text shown in brackets are the current values and may be changed by pressing the appropriate letter:

Name: this allows you to change the name of the robot. This can be up to six letters.

Type: this allows you to change the type of the robot.

Image: this changes the rotating image shown while in *Cholo* menu mode.

Direction: this changes the direction (N,S,E,W) that the robot faces.

X,Y locations: this changes the position in *Cholo* that the robot occupies.

Z location: this changes the vertical position of the robot.

Health: this changes the health of the robot (useful after a fight!).

Controlled: this allows you to take control of a robot without having to paralyse and enter the password.

Password: this allows you to enter a new six letter password, which *must* be six letters long.

Modify RAM Paks: this allows you to select which programs a robot is carrying.

Change displayed robot: this allows you to change the displayed robot on the right hand side. This value is remembered for the next time you edit a robot.

For options B, C, K and L, a new menu is presented to allow you to make your choice. For options A and J, a prompt is presented to allow you to enter the new text. Selecting option I will invert the controlled flag while selecting the other options (D, E, F, G, H), an appropriate value must be entered in hex.

There are times when changing the X/Y location of a robot will result in the robot dying. It is not apparent why this should be so.

You may quit by selecting "D" at the main menu which will return you to BASIC.

The game file

The editor does not allow access to all of the data held within the game file, mainly because I could not figure out what the rest did. As a guide, here is a map of the data (as offsets from &900):

000...041 robot type names. 6 chars each.
042...04C unknown 1.
04D...0BC general robot info. 8 bytes each: direction + image, &FF, X position (2 bytes), Y position (2 bytes), Z position (2 bytes)
0BD...110 robot names. 6 chars each.
111...164 passwords. 6 chars each. There is one entry per robot.
165...172 robot types. 1 byte each. There is one entry per robot.
173...18E unknown 2.
18F...19C health + controlled bit
19D...1D4 unknown 3.
1D5...1E2 possession of programs. 1 byte per robot.
1E3...1F5 unknown 4.
1F6...235 program names. 8 bytes each.
236...2B9 unknown 5.

There are 11 robot types. By default, these are RAT, LEADIE, COLA, HACKER, TANK, GUARD, DOCTOR, FLYEYE, CAMERA, PLANE and SHIP.

There are 14 robots. By default, these are RIZZO (REBELS), GORT (KLAATU), KOKE (LOREAN), IGOR (PLUGIN), RIDLEY (LEGEND), DRJOHN (STETHO), ANNEKA (BRAZIL), FELINI (ACTION), AVIATA (FLYBOY), QUEEN (FREDDY), SKUM01 (XXXXXX), SKUM02 (XXXXXX), SKUM03, (XXXXXX) and SKUM04 (XXXXXX). The robots' passwords are given in brackets. Note the guards' names (Philby, McLean, Blunt and Hollis) — a typical example of the programmers' humour.

The facility to edit the program names and the robot type names was omitted on the basis that it does not help much. It is not hard to add the facility. The block "unknown 1" is related to the robot type (since there are 11 bytes taken). Modifying the values seems to affect the physical height of the robot, but I was not sure.The block "unknown 2" is related to the robots, seeming to be two blocks of 14 bytes. It was not apparent what these values were for.

The block "unknown 3" is also related to the robots. It appears to be another set of X and Y positions. The first block of fourteen bytes is the LSB of the X coord (&19D ... &1AA), the second block is the MSB of the X coord (&1AB ... &1B8), the third block is the LSB of the Y coord (&1B9... &1C6) and the fourth block is the MSB of the Y coord (&1C7 ... &1D4).

The coordinates appear to be the starting position of the robot in the game, but manipulating the values does not appear to do anything useful.The block "unknown 4" is totally unknown, mainly because it is not 11 bytes long or 14 bytes long. Similar reasoning for "unknown 5".

The only feature which I know is held in the game file but which I could not track down is a group of bytes indicating which programs the Cyber computers hold.

Hints and tips

First, a bit of trivia. The tune played at the beginning of *Cholo* is held in the file "m.Jar". This file may be manipulated & played using Island Logic's *Music System*!

In addition to the Cyber computer in the main building, I have found three others:

Cyber 2: This is inside the church on Logic Island. It contains ReadmTXT and SurvyPRG. The former is a list of credits for *Cholo* while the latter gives control over two cameras (see below).

Cyber 5: This is inside the Turing Institute building. It contains Hack1PRG and gives control of the ship.

Cyber 3: This is inside a building in the top right hand area of the map (a solitary dot on the screen). Hack1PRG is required to gain access. It contains Hack2PRG.

The following list of initial locations for the robots may help:

RIZZO: Inside the main building.

GORT: Inside the building containing Cyber 3.

KOKE: Inside a building in the middle of an island in the bottom right hand area of the map. Use the positions of the two cameras to locate it (*RIDLEY* is looking right at it).

IGOR: Just outside the main building.

RIDLEY: Inside the same building as *KOKE,* facing it.

DRJOHN: On the main path to the Turing Institute.

ANNEKA: Round the right hand corner of the main building.

FELINI: South-east corner of the island *KOKE* is on.

AVIATA: At the north end of the run-way.

QUEEN: Slightly off-shore from the harbour.

SKUM01..04: Varies according to their personality. This may seem strange, but these four can change their personality according to your

FLOPPY FUN!

current location. For instance, down by the harbour, they turn into rats while by the Manic Mine, they turn into tanks. It should be realised that whatever their image, their password is always the same. Any *SKUM* robots under your control will shake off that control in any areas with other *SKUM* robots — there is no point trying to gain control of them.

I have found four transporters — two land-to-land and two land-to-ship. The two land-to-land transporters allow access to Logic Island. There is one on the island while the other one is on Group Pier.

The easiest way to put a robot such as the *HACKER* on the Island is to place it on the pad on the pier and land the *FLYEYE* on the pad on the island. The first land-to-ship transporter is by the harbour. The other pad is on the island where the *KOKE* robot is to be found.

To dock the ship, approach the short pier from the side of the pad. In other words, to dock in the harbour, approach from the west, while to dock at the island, approach from the north (the pad is on the east side of the island).

The two cameras, (*RIDLEY,FELINI*), can be obtained in two ways:
1) via a legitimate way. Take the *HACKER* to the Cyber computer inside the church on Logic Island. The program SurvyPRG held there will give instant control.
2) via a bug in *Cholo*. Interface the *RAT* to the *HACKER* and write an empty file. This will create the program SurvyPRG, thus giving control over the cameras.

Once you have the two cameras, you will realise that there is an island in the bottom right hand area of the map. One of the cameras (*FELINI*) is over the island, while the other (*RIDLEY*) is inside looking at the robot *KOKE*. Since *RIDLEY* is inside, you can now save the game position at any time.

One of the TXT programs available will inform you that this is a shock-sensitive bomb. By putting this robot on the ship and taking it back to the mainland, you can load it into the plane, take the plane up into the air, crash it into the pyramid (in the U shape at the top of the map) and thus

finish the game.

Several unknown factors, again, though:
1) How do you get control of the plane?
2) How do you get control of the bomb?
3) How do you get over the suspension bridge?
4) What is the doctor for? And how is it utilised?
5) Where are all the TXT programs held?
6) What does Hack2PRG do?
7) Where are the other Cyber computers? There should be at least Cyber 4.

Well, I hope that this gives some insight into *Cholo*. There is obviously a lot more to the game than first meets the eye. Even with the help of the game file editor, there is still a lot to solve. Perhaps other readers could add to the features of the game file editor and solve the other puzzles.

My thanks to Peter Atkin for getting me hooked on *Cholo*.

Solution

Various solutions to the game have been sent in and the following is an amalgam of the work of Mark Nuttall, Trent Gray-Donald, Bob Bembridge, R.York, Tony Bourne and Michael Knight. Thanks to them all!

