

QL corner

by Sid Martin and Timothy Green

Sid and Tim report the latest QL and Thor news, uncover the secrets of multi-tasking, and still find time to answer your letters

THOR News

CAMBRIDGE Systems Technology, designer of the Thor XVI, has re-surfaced in Denmark, home of backers Dansoft and the new marketing firm Thor International. CST has new offices just outside the Danish capital, but somehow 'Copenhagen Systems Technology' doesn't have quite the same ring about it!

Dansoft boss Helmuth Stiven has been rushing round re-assuring people that Thor production is still underway despite the move, management hiccups and wild claims from competitors who would like to have got there first.

Thors are now being made in Denmark, and Helmuth insists that all outstanding orders will be satisfied by mid-December. The Russian deal is going ahead, and they have asked for a 32-bit 68030 version which is in the works now that CST are free to concentrate on product development.

A new release of Argos, the Thor XVI operating system, is due in January. This will include improved SuperBASIC and device handling.

Thors are now being manufactured by a large Danish lab equipment firm, Bruel og Kjaer, who have also taken over from CST the task of finding components. Thors are being made in all models from the 512K network station to the 20 megabyte Winchester disk variant; a new 40 megabyte version is 'almost ready'.

Thor International had not appointed a new UK distributor at the time we contacted them, but the appointment was imminent - we hope to have details next month. UK servicing and repair work has been taken over by P.M. Engineering, Unit 8, Cromwell Business Centre, New Road, St. Ives, Cambridgeshire, PE17 4BG.

For the time being Thor International need to build confidence, and the best way they can do that is by building and shipping machines to customers.

The new phone number is in Denmark, which has the dialling code 010 451 if you're ringing from the UK. Danish afternoons start an hour earlier than ours, so you'll need to call before 4pm GMT.



The Thor. Today Copenhagen. Tomorrow, Valhalla?

The number for Thor International is 93 03 05. Postal enquiries should be sent to Lundeskovsvej, 5, 1, 2900 Herrerup, Denmark. Letters sent to the original Stevenage address are being forwarded, but they take a while to crawl across Europe.

Thor Zone

If you want instant contact with a Thor, you can't get much more intimate, computer-wise, than by calling *Zone*, a new interactive modem game. *Zone* runs on two linked Thors, which handle incoming calls and commands; the master control console is a customised Spectrum.

Zone, short for *Erogenous Zone*, is a sexual adventure fantasy played by computer. Up to eight people equipped with a modem can compete or co-operate in the usual computer-game way - except in this game you gain points by bonking, rather than fighting! Drink is much featured, and has advantages and disadvantages, as you might imagine.

All the action gets typed over the telephone line, so the risk of viral infection is confined to your micro. Of course, this arrangement has its disadvantages too, but not from the point of view of British Telecom.

Zone has been running in a

small way for a while, but the original system could only handle two 'phone lines at once, rather limiting the players' choice of partner. There are a few computer moderated characters, or 'mobiles', in the game - these

include a Sloane Ranger and an unconventional nun.

A few weeks ago a 'multiplexer' was hooked up, so that *Zone* could cope with eight simultaneous users. It was not long at all before the first interactive orgy got underway...

You can play *Zone* by hooking your computer to the 'phone and calling (01) 994 9119. The communications protocol is eight bits of data, no parity, one stop bit. The display is scrolling text, so you can't use Prestel software.

Two speeds are supported: 1200/75 baud or 300/300. *Zone* who laughs last is at 300 baud, as the character 'HonkHonk' on the relatively pious game *Gods* is fond of saying.

QLby design

Most of the best things about the QL stem from its design, which is unusually flexible and expandable. One useful feature is 'multi-tasking' - the facility to run several programs at the same time.

This month we explore multi-tasking and explain how it works. Next issue we'll take a look at 'Slave Blocks' - the QL feature which speeds up disk and micro-drive access.

