An Expensive Alternative?

. . . That seems to be the verdict on 68K/OS – the QL's 'other' operating system.

The alternative to the QL's built-in operating system QDOS, called 68K/OS is now available as a £99.95 package which turns out to be a premium price for a product whose viability must be under question.

Why pay £100 on top of the QL price for an operating system that turns out to be merely an alternative as opposed to a 'go-for-it' replacement?

The suppliers, GST of Cambridge had originally planned that K/OS would become a de-facto standard operating system for Motorola 68K systems, and Motorola's support goes some way to ensuring this, but only for the OEM market.

There can be only one reason for a QL owner (version AH and above!) purchasing K/OS – to write applications and possibly systems programs for the potentially large K/OS market. To buy because it seems preferable to QDOS turns out to be a false investment.

The system supplied by GST to end users will plug into the QL's expansion port with a switch to select between QDOS and K/OS. The version tested consisted of two 16K EPROMs that had to be swapped for the QDOS EPROMs already in the machine, a trivial process on a 'dongled' QL as later versions require three PCB alterations.

A number of utility programs are supplied on a microdrive cartridge and the package reviewed also included the K/OS version of GST's QL assembler – the one Sinclair is adopting as the official QDOS assembler. Unfortunately!

When a system containing K/OS is booted up a red screen is drawn with five screen layout options. Pressing the requisite function key sets the layout, which is where the first problem arose. All five of the screen modes got the windowing slightly wrong losing the extreme right hand edge of the display, on both an RGB monitor and a domestic television. No doubt this could be cured with Microvitec's special QL monitor.

After selecting the screen mode the system's **data** and **program** devices are set to **ROM:** and the command program is initiated. This is called **ADAM** and provides a basic interface between the user and the machine. The command program operates in two distinct modes, system and user. In the system mode programs can be suspended, killed, released and alternated between, and OS functions such as window size can be altered.

The user mode is that in which individual programs are run. The system is naturally a time sliced, single user multitasking environment, just like QDOS, so more than one program can be present (and possibly running) at one time. Whenever a program is loaded it must be derived from something other than the initial ROM setting as this only contains the ADAM program. Instigating another device is known as **mounting** and must be done each time a new microdrive cartridge is put in. This process sets the default pathname to the device specified, and if this supports a directory structure (such as floppy discs and microdrives) then this directory is defaulted too. Once the pathname has been established (on either the program, data section or both) files and the like can be loaded and run simply by specifying their file name.

If a file is to be loaded from another device then as much of the full **device:directory/filename. extension** path as is required by that device must be used. This is standard and very similar to Unix™.

68K/OS supports all the normal QL devices but tends to use different names: SCR_ is SCREEN:, CON_ is a mixture of SCREEN: and KEY:; microdrives are MD: and the serial ports are TXn: and RXn: depending whether they are being used for input or output. There is also the PIPE: device, which exists within QDOS but is not advertised by Sinclair.

Documentation consists of two A4 folders, one describing K/OS to the novice and one describing the system and all its calls to the programmer. The former manual also contains a description of the assembler. These manuals are written in an easy style with the odd bit of flippancy – mainly knocks at QL hardware! The introductory manual is a little thin on facts and tends to dedicate more time to K/OS utilities than the operating system.

Being eager little beavers this manual was followed to the letter, which resulted in getting nowhere fast – an attempt to run any program re-booted the machine. The fault was eventually discovered in the date and time program, DATE.PROG, which runs concurrently with anything else the machine is doing. This caused any other program to crash, so the date program was ignored and progress was a little faster.

One gets the feeling that GST is proud of getting the operating system to fit into 32K, but it must be remembered that a substantial number of essential utilities are held on the supplementary microdrive cartridge.

The manual offers a relaxed tutorial which includes the backing up of the system cartridge using the COPY.PROG program. This uses standard wildcard syntax and can have its parameters passed to it on the command line, such as

COPY.PROG *.* BACKUP/

or each parameter can be entered in turn in response to the prompts. Unfortunately errors or misunderstood commands at this stage result simply in a further 'Copy from?' prompt. No indication of the actual error is given and the unfriendly syntax can confuse the novice, but it is at least less verbose than QDOS.