Firstly, some general tips: do not save game positions on the *Cholo* disk — there are 'hidden' files on it and you can wreck it; save the game whenever you get another robot; the map is pre-war and is **not** accurate; take great care by water's edge; always transfer as many programs into robots under your control as often as you can; stay away from Manic Mine and Sladek Robot Factory until heavily armed; guard robots on bridge, at airfield and at Manic Mine are only ones that will shoot on sight, others wait until you fire.

More specific tips: some robots have cannons fixed at different heights; return IGOR to CYBER 1 when you gain access — there is more in memory than first appears; shoot FLYING EYE as it descends, not when on ground; it is safer to go round Church anticlockwise; check all 'dead' robots; take care going too far east or near edge of maps, bugs in game can cause it to lock up.

So, the solution! You start as RAT — exit building and take over HACKER, return to start building, interface with computer and read/store PasslTXT and RadarPRG. Return to RAT, get FLYEYE (South east of start building), take over and explore city — note aircraft at northern end of airfield, club house on golf course and radio-active island (location X=008, Y=010).

Fly to Logic Island and land on teleport pad, drive HACKER to Group Pier and teleport pad, robots will change places. Enter Church with HACKER and get SurveyPRG (run to get control of cameras FELINI and RIDLEY) and ReadmTXT. Running cameras will allow you to survey the unmapped island.

Reverse teleport instructions to return HACKER to mainland. You now need to cross bridge — there are two possible ways. Firstly, use RAT, cross at full speed, firing and dodging guards or, more elegantly take HACKER across and swop to another robot when damage indicator shows 75% damage. Explore generally with this other robot for a couple of minutes and return to HACKER — auto repair will have repaired damage. Repeat as necessary. AUTODOC is NE of bridge if needed.

Enter Club House and find computer — past it in the alcoves are dead robots. The last of these is GORT — take control. Go to Science Institute (two computers there) and get control of SHIP then locate it by shorter pier in western harbour (dock ship at 45%). Get GORT on board ship to uncharted island, dock and disembark, hack into computer and read programs AtmosTXT (gives location of entrance to underground bunker — just north of start position) and C-C-CTXT, which explains that KOKE COLA is an unstable bomb. Locate KOKE, get control, return to ship and pier (use the other one), then go to airfield.

Get control of PLANE (just bump into it) and take off. Fly to bunker cap (the area shown by Manic Mine on the pre-war map) — through the gates and crash into pyramid shaped building.

Congratulations — you have now completed *Cholo!* However, this solution is not necessarily the only one — parts of the game can be played in different orders and you'll need to experiment a little to work out when it is best to swop robots, save your game position and discard programs as no longer necessary.

Meanwhile, send your tips and pokes for disk-based games to me via **Micronet mbx 919999020** or to:
Dave Reeder,
Disc User,
ASP Ltd,
1 Golden Square,
London W1R 3AB.

ADVANCED UTILITIES

Advanced Control Panel

Nicely packaged this one. On folding open the box the neatly daisywheeled manual drops out to reveal the ACP EPROM sitting in the centre like a jewel in its setting!

The Advanced Control Panel (ACP) is a 'front end' designed to help the user control all aspects of the micro setup, not just those related to the disk filing system. On a Master 128 or Compact the configuration is held permanently in the CMOS RAM, on other models it has to be saved to disk and retrieved at switch on. If you use ACP in sideways RAM, the configurations are held in the last few bytes of the RAM thus surviving any clearance of main memory.

ACP is a language ROM and when placed in the highest priority position in your machine or configured as the priority language, it will take over at power up. ACP is fast in operation, simple to use through its menu structure, RETURN to move down or, at the lowest level, select, ESCAPE to move back up. The cursor keys move along the menu titles, revealing, rather than pulling-down, the options assigned to each default 'panel' LANGUAGE, MOS, FILE and PANEL. The latter option is the key to the user creating their own new panels. The system is ingenious and effectively realised.

The calculator - usually a feature of pop-up utility systems such as Genie or Sidekick - is a bit out of place in the control panel but very nicely implemented. Working in binary and hex as well as decimal, it could certainly be useful for programmers.

It is not possible to move from an application such as View or Wordwise, use the calculator (or any other part of the panel) and return without losing data so ACP does not compete with Genie or Sideset in this respect.

Of most interest to disk users is the FILE panel, which allows the user to select the current filing system, to select the default drive, to title a disk and to set the !BOOT option. Finally selecting DIRECTORY brings up a menu program, a display of files

which can be highlighted by moving a cursor. Information about the currently highlighted file is given at the bottom of the display. This is an all important indication of how the menu will treat a file if asked to act upon it. Menus of this type make use of the file address information to decide on a course of action, whether to CHAIN or LOAD in BASIC, EXEC an ASCII file or RUN machine code data. Five further actions can be taken manually, delete, dump, type, enter View and load file and enter Viewsheet and load file.

Do it yourself

So far we have dealt with the default ACP panel consisting of four windows. The power of ACP is that the user can create a personal panel, or panels. All you have to do is respond to the prompts for panel name, window names and choose from a list of possible window functions. These include directly dealing with a file, CHAINing, *RUNing etc, creating a panel of text - for help screen or information - or choosing one of the standard features, such as DIRECTORY or ROMS.

Naturally you can LOAD and SAVE the panels. When in memory (the first resides at PAGE), a list can be displayed from which a single key press will activate the highlighted panel name. Creating hierarchical menus with windows is as easy as falling of a computer stool. The menus are not dynamic but if you wish to create a foolproof 'turnkey' system for a standard application with known aims and standard files, there is no easier way.

ACP is much easier to use than the control panels supplied on either Master 128 or Compact and is the most stylish and comprehensive of utilities. Beyond the style there is the very useful ability to manufacture an environment and the potential of the ALARM facility for activating some function in the absence of the operator. I know someone who wakes up to his BBC pulling off the appropriate pages of CEEFAX travel information for his morning journey

with a special program he wrote himself. ALARM means that no-one should find any similar function difficult to implement.

The Model B or B+ owner does not get quite the same value from ACP because of missing configuration features and lack of real time clock. Even so the panel creation feature is worth looking at.

Advanced Disk Filing System Utilities

Pineapple Software's Advanced Disk Filing System Utilities (ADU) is a set of * commands for users of the Advanced Disk Filing System. Although DFS is the lowest common denominator of the Acorn filing systems, many disk users now make use of the ADFS for its higher storage per disk and the natural hierarchical structure which it can give to their stored programs and data.

One of the problems of ADFS in the past has been the disk based utilities, designed originally for hard disk and at least three keystrokes and a number of seconds disk accessing away from the user. ADU does the decent thing and replaces *FORMAT, *VERIFY, *VFORMAT (both), *BACKUP, *CATALL, *DIRALL and *DIRCOPY in ROM. There is a *DFSADFS command which takes over from Acorn's copyfiles. The command copies all files from a DFS disk without any discernment to a directory on an ADFS disk. It won't destroy ADFS directories and you are prompted if a locked file is encountered. These commands are not particularly faster than the Acorn equivalents but they are instantly available.

In addition we now have *DIRDESTROY, *DRIVE, which increases basic compatibility with DFS software, *PWRBRK to reset the system, *KILLADU to turn the utilities off and *ADU to turn them back on. Finally there is *MENU and *DISKEDIT.

Even within Acorn some would agree that the ADFS is clumsy and slow in a number of aspects. Often it is difficult to move around the directories in an easy manner.

A sudden, and welcome, rash of utilities. We examine their potential for disk users

Various menu style programs have been published to overcome this. ADU, as well as offering utilities, also offers a menu system (*MENU) which allows the relatively simple movement around directories and also manages your sideways ROM and RAM, loading and saving banks at your behest, inserting, unplugging, wiping and telling you what sort they are. Files are activated by highlighting them and a large range of actions can be automatically taken. The system recognises file types (including Interword, View family, Basic Editor, Edit and Wordwise) and a ROM (previously marked) can be entered and a file loaded, as well as more normal operations such as type, exec, chain and run triggered. These actions can also be overridden if required.