Most micros run one program

Quanta workshop

Over 200 people attended Quanta's winter QL User Group workshop outside Northampton. The three-day workshop included talks by Tony Tebby (designer of QDOS and QRAM), Simon Goodwin (*Supercharge*, *Turbo*, *Speedscreen*), Dave Walker (*DiscOver*) and other famous names from the QL world.

The sessions started on Friday evening and continued until Sunday. The ground floor was used for talks, while dozens of computers were set up and working in a large room on the first floor. Most of the machines were expanded QLs, but there were also a couple of Atari STs running Strong Computer Systems QL emulator, and one new Thor XVI.

A new kind of 'QL clone' put in an appearance, courtesy of Quanta member Mike Lilley. His 'QL-PC' is a souped-up QL based around a new circuit

board available from Quanta.

Other members were running Winchester disk drives and amateur radio equipment, using home-made interfaces under QL control.

The workshop was mainly intended as a gathering-place for QL enthusiasts, but several commercial suppliers turned out, including Adman Services (spares) Creative CodeWorks (*Speedscreen*), Tony Firshman Services (spares and communications packages), PDQL (Thors and business programs), QJump (GRAM) and Sector Software. It was a good meeting, and proof that there's still a lot of life in the QL scene.

The next Quanta weekend workshop will be held on 11th March 1989. Call the club secretary Phil Borman, on (0472) 49850, if you need further details, or write to the address given in the 'Club Contact' section.

Continued on page 130

Continued from page 128

at a time. When you turn the computer on it runs a program built into its circuitry, which loads the 'operating system'. This lets you issue commands and look at files and directories, but its main purpose is to allow you to load other programs - games, packages, compilers, editors and so on.

The processor executes each program when it is loaded. When it finishes control returns to the 'operating system' so you can select and load another one.

This is all very well, but it takes a while to load and set up each program, so you can't chop and change between them as quickly as you might like. Furthermore, many programs waste most of their time waiting for disks, or data from a port, or just waiting for the next key-press.

You might get more done if you could load several programs at once, and swap between them in response to whims or telephone enquiries. The machine could use its spare time to get on with 'background tasks'.

When you turn on the QL there is no need to load the operating system, because it is programmed into chips inside the computer and on expansion boards. In fact there is no operating system, as such - just an expandable collection of routines which can be called by any task, and one pre-set task, the SuperBASIC interpreter.

SuperBASIC is a programming language; the task is called an interpreter because it interprets commands and program

Many programs waste most of their time waiting for disks, or data from a port

lines as they are typed in. You can use SuperBASIC to load other tasks, but SuperBASIC carries on running alongside the other tasks. Processing time is shared between the tasks, and you can re-prioritise them or swap between them at any time.

QL tasks use a 68000 processor feature called 'position independent' code. This means you can load and run them anywhere in memory. It doesn't matter what order you load tasks in. Older processors expect code to run at fixed places in memory, so clashes crop up if you try to load several at once.

QL tasks can be any size from about 20 bytes up to the limits of your memory. Compilers boil them up from SuperBASIC, C, Pascal, Forth, and other languages, in seconds. There's no excuse for writing a QL program that won't multi-task.

Tasks tend to be quite concise, as they link themselves into existing routines to handle commands, calculations and devices. System routines have a standard interface, regardless of the detail, so a task written on a microdrive system will use floppy, Winchester or RAM disk without changes, if you load it on an appropriate system.

The QL shares its time between tasks using 'time slices'. Several times a second it stops what it's currently doing and considers swapping to another task. Tasks run in short bursts, one after another, so that several programs appear to run at once.

Interruptions

The QL considers swapping between tasks whenever the current task calls the system to use a device, or the mix of tasks changes. Some tasks may not call the operating system regularly, so to keep multi-tasking smooth, an interrupt signal is sent every 50th of a second. This diverts the machine into the 'scheduler' unless it's already running a system routine.