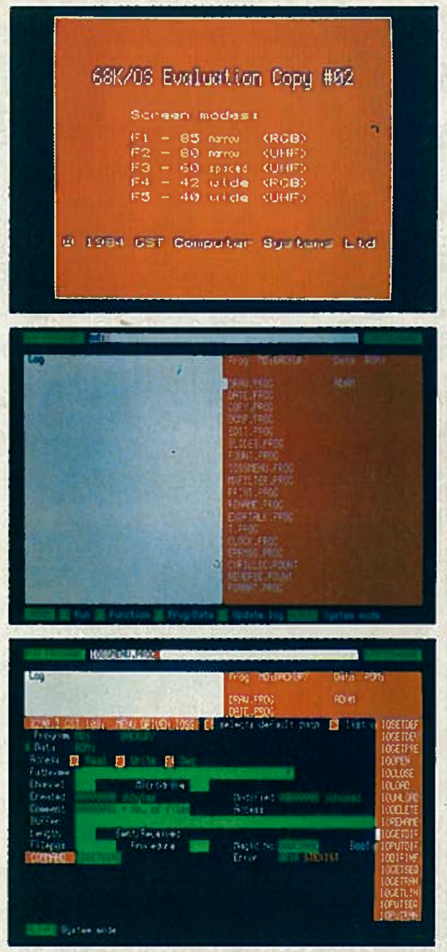
The use of COPY.PROG is invariably preceded by the use of FORMAT.PROG, which formats the microdrive cartridges prior to use. This program is far more impressive as it does not allow the formatting of previously-used cartridges without authorisation, and it reports the state of affairs if it thinks the cartridge contains data saved under ZX Spectrum format. It is unfortunate that it regarded QDOS format as Spectrum format too!

The utilities provided on the microdrive cartridge are:

1) **COPY.PROG:** which copies files from one device to another.

2) **DATE.PROG:** which (when it works) displays the time in the extreme bottom right hand corner of the screen.

3) **DRAW.PROG:** a utility graphics design program designed to show off the capabilities of the QL's operating system. It is rather tedious to use through having to constantly switch between two menus when drawing a figure at a specified position, but it is something that doesn't come with the QL. The QL on the other hand does come with four rather more useful pieces of software.



```
68K/OS Evaluation Copy #02

Screen modes:

F1 - 85 narrow  (RGB)
F2 - 80 narrow  (UHF)
F3 - 60 spaced  (UHF)
F4 - 42 wide    (RGB)
F5 - 40 wide    (UHF)

© 1984 GST Computer Systems Ltd
```

GST's K/OS in full flow – the startup screen, the starting position just after mounting a drive and one of the utilities in action.

4) **DUMP.PROG:** a program that will dump (i.e. print out) the screen to a printer. Its drivers are set up for an Epson FX-80 and there doesn't seem any way to alter this.

5) **EDIT.PROG:** a full screen editor that suffices, but I am inclined to use the Metacomco screen editor when creating text files. If the QDOS and K/OS microdrive formats were the same I would expect others to follow suit.

6) **EXORTALK.PROG:** a terminal emulator that configures the QL to act as semi-intelligent terminal for such things as Motorola EXORmacs. It didn't seem to work on the version we received – once the HELP menu had been selected all keys failed to produce any further effect.

7) **FORMAT.PROG:** used to format microdrive cartridges. The process is a little different to the QDOS format, so cartridges prepared on one system are not (easily) readable on the other. K/OS sectors are 1,024 bytes long.

8) **FOUNT.PROG:** allows the user to select a pre-defined fount (of which a few are provided) on a selected channel. Could be interesting.

9) **IOSSMENU.PROG:** probably the most interesting program for hackers. It makes all the K/OS calls for input and output directly accessible but screen layout is unfriendly and error messages unhelpful.

10) **MXFILTER.PROG:** converts Motorola S-records to K/OS procedures and programs. Useful if you're serious about program development and use a system that creates Motorola S-records to assemble your source, otherwise a triviality.

11) **PRINT.PROG:** essentially a printer driver for the screen editor, again permanently configured for an Epson FX-80.

12) **RENAME.PROG:** an obvious function.

13) **SLIDES.PROG:** another piece of trivia that displays screens full of information as a sort of slide show. It may be useful for the odd shop-assistant programmer. Pictures displayed are created with DRAW.

14) **T.PROG:** written in Pascal (all the others are in 68000 code) and serves absolutely no useful purpose at all, as GST admit, although considering it to be of great amusement value themselves.

Another microdrive cartridge which costs a further £39.95 contains the assembler and two 'include' files holding K/OS definitions and keys. See the review elsewhere in this issue for views on that.

Briefly the main objection is that K/OS is an alternative to, not a replacement for QDOS. Now prove us wrong!

For the technically minded, QDOS uses a system of 68000 TRAPs to perform most of its functions, register D0 being used to hold the function key and return the result error code. This ensures that certain operations are not interrupted by other programs requiring the same facilities, so that time sensitive programs can be simply implemented.

68K/OS on the other hand follows the time honoured system of vectored addressing to each routine. Each method has its advantages and shortcomings, but TRAPs seem so much easier to write around. Being purely an operating system with no interface to the BASIC language, K/OS also lacks a set of floating point routines, or at least any present do not appear to be available to the user. With the Sinclair system of 6 byte numbers with offset exponents and un-normalised mantissas, this may be an advantage!