The menu doesn't present the most attractive display I have seen but all the information is there and you are not forced into any actions you don't wish to take, you can <ESCAPE> back to the main display from prompts for filenames and so on and <ESCAPE> then takes you out of the menu back into the current language. While in the menu the other utilities are available through various key presses. You can change fore and background colours, change drives and compact files.

The disk editor (*DISKEDIT) takes you into a display of the data held on the disk - a similar display on DFS can be viewed via the Expert Investigator on Disk User this month. The ADU editor allows you to directly change data on the disk (in hex or ASCII) and intelligently saves your version, making sure that the ADFS will not reject the changes on error checking. A neat feature when editing from the menu is that you are presented with the first disk sector of the object file highlighted by the cursor.

ADU comes with a neatly printed 28 page manual with an introduction for beginners to filing systems and a full specification of each command.

Old timers

Having looked at the new boys we should briefly reconsider Advanced Disk Toolkit (ADT) from ACP which really comes into its own when combining use of both DFS and ADFS.

Those utilities missing from the ADFS ROM are all found in ADT. It is very nice to have them back on immediate tap. ADT's DUMP is far superior to Acorn's original, allowing output in a variety of formats, even disassembling machine code. *XFER replaces Acorn's Copyfiles with a much faster and easier to use routine. This allows tranfers from tape to ADFS which Copyfiles does not.

One particularly useful command is *MENU. This works in much the same as *MENU which you can use it in any Mode. When used with the ADFS it's a dream. After *MOUNTing a disc, enter the *MENU command and a copy of the directory is displayed with directory entries highlighted with a D beside them. There is an arrow which points to one of the entries. This can be moved around by the cursor keys. When an entry has been selected, pressing return will either try to load and run a program file of select a new directory display. ADT intelligently looks at the file's execution address to determine what type of file it is and will attempt to CHAIN, *RUN or *EXEC the file. Other keys can be used while in the menu mode. L will load a file into memory, whilst $, , and B help you move around the directory system. $ takes you back to the root directory, returns you to the parent directory while B takes you back to the Previously Selected Directory (PSD) in a similar manner to *BACK. It really is easy to move around directries using *MENU, so much so that I even use it just to select a directory.

ADT is full of other useful commands, *MLOAD and *MRUN enable loading of programs which have to be downloaded in memory. *SETADR allows you to change the load and execution addresses of a program - this can be very handy when used in conjunction with *MLOAD. *CATALL and *DIRALL are useful in preparing documentation on ADFS discs. *CATALL will display all the entries on a disc by directory (see Table 2), each level of directory being indented. *DIRALL does a similar job, but only displays the directories.

Plethora

Not easy to make comparisons with such a plethora of features available in different combinations. Although it can act as a utility, ACP only really makes sense if it becomes the default language and your 'front end'. ADU is specifically for the ADFS user and offers good value in that direction. ADT remains a comprehensive set of * commands.

ADU commands:
ADU
BACKUP
CATALL
DFSADFS
DIRALL
DIRCOPY
DIRDESTROY
DESCEDIT
DRIVE
FORMAT
KILLADU
**MENU**
PWRBRK
VERIFY
VFORMAT

Michael Spalter

TOOLKIT

Continue to build up your toolkit courtesy of Disk User this month with a dual catalogue creator and a break message routine

Dual Catalogue Creator
The Acorn DFS only allows 31 files per disk, and if you have many short files, you soon run out of catalogue space. Some DFS's allow 60 but this then makes your disks incompatible with other DFS's.
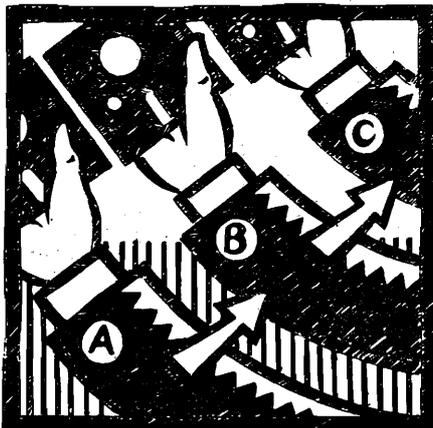
This utility will create two catalogues on the disk allowing 58 files in total.(There are two special files on each catalogue.) Typing '*/SWAP' or even '*SWAP' if it doesn't interfere with any ROMs will change over to the other catalogue. If you get a 'Disc Full' message, there may still be room on the other catalogue. The only input is that of drive number for the dual catalogue disk.

Note: This will destroy all files currently on the disk.

Break Message Routine
This utility will enable you to personalise the reset message by adding your own line of text, so that when <BREAK> or <CTRL-BREAK> is pressed your message is displayed (between "BBC Computer" and "BASIC").
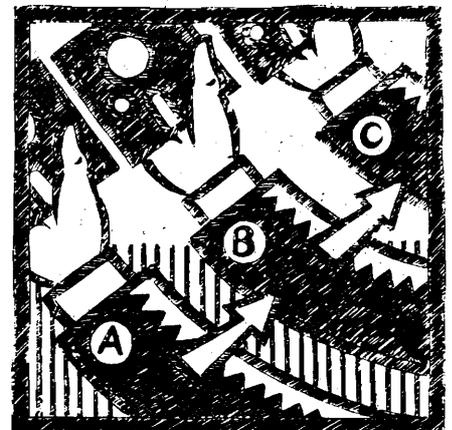
The program will then produce the machine code which can be saved to disk and used within your own programs.

COLLECTORS' ITEMS

Type the letter you like (A to Z)
to see moving pictures
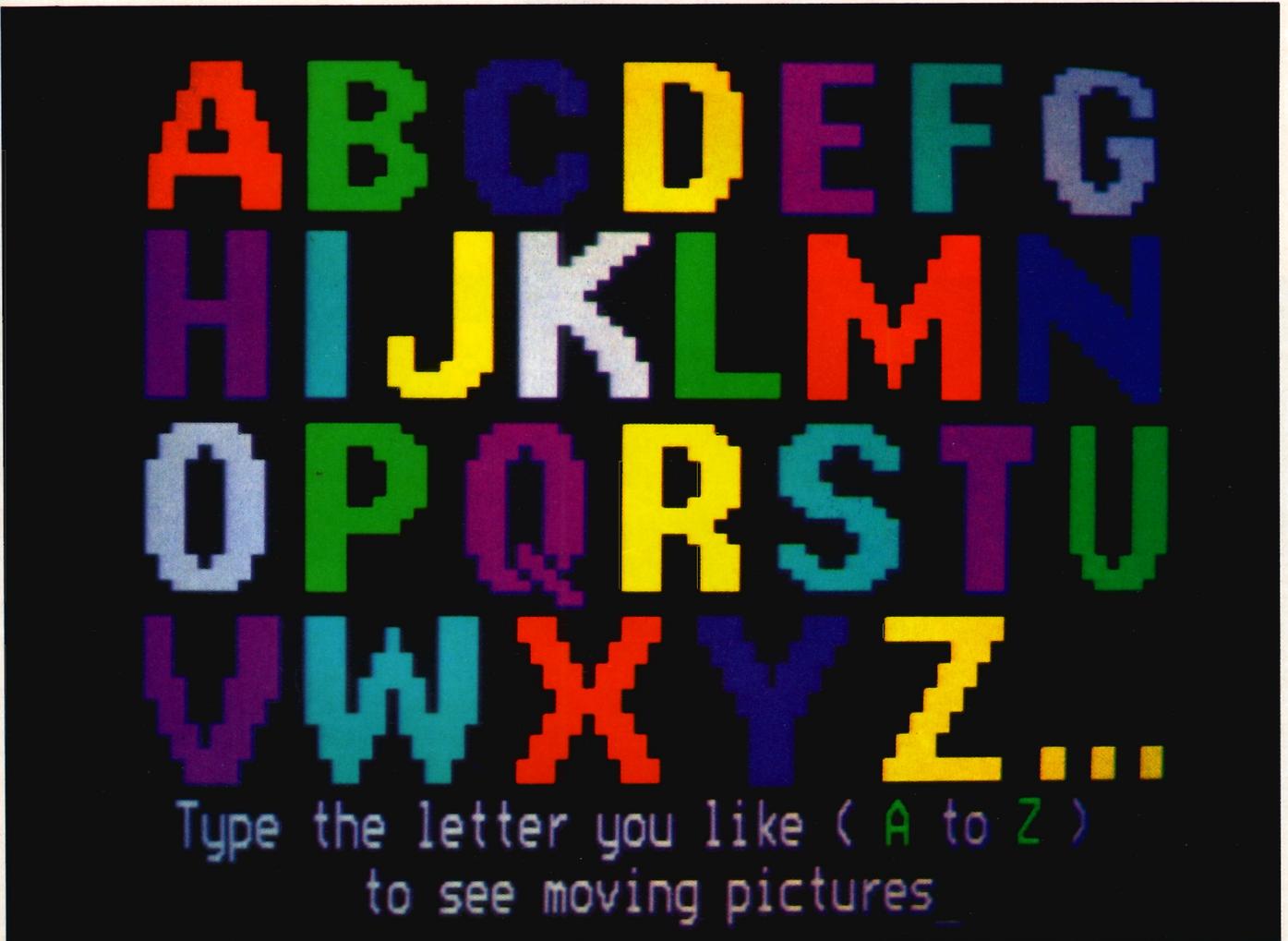
Animations from graphics wizard Abbas