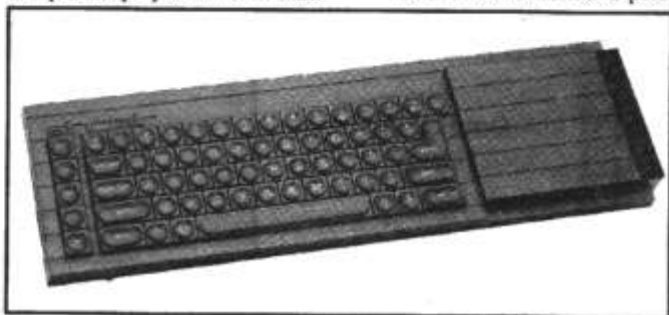
The scheduler saves the contents of processor registers and other details of the processor state, in a space at the start of the 'current' task. It will use this information later when it decides to re-start that task. Then it looks through the other tasks to find the next one to be executed.

Priorities

The QL scheduler works very

simply: each task has a 'priority' associated with it - a number between 0 and 127. Tasks with a priority of 0 get no processing time. Others get a share of the available time which depends on their priority, and the priority of other tasks.

Relatively high-priority tasks get a greater proportion of available processing time than tasks of lower priority. The exact ratios depend upon the number of tasks being run, the devices in use, and the precise purpose of each task.



The QL. Better by design

You can get a feel for the way the scheduler performs by running several copies of the same task, at various priorities.

Absolute priority

If two tasks run at a priority of 4, and one at a priority of 12, the higher-priority task will receive about as much time as the other two put together. It is only the RATIO of priorities that matters - not the values themselves. The same is true if the priorities of the tasks are 6, 6 and 18, or 40, 40 and 120.

As far as the QL is concerned, there's no such thing as an 'absolute priority'. All priorities are relative.

Unequal ratios

The ratio of timings does not correspond directly to the ratio of priority values. Timings change less than priorities.

For instance, consider a priority ratio of 10. This might mean one task at priority 90 and another at priority 9. It gives a speed ratio of 8 between two tasks.

A priority ratio of 5 gives a speed ratio of 3.7. A priority ratio of 3 causes the task with the higher priority to run 1.9 times as fast as the lower-priority task.

According to Sinclair's system documentation, and every book on QL machine-code, priorities are numbers between 0 and

127. But *Shopper* correspondent and QL guru Simon N Goodwin has discovered that the system call MT.PRIOR accepts higher priority values, in the range 128-255.

There's no built-in SuperBASIC command to call MT.PRIOR, but QL Toolkits remedy this deficiency, with extra commands like SuperToolkit's SPJOB or Supercharge and Turbo's SET-PRIORITY.

SET-PRIORITY follows the book and rejects priorities greater than 127, but SPJOB and similar commands let you set higher priorities.

Negative values work, too - SPJOB 0,0,-1 sets the priority of SuperBASIC (task 0,0) to 255, and other negative values -2 to -256 correspond to priorities from 254 down to zero.

Each 'new' value has a posi-

tive and negative representation, so it's interesting to see how Toolkit commands display the values. The SuperToolkit JOBS command shows non-standard priorities as positive numbers, but LIST-TASKS shows them as negative.

Group tasks

So far we've only considered pairs of tasks. When three or more tasks are active, the ratio of execution timings only bears a vague relation to the relative priorities. It takes a big difference of priorities to make a difference in the speed ratios.

Tasks tend to settle out into 'groups' which run at much the same speed, despite small variations in priority.

If three tasks are running at relative priorities of 1, 2 and 3 (or 40, 80 and 120) the one with lowest priority will run at about half the speed of EITHER of the others.

Priority ratios of 1, 2 and 4 make the highest priority task run at about twice the speed of the other two. Three tasks at relative priorities of 4, 5 and 6 run at much the same speed.

We tested non-standard priorities by running three compiled SuperBASIC integer print loops at the same time. One ran at a priority of 255 and the others at the default priority, 32.