We've been honoured in Disk User with the superb Mode 7 animations from computer artist Abbas. Mode 7 was chosen, not to exclude Electron owners, but because it is possible to provide decent length animations in fairly small data files and at acceptable speeds.

The colourful still frames are rapidly written to the screen from a RAM buffer.

Disk User will continue to feature animations from the A to Z collection and it is intended to publish the complete set at a later date. The 26 letters of the alphabet form a simple menu. Watch out for a special offer exclusive to Disk User readers. Meanwhile this month it's D for...

Richard Oldman

COUNT WITH TEDDY
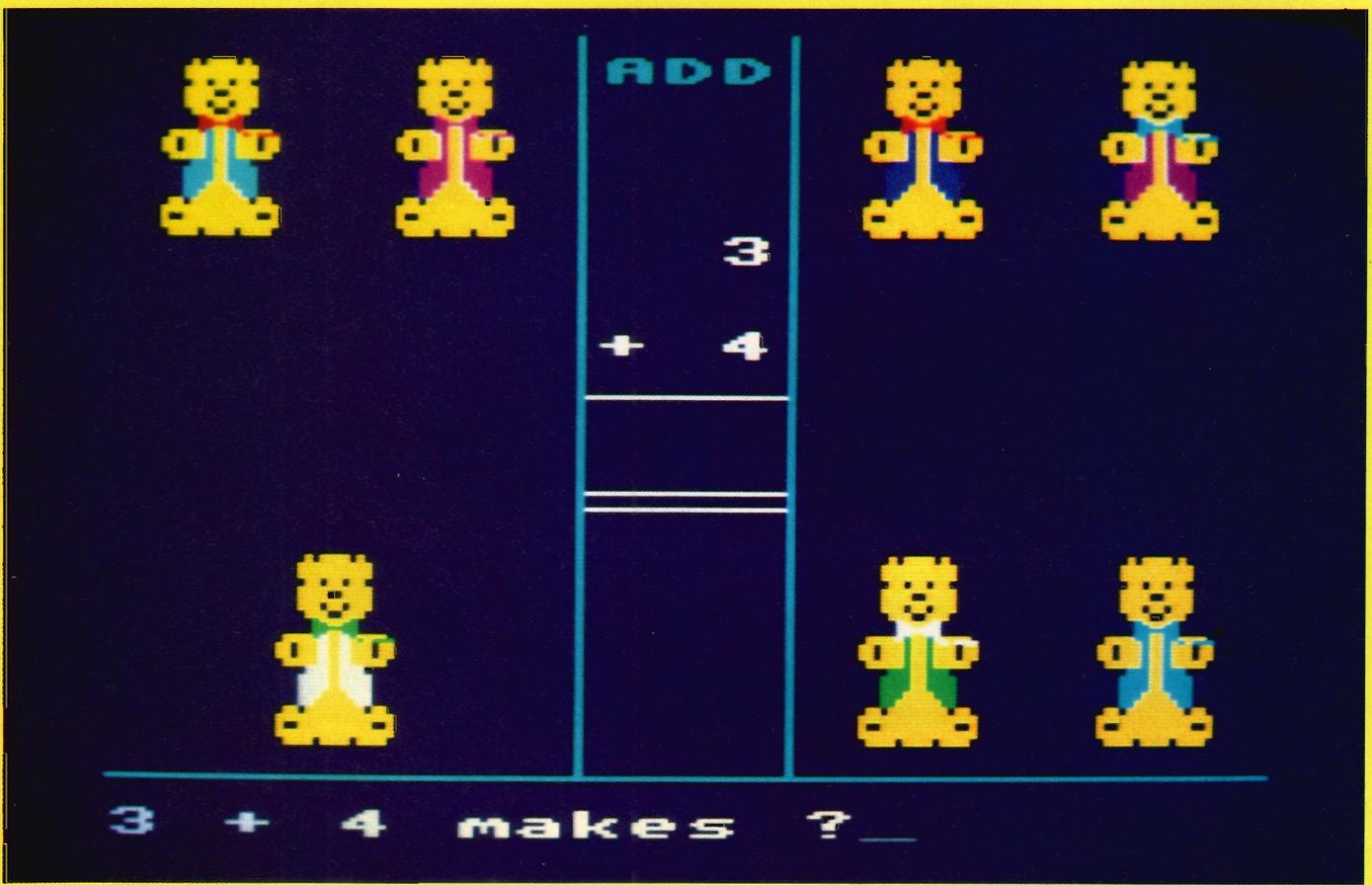
There comes a time when games have to take a serious turn

There I was zapping aliens (or had I just taken off?) when 'er indoors came home.

"Why don't you stop playing games and write something useful instead?", she asked.

"Like what?", I said.

"Like something that will help Tina learn to count".

"OK", I said, "I'll start now...."

Well, that's how this program started. It is quite simple and ideal for young children.

The screen is divided into two halves. A number of 'teddies' is displayed on the left and these have to be added to the 'teddies' on the right. The sum is shown at the bottom of the screen and also in the centre in 'vertical format'.

If, after three attempts, the child still gets the sum wrong, then the computer slowly adds the 'teddies' up with the child joining in. Another attempt at the sum is then allowed but if the child gets it wrong again, then a new sum is displayed. There are 10 questions in all and four levels of difficulty.