Continued on page 132

The 'zine that reads itself

The first three issues of the magnetic magazine *QL Adventurer's Forum* are now available on 3.5" disks, thanks to *Shopper* reader Michael L Jackson.

QLAF is now published on boring normal paper, but the first three issues came out in the form of *Quill* files on microdrive cartridge. Cartridge and 5.25" disk editions are still available from CGH Services (address as for QLAF, in the Club Contact section).

The new 3.5" disk version costs £3.50 including the disk and postage. Michael has compiled a 'front end' program that lets you read the files without loading them into the *Quill* word-processor first. This makes the magazines much easier to read. For a copy, send £3.50 to Michael L Jackson, 17 Stanhope Road, Swadlincote, Burton on Trent, Staffs DE11 9BQ.

Continued from page 130

Don't be shy

If you've got a tip or a question for *Sinclair Scene*, please write to us, c/o *Computer Shopper*. We can't reply personally but we can answer your questions through the column. There's a free subscription up for grabs to the reader who sends in the best 'tip' each month, so don't be shy - share your discoveries with other Spectrum, QL or Z88 users. We are also interested in publishing short, useful utility programs and routines, in any language.

The high-priority task counted about five times faster than each of the other two. This matches the results for similar ratios of conventional priorities, so it seems that values above 127 work as an extension of the normal range. Both the QL and Thor XVI seem quite happy to run tasks at priorities up to 255.

Degradation

The number of tasks loaded does not have much effect on the total amount of processing time shared between tasks. When a task is waiting for data it is suspended and consumes very little process-

ing time.

Swapping between tasks is fast, and happens at the same speed regardless of the number of tasks. Even if there's only one task running, QDOS still swaps it in and out every time it runs the scheduler.

If a test-run takes 27 seconds from BASIC, four concurrent runs from tasks will take 112 seconds, with the interpreter as an idle fifth task, waiting for the next command.

This is only four seconds more than if the runs were performed sequentially by a single task. The overhead of multi-tasking is small compared with the convenience - you can type commands or use BASIC as the concurrent tasks run.

Overheads

The QL could run a bit faster if it never allowed multi-tasking. We measured the 'overhead' of multi-tasking and other 'background' activity by adapting our speed test routine, listed in *Shopper* issue 2, to run with no interruptions allowed. This boosted the effective speed of all types of RAM by about ten per cent.

This overhead is taken up by the 'housekeeping' activities the system performs as it swaps from one task to another. In between

time-slices it reads characters from the keyboard and serial ports, blinks the cursor, checks whether or not there are 'slave blocks' that must be read and written by the file system - and switches between active tasks as priorities indicate.

The underlying algorithms in the scheduler are quite simple, but sometimes the interplay between tasks may cause weird and wonderful results. Multi-tasking is fascinating as well as useful!

QL RAM speed

In the April issue we tested a number of QL expansion memory boards, showing the very wide variation in speed between internal QL memory and add-on RAM. AJ Newell has tested his Inpholink QL expansion using the program we listed in April, and finds it has a bus bandwidth of just 867 KHz, rather than the 1295 KHz quoted in our table.

We already have evidence that some manufacturers change the speed of their memory boards from one production run to the next. It appears that this Inpholink unit is synchronised with internal memory access, which makes it much slower than it need be.

We'd be very interested to receive timings for other memory boards, and other Inpholink units. Please write in if your system does not match the results printed in issue 2.

I Spy

In next month's *QL Corner* we'll check out *Spy* - a fast multitasking editor from ARK Distribution. *Spy* is only 22K long and costs £14.95, yet it has quite a few unique features.

Spy can edit just about anything, from assembler and BASIC source to tasks, toolkits, code - even corrupted Psion data files! What's more, you can use it to edit and move data between as many files as you can fit in memory at one time.

For good measure, we'll compare *Spy* with the many other editing and word processing utilities for the QL. Don't miss next month's *QL Corner*, and if you know anyone else with a QL, tip them off about *Shopper* too!

Sid Martin and Timothy
Green are Sinclair aficionados