Program Notes

There are two listings to the program. Listing one disables the ESCAPE key, alters the flash period of the first and
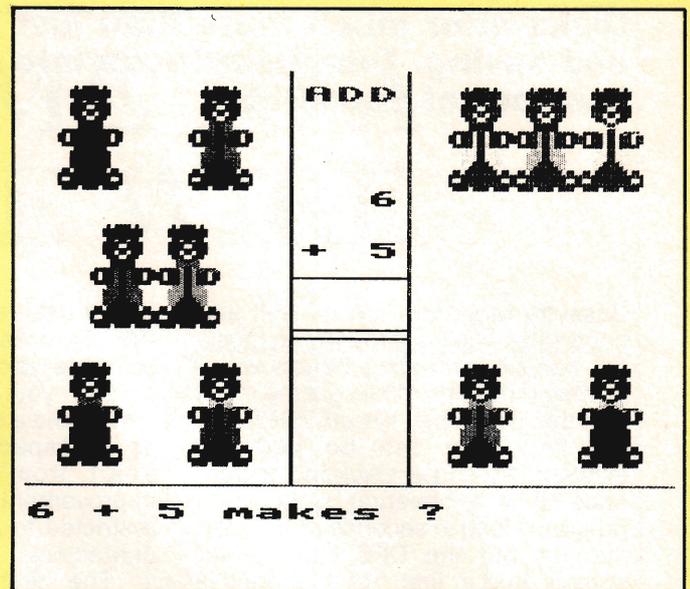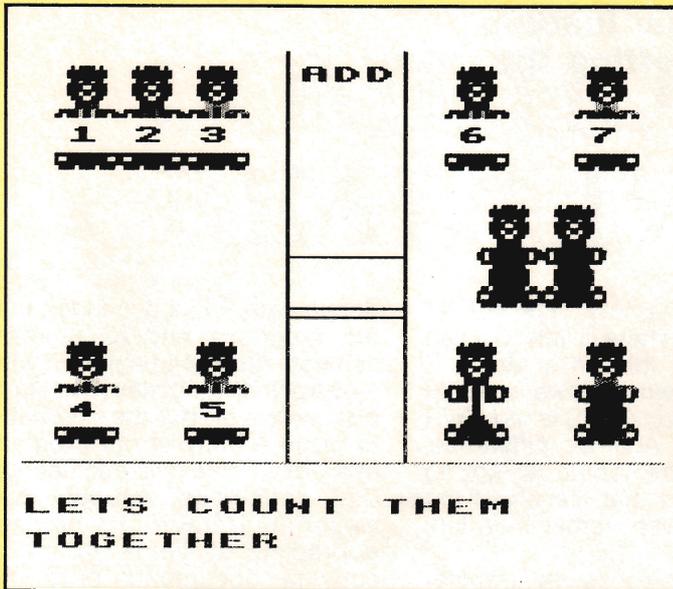
second colours, cancels the auto repeat, defines the user defineable characters and gives the instructions, which are displayed while the second part of the program is being loaded.

That brings us to listing 2, which contains the main body of the program. Lines 1 to 6 are included so that the program will run on disc based systems as well as cassette based systems.

PROCinit builds up Teddy from his respective parts. The A$ array holds the co-ordinates for the left hand teddies and B$ array for the right. The DATA statements for the co-ordinates get quite long. The left hand side uses DATA lines 310 to 340 and the right 350 to 390. I have tried to display the teddies in a symetrical fashion because I believe this helps with counting. For example, if the number one is generated, a teddy will be put in , say, position A. If the number two is then generated, the program does not just add a teddy next to the number one teddy — two new positions are used.

PROCdifficulty uses variable DIFFICULTY% to set level G% (lines 1640 — 1750).

The main loop, in MODE 2, is lines 80 — 130 and is quite straight forward.

PROCquestion handles the display as well as asking the question. Teddies are drawn by calling PROCteddy. Once the answer is given (line 600), PROCcorrect is called if it is right or PROCwrong if it isn't.

PROCteddy draws the teddies with random coloured waistcoats and bow ties.

PROCcorrect simply says 'well done' and moves on to the next question.

PROCwrong plays a 'wrong' noise and gives the child another go. It will do this three times until the variable D% is greater than 2, when it calls PROCtell.

PROCtell sets FLAG% to 1 giving, the child one more try. It then counts the teddies and shows the sum with the answer. There is then a pause before the screen is wiped and the sum displayed for a final attempt. This PROC uses *lower case* x% and y%

for the MOVE co-ordinates.

Once ten questions have been asked the results are displayed and another game is available.

PROCon and **PROCoff** turn the cursor on and off.

PROCwait is the delay and **PROCwipe** clears the screen.

PROCtick and **PROCcross** need no explaination.

Main Variables

A% number for left teddies (L$ used to find length)

B% number for right teddies (LL$ used to find length)

C% sum of A%+B% (LLL$ used to find length)

QUESTION question number

AN% answer typed in

FLAG%, D% see PROCwrong, PROCtell **X%,Y%** various loop indexes

x%,y% co-ordinates

V% random teddy colours

N% counts the teddies in PROCtell **E%** number of sums got right

F% number of sums got wrong.
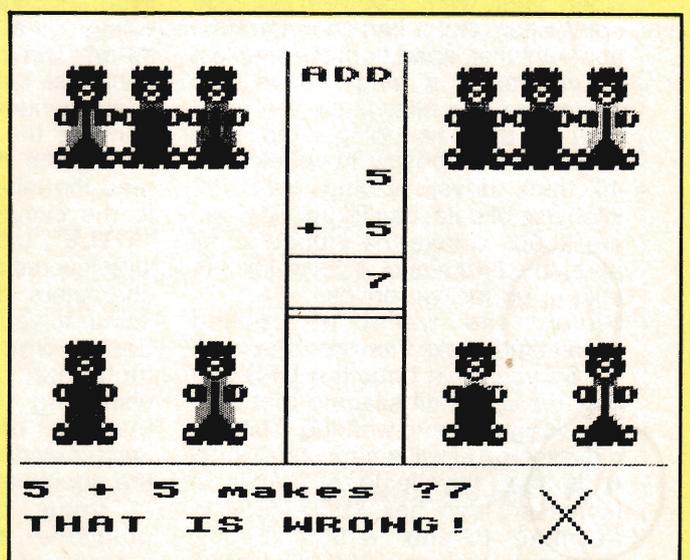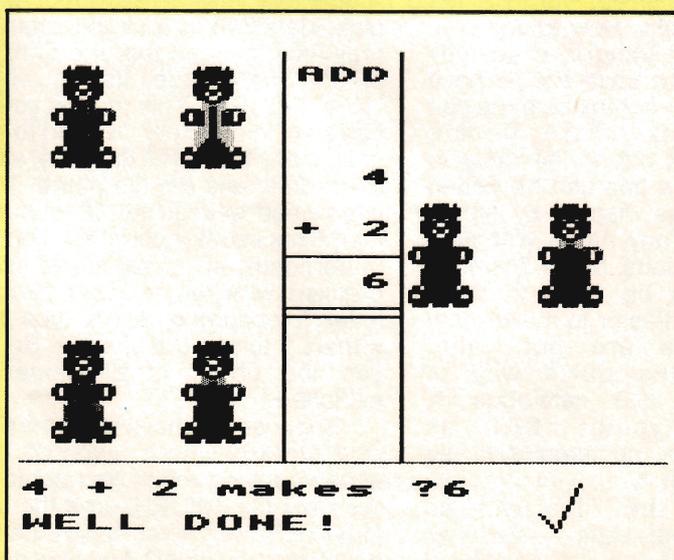
Disks offer much more than just fast loading and saving. Disk User looks into getting the best out of your DFS

Cassette recorders are all well and good when you want to listen to music but can be infuriating beasts when loading programs. While getting stuck into the disk filing system (DFS) for the first time can be both an exhilarating and a daunting experience. It is a cheering sight when programs load in seconds rather than minutes but the DFS can appear complicated at first. Yet all it does is take over many of the dreary tasks you used to do manually, such as finding the beginning of a file and getting the media (disk or tape) spinning. But I guarantee that after a week or two using disks, you will wonder how you did without them.

The convention I have used in this article is to put commands to be typed in in a **bold** typeface.

Which disk drive?

Most people start their disk-using out of usable memory. Type **PRINT PAGE** and see. There is little that can be done about this on a Model B unless you own sideways RAM (SWR) and a special DFS to use SWR as workspace. Another difference, apart from the obvious speed improvement, is that filenames are restricted to seven, rather than ten, characters.

The first step is to transfer software. Most serious software these days is disk compatible but games of course use all sorts of devious (and Acorn-illegal) crannies of memory as well as program protection. Many will prove difficult if not impossible to transfer. But the thrust of this article is to help you get the best out of the DFS, so I must assume you have transferred the software you want. Maybe the kind editor will let me write about software transferral at some future date....

documents. It is a good idea to keep the programs and documents on separate disks, particularly if you use a 40 track drive. It may result in more disk swapping but the alternative is to place a copy of the program on every disk containing documents and that consumes valuable space. Secondly, when it comes to upgrading at some future date, copying documents onto higher capacity disks, for example 80 track, is quicker if you do not have to keep deleting duplicate copies of the program.

Naming files is an important task to get right. Choose a convention and stick to it. You may remember today that "NEWFILE" is your latest masterpiece but you won't in a month's time. But what happens when you have several files of similar types? This is where directories are invaluable.

BRIDGE

career with a single-sided, 40 track drive which is perfectly adequate when your work involves just one application. But it can soon become apparent that, apart from the simplest of programs, a single drive has limitations and often means a good deal of disk swapping. And it is sometimes surprising how quickly a 40 track drive, holding only 100 kilobytes of data, can fill up. So if you are about to take the plunge, think about the future and, if the wallet can take it, go for two drives. Should the wallet be feeling overstuffed, make them double-sided.

So you have bought a DFS and disk drive; I shall assume that both are plugged in and working. The first difference you will notice on a Model B is that the value of PAGE is increased from hex &E00 (3584 in decimal) to hex &1900 (6400), immediately taking a 2.5K bite (sorry!)

Getting organised

Programs transferred, you are left with a pile of disks. How to organise them? The first thing is to identify each disk clearly. Each should have a label written in felt-tip, as biros can damage the disk surface. To save pulling the disk out of the drive to read the label, a title can be added to the catalogue display by typing ***TITLE** , then any name you want with few restrictions. Mode 7 teletext characters can be used to bring colour to your titles or to make them flash. Spaces are not valid. Incidentally, the quick way of displaying the disk catalogue is, instead of typing ***CAT** as recommended by the manuals, to use the abbreviation *****.

Some of the files will be programs, some data - such as wordprocessor or spreadsheet

Directories...

A directory is a compartment on the disk, identified by a single letter. You may have noticed that the third line of the catalogue listing shows "Directory :0.$". This means you are currently in directory "$" and to save or load any file in that directory, all you need do is use the file name. There is no need to use the "$" character first. Files in other directories appear underneath in a separate listing, prefixed with the relevant directory letter. To change directory, type ***DIR x** then *****. to see the change. So how can the power of directories be exploited?

Suppose you have written a piece of work which needs an accompanying letter. You could call the work file "WORK" and the letter "LETTER" but what do you call the next letter you write? And how do you

know, weeks later, which file goes with what letter? It is better to save the work file as "W.xxxxxxx" and the letter as "L.xxxxxxx" (where xxxxxxx is the subject matter). That way, when upgrading to higher capacity drives, it will be easy to keep related files together using the *COPY command.

...copying...

The ability to copy files with ease is another advantage of disk over tape. On typing *HELP DFS , the syntax of the *COPY command can be seen. Source drive and destination drive are simple enough but the <afsp> at the end (standing for alternative file specification) makes it a powerful utility as related files can be copied at a stroke. To use the example above, typing *COPY 0 1 *.xxxxxxx will copy the two files with

any single character. So entering *COPY 0 0 W.   will copy all files with names two characters long. Wildcards can be mixed with proper characters too: *COPY 0 0 T* copies all files in the $ (or root) directory that begin with T.

...and libraries

Directories can be used in other ways. You probably have a group of machine code utilities that may be called using *filename , AUTO for instance on this month's disk. But you do not want to get them mixed up with your latest magnum opus so the answer is to put them in a directory of their own - M, for example. By typing *LIB M , the computer will look in directory M for any *filename it does not recognize.

altered or deleted and it is a wise move to lock all files that should not be changed or erased. In common with most disk users I know, I do not bother and, as a result, have more than once had to poke around the disk with a sector editor to recover files I should not have deleted. It is not a procedure to be recommended if you want to keep your temper. File locking is achieved using *ACCESS (or *AC. for short). To lock a file type the command, the filename, then L ; to unlock it, repeat the above but without the L . The <afsp> function works identically to *COPY.

Other commands where a moment's loss of concentration can spell disaster include *BACKUP (which wipes out all programs in memory) and *DESTROY. Both need the user to enter *ENABLE (or *EN. for short) before the command will go ahead.

THE GAP

the same name but a different directory from drive zero to drive one on a dual drive system. With a single sided drive, you would type *COPY 0 0 *.xxxxxxx and the computer will prompt you to change disks when the copying buffer is full.

And here we bump into the main limitation of *COPY. The computer uses the whole memory between PAGE and HIMEM as a buffer to hold data to be copied, so makes sure you have saved any program you want to keep before copying or backing up. The buffer's size depends, of course, on the screen Mode; Mode 7 leaves you 25342 bytes, Mode 0 only 5886. The smaller the buffer, the more disk swapping you will have to do so change to Mode 7 first. The asterisk before the filename is a wildcard - that is, it can stand for any character or number of characters. The other wildcard is " ", which substitutes for

Safety first

Backups are next on the list of dos and don'ts. Do make backup copies using *BACKUP of all disks on which files have been modified.
No matter how reliable the disk drive, if the power goes down, you spill coffee or a mains spike corrupts the program which then messes up the disk, you will curse the lack of backups. Keep backups physically separate from originals, to avoid the coffee spillage problem.

The ease with which files can be created is paralleled by the ease with which they may be deleted. As a result, Acorn built in safety measures to avoid accidental mass file destruction.

Take a look at the files on the Disk User disk. They have the letter L after them, standing for locked, as a safety measure. Locked files may not be

Finally, starting programs by the simple expedient of pressing SHIFT-BREAK is a boon. Any disk containing a file called !BOOT can be made to do this. Firstly, the !BOOT file must contain a list of instructions identical to those you would have entered at the keyboard to start the main program. Secondly, you must have typed *OPT 4,3 with the disk in the drive. This tells the computer the file !BOOT must be *EXECuted (look at the catalogue header).
Other possibilities include the option to *RUN or *LOAD a !BOOT file - but that is beyond the scope of this article.

This has been a skimpy tour around the power of the DFS but space has run out and the editor says stop. But there is plenty more where that came from....

FEELING THE SQUEEZE

Feeling the squeeze

W.Kolkman

Here we bring you an up-to-date pair of utilities to help take some of the grind out of producing memory-efficient BASIC programs. These programs are for all the BBC micros up to and including the different versions of the Master. It even can cope with the new Master assembly instructions.

Program description

SHRINK is the ultimate, fast working BASIC compacter, and it is written in assembler. It compacts the BASIC program in memory located at PAGE. It removes all superfluous spaces, remarks, brackets. It shortens variable, function and procedure names. New names are evenly distributed throughout the alphabet, thus optimising speed of the BASIC program.

Assembler source code can be compacted too, including the new mnemonics for the Master 128.

There is a provision for using overlays (or "true chain") so variable names in a second program part get the same new names as were allocated in the first part of the program.

Further compaction can be achieved by using LINCOMP afterwards.

How to use

A BASIC program that you want to compact should reside in memory. With the disk containing the compactor in the disk drive in use, typing *SHRINK will activate the compacter. The compacted program can then be used immediately.

Pressing <F0> will restart the compacter on another BASIC program loaded into memory, without the need to read SHRINK in again from disk. <F1> should be used when compacting a second part of a program when using overlays to make variable names the same. Pressing <F2> will start LINCOMP.

Note that you should debug and save your programs before using the compacter.

Technical information

SHRINK is loaded in memory at &7000. A variable table is located from &6D00 downwards, and this will accumulate all variable, function and procedure names encountered. A check is made when the original program gets corrupted by the variable table. A table with line numbers referred to in the original program is located at &6D00 upwards.

The program cannot trap the special use of the EVALuate function, with function names entered in strings, as function names are compacted too!

Line Compacter

LINCOMP is a BASIC program compacter that creates lines as long as possible in an intelligent way. But note that after using LINCOMP there will be some lines that you will not be able to edit fully because of their length! Like SHRINK, it is written in assembler and is very fast in operation.

To use LINCOMP, a BASIC program should be present in memory. Typing *LINCOMP, or <F2> after the use of SHRINK, will start line compacting.

Technicalities

LINCOMP is located at &7000. A table with line numbers referred to in the original program is located at &6D00 upwards.

Note that the line compacter cannot cope with calculated GOTOs, when deciding which lines should not be combined with previous ones. Previous use of SHRINK is detected by looking for the text "SHRINK" high in memory at the end of the SHRINK utility, and just below the variable table.

F Munkert

CALCULATOR

Simulate a scientific calculator on the screen

This program simulates a scientific pocket calculator. An image of a calculator will appear on the screen and you can enter commands and values by moving a cursor around and pressing **COPY** when the cursor is positioned at the calculator key corresponding to the desired action.

Alternatively, commands can also be entered by typing in abbreviations on the computer keyboard.

The program is compatible with both tape and disc filing systems.

The program consists of two listings: **CALCU** and **CALCU2** . The second program actually generates the file containing the image of the calculator keyboard, in a graphics file called CALC, so this program should be RUN before any other use is made of the programs. If you wish to start the program directly, type:

CHAIN "CALC1"

and the program will commence.

Initially, commands have to be entered via the keyboard. If you prefer inputting by cursor keys and COPY, press ESCAPE to enter the cursor editing mode. The current input mode is shown in the bottom right corner of the screen.

Type in values and mathematical commands in the same way as you would with a pocket calculator. Users of Hewlett Packard calculators please note however, that this program does *not* work in reverse Polish notation. Calculations are performed according to the arithmetical operator precedence (BODMAS to those who remember him!).

When entering commands via keyboard use the symbols +, -, x and : to perform the basic operations of add, subtract, multiply and divide. The symbols * and / may be substituted for x and : if prefered.

M, RM and M+ are commands used with the calculator's ten memories. These commands must be followed by one of the ciphers 0...9 which address the respective memory cell. For example, to enter the value 100 in memory cell five type:

100 <Return>
M <Return>
5 <Return>

RM will recall the stored value and M+ will add a value to the value already stored in the memory.

Typing = will force the calculator to complete the operation in hand, even one it has suspended due to the precedence of operators.

Functions

The function YX will raise the current value to the power of the value entered next.

The following unary operators will compute the result of the function described applied to the value in the calculator's display immediately without need for the use of the = symbol:

X2 square
SQR square root
EX exponent function
PI produces the value of pi (3.14159 etc)
SIN sine

COS cosine
TAN tangent
ARC inverse prefix; ARC has to be followed by SIN, COS or TAN
LOG common logarithm
LN natural logarithm

n! the factorial function
C resets the calculator without affecting the memories' contents
" evaluates minutes and seconds of an arc
1/x the reciprocal function
& displays the current value in hexadecimal format
BIN shows the current value in binary format
DRG changes the current geometric angle unit (the current unit is displayed in the bottom left corner of the screen and is degrees, grads or radians)
TB affects the format used for representation of numerical values. TB must be followed by any of the ciphers from 0 to 8 or a full stop. The latter will select floating point display, which is selected by default, and the ciphers will select fixed format, where the cipher represents the respective number of digits to follow the decimal point.

When entering actions via the cursor keys, use the key marked with "E" to select the exponent and use "+/-" to change the sign of either the whole value or the exponent.

Commands can be entered in both upper and lower case. To leave the program type **OFF** or 'press' the OFF key on the calculator.

**Andrew B.
Hepstonstall**
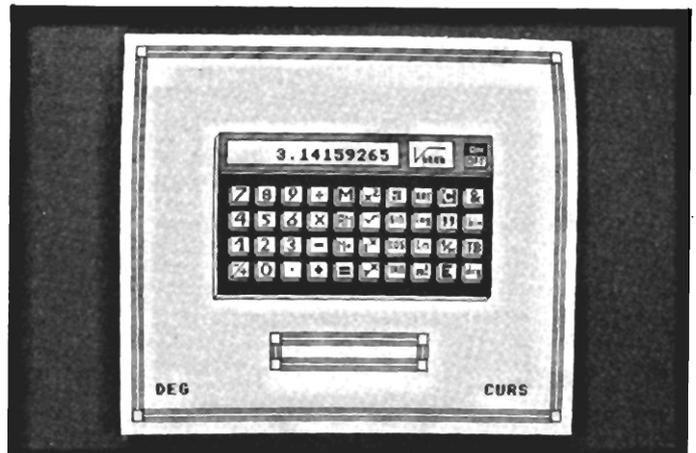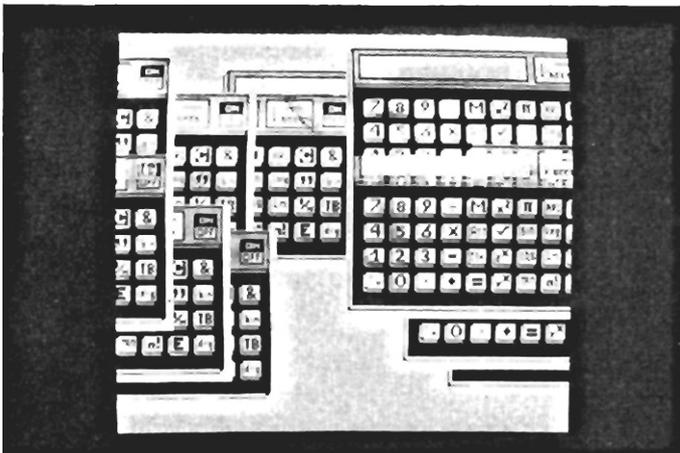
QUICK COPY

Does the time taken copying a disk drive you crazy? Help is at hand.

Probably the most frustrating thing about being the owner of a single disk drive is that it takes so long to copy a disk.

The method used by the disk filing system means that you have to switch between source and destination disks, pressing a key in-between, a great many times. This is very inefficient as only one file is loaded at a time, even if it is only a few bytes long.

Quick Copy gets around this problem by loading as many files as it can fit into the BBC's memory (up to 22K). In this way, if all the files are very small, they can be copied with just a few swaps of the disks. To give you an example, I used Quick Copy to copy a disk containing over 100K of data consisting of 31 files in four and a half minutes with only 10 swaps of the disks, instead of over 10 minutes and 62 swaps of the disks using the *COPY command.

The program also has the advantage of being able to copy from one side of one disk to the other side of the other disk.

Quick Copy in use

If you would like to use this utility, select Quick Copy from the *Disk User* menu.

If you wish to run Quick Copy directly, type:

PAGE=&1100< return >

This so that the program is LOADed lower into memory to allow more space for the programs being copied. Then type:

CHAINQUICK

where filename is the name you saved Quick Copy under.

The program will then ask you to enter the number of the source drive. This is the drive from which you want to copy programs. Once you have done this and pressed RETURN, you will then be asked to enter the destination drive, followed by

RETURN. This is the drive that you want to copy to. Drive numbers up to seven will be accepted, so that the program is compatible with the LVL double density disk filing system and RAM disks.

You will then be asked to enter the source disk, and press any key. Once you have done this, a *INFO ** command will be performed causing filenames and information about them to appear on the screen. This is so that this information can be loaded into the micro's memory, and it is transferred from there to arrays (this will take a short period of time). This is done because the information would otherwise be lost as it is overwritten with that of the destination disk when they are swapped.

The computer will then load as many of the files as it can into its memory, and when this has been done, it asks the user to enter the destination disk and press a key. The files are then transferred to this disk. This process is repeated until all the files have been copied.

If a file is greater than 22K (&5800) in size, then it cannot be copied by Quick Copy. This is because not all the BBC's memory is available to Quick Copy as some must be used to store the program and details on the files. When the program has finished, it displays the names of any files that have not been copied and instructs the user to copy these with the *COPY command. (Files of this length are very rare).

Procedures Used

PROCinit performs intialisation, setting up arrays ,etc
PROCinput performs the input, looping until the source and destination drive numbers are correctly entered
PROCinfo echoes the drive numbers to the screen, then prompts the user to insert the source disk, then makes up its own catalogue of the disk contents, including file lengths
PROCcopy copies as many files from the source as it can accomodate in the available memory using

PROCload, then uses PROCsave to write them to the destination disk
PROCload loads files into the memory
PROCsave prompts for insertion of the destination disk, saves all the files in the memory to this disk using osfile, then prompts for re-insertion of the source disk
PROCoscli changes the source drive
PROClarge deals with files which are too big to be accomodated in the memory by displaying a message to the user suggesting that some alternative means of copying these files is used.

Variables.

A% used to hold the value of the contents of the accumulator
adj% used in a FOR...NEXT loop which removes trailing Fs from addresses
big% set to TRUE if one or more files are longer than &5800
C% holds the number of files on the source disk
D% holds the number of files that are about to be saved
E% holds the number of files that have been saved so far plus the number that are about to be saved
from% holds the number of the source drive
L% used in a FOR...NEXT loop
loop used in a FOR...NEXT loop
loop2 used in a FOR...NEXT loop
T% used to keep a running total of the ammount of memory used so far
T1%(x)-T6%(x) hold the addresses and names of files on the source disk
text%(x) holds operating system commands to change drives
to% holds the number of the destination drive
oscli holds the address of oscli
osfile holds the address of osfile
X% holds the low byte of the parameter block for loads and saves
Y% holds the high byte of the parameter block for loads and saves
comm$ holds the operating system commands to change drives
**G$** holds the result of a GET$ (only used to wait for any key to be pressed)

A&B COMPUTING

FOR BBC MICRO, BBC MASTER & ELECTRON

£21.80
for 12 Issues U.K.
£26.00 for 12 Issues
Overseas Surface Mail
£53.00 for 12 issues
Overseas Air Mail
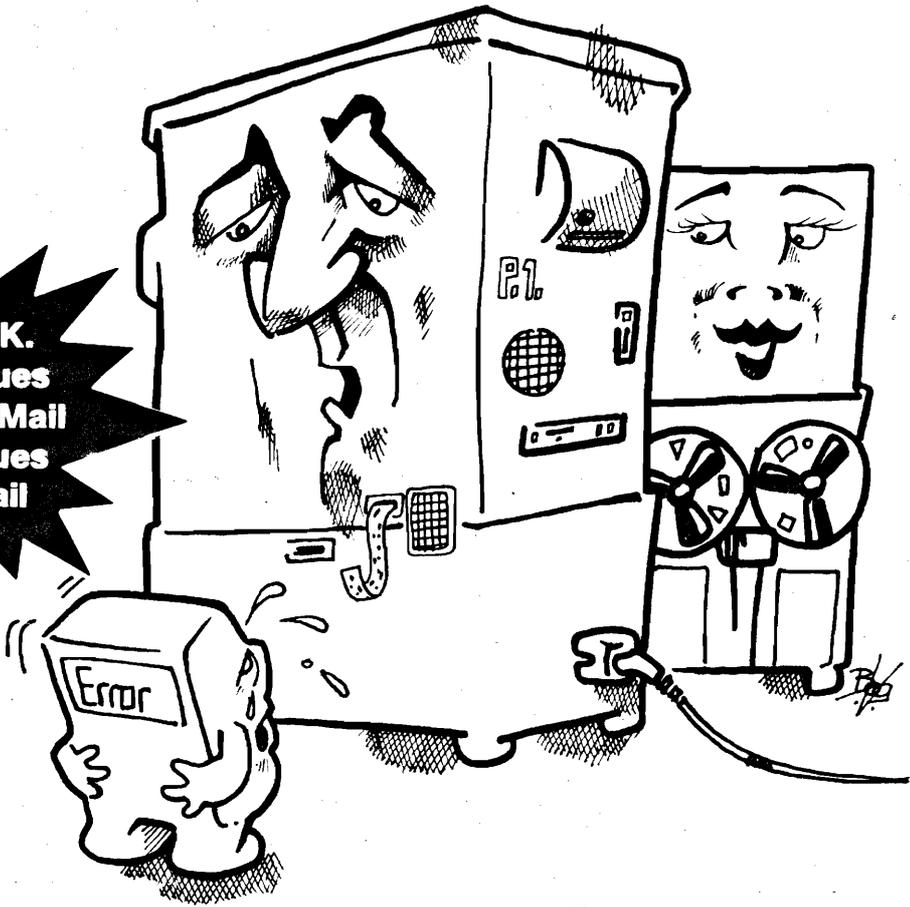
Listen Son, You may be a **2**nd Generation Computer
but just remember that compared to a
human you've only got the
brain-power of a
HADDOCK.

CRAZEE RIDER

SUPERIOR SOFTWARE LIMITED

A PROGRAMMING BREAKTHROUGH
The first Electron motor racing game to faithfully represent the bends and chicanes of major international racetracks.

Setting the Scene	Going round the bend at Misano (BBC MICRO VERSION)	Knocking off an opposing rider (BBC MICRO VERSION)	Clever Electron progamming techniques (ACORN ELECTRON VERSION)
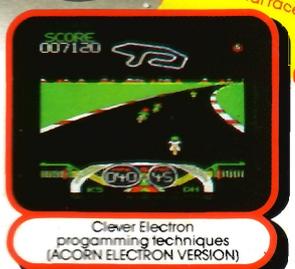
If you can't beat them, knock them off!

Howling round the bends at Le Mans at 150 m.p.h. is a daunting experience. You'll need nerves of steel and slick reflexes to finish in the Top 6 and qualify for the next race. If, at first, you can't make the grade you can always let off steam (and gain bonus points) by bumping into the other riders and knocking them off their bikes; listen for the terrified wail as the rider is sent flying to the ground.

Each race has 60 participants — and the further you get, the tougher the competition becomes. Bonus points are awarded for a good finishing position, and your "hit-count" shown at the top of the screen records the number of other riders you have sent tumbling from their steeds.

Seven top international racetracks are featured:-
● Le Mans (France)
● Anderstorp (Sweden)
● Paul Ricard (France)
● Brands Hatch (U.K.)
● Misano (Italy)
● Silverstone (U.K.)
● Nurburgring (W. Germany)

An enhanced version has been specially produced for owners of the BBC Master Series computers. Included free with all BBC Micro cassettes and discs, this version features several extra race circuits, a beautiful 6-minute musical soundtrack, and a host of extra game features — making full use of the Master's extra memory and capabilities.

BBC Micro Cassette......... **£9.95**   Acorn Electron Cassette... **£9.95**
BBC Micro 5¼" Disc .... **£11.95**   Acorn Electron 3½" Disc. **£14.95**
BBC Master Compact 3½" Disc **£14.95**

(Compatible with the BBC B, B+ and Master Series computers).

PRIZE COMPETITION

A superb MZ Simson sports moped (illustrated above) worth £500 is the first prize in our competition.
To enter the competition, you must finish in qualifying position on the first 21 racetracks, and note down the congratulatory messages you receive.
Closing Date: 31st March, 1988.