QL corner

by Sid Martin and Timothy Green

Step inside and see Ant's PC Emulator put through its paces and an adventure game for the QL that's a real hoot!

Ant PC Emulator V2

AT LAST we've received a copy of Ant Computing's *PC Emulator* for the QL, announced here in February. Since that report the market for QL PC emulators has gone wild, with a barrage of criticism directed at Ant's first release, accompanied by the launch of Digital Precision's rival product, *The Solution*, and rumours of a hardware-based emulator under development. Soon after the launch of its emulator, Ant licensed it to Schoen, mainly known for QL keyboards. That deal fell through, and Ant is selling the new version of the emulator directly, while Schoen awaits delivery of a third, hardware-based emulator, under development in Germany. That emulator is called Transformer and is expected to sell for around £100.

We've spent much of the last month testing version 2 of Ant's emulator, which costs just £14.95 – much less than its predecessor and competitors. The good news is that it works. The bad news is that it's slow, has some rough edges, and will not run all PC software.

We tested *PC Emulator* on two 640K QL systems, with CST and Sandy expansion units. Timings were performed on the CST system, which has relatively fast RAM; other QL configurations may be slower. We tried to run *PC Emulator* on our 1MbThor XVI, but it crashed the machine, displaying a register dump.

Ant PC Emulator

Menu 2.5

* data can be displayed in three formats:
  prior 1) directory format   (for interpreting directory data)
  next   2) hexadecimal format (for technical use)
         3) text format   (for word processing data)

* pressing the Enter key will switch from format to format
  with this help screen appearing between formats

* pressing Enter once more will move to the next format
* pressing Enter four more times will return to the prior format

* use the cursor keys to move through the data

* in the hexadecimal format only, you may
  * enter changes, in either hexadecimal or ASCII characters
  * use the Tab key to switch from the left side (hexadecimal)
    to the right side (ASCII characters)

Tue 20 Jul
20:23:37

Press Enter to continue; press Esc to return to the menu...

The Ant PC Emulator showing ragged box graphics when running Norton Utilities; note the QDos clock and SuperBasic 'RUN' command (to save the screen) at the edges of the printout

The emulator is supplied as a 40K task which loads in under five seconds. Then you supply details of your disk drives. *PC Emulator* supports two drives, A: and B:, each of which may be either 3.5" or 5.25" in diameter. The emulator will read and write 40-track PC disks on an 80 track drive, so you don't need a switchable drive.

If you only have one drive, MsDos lets you assign it to both drive A: and B:, printing a message and waiting for a key-press every time the disk needs to be changed. CP/M Plus on Amstrad machines works in much the same way.

Once the drive details are set up, you must put an MsDos or PCDos disk in drive A, so that the emulator can read the PC operating system. This is much slower than loading the QL task, because each PC instruction must be interpreted by calling a 68008 subroutine.

Calls to the operating system are intercepted and handled by QL code, but there's quite a lot of 'overhead' involved in making each call. The QL processor, ports and memory map are nothing like that of a PC, and the emulator has to work very hard to simulate all the PC's hardware.

PC emulation squanders some

of the QL's power, and shows up some of its weak points, like a relatively low bus bandwidth. The best features of the 68008 – like 32-bit register arithmetic and flexible addressing modes – are

not available on the 8088, so PC programs do not use them.

PC programmers might use several 8088 instructions to do something that the 68008 could do in one step – but the emulator must call a whole 68008 subroutine for each 8088 instruction. The emulator might try to speed things up by analysing whole groups of instructions, but the basic problem remains: it takes several steps to recognise a sequence which the PC can execute without pause.

Loading

The emulator loaded PCDos 3.3 in just over a minute – compared to 10 or 15 seconds on an Amstrad PC, or five seconds to load the same amount of code with the QL's native operating system, QDos. Similarly, Amstrad's MsDos 3.2 loaded in 58 seconds. Ant Software says the emulator works with version 3.1, 3.2 or 3.3 of IBM's PC Dos, or equivalent versions of MsDos, the 'compatible' system used in PC clones.

It won't run the latest version, MsDos 4, which was recently released and is still being debugged by Microsoft. We tried it on MsDos 2.3 for the old Apricot F1, but it wouldn't load; this did not come as much, sur-

Running PC Tools

Path=A:\NORTON*.*

UD	EXE	15896 ...A 3/01/87		UL	EXE	7456 ...A 3/01/87
SI	EXE	14750 ...A 3/01/87		FA	EXE	7296 ...A 3/01/87
FI	EXE	14668 ...A 3/01/87		BEEP	EXE	5324 ...A 3/01/87
QU	EXE	14554 ...A 3/01/87		NUDEMO	BAT	4912 ...A 3/01/87
FR	EXE	12610 ...A 3/01/87		SA	EXE	4586 ...A 3/01/87
LP	EXE	11616 ...A 3/01/87		FILEINFO	FI	2681 ...A 3/01/87
FILE1	BAT	11385 ...A 3/01/87		READ	ME	1222 ...A 3/01/87
WIPEFILE	EXE	10936 ...A 3/01/87		ASK	EXE	1184 ...A 3/01/87
NU	HLP	10897 ...A 3/01/87		TUTORIAL	NTS	914 ...A 3/01/87
WIPEDISK	EXE			MARY		618 ...A 3/01/87
FS	EXE	8536 ...A 3/01/87				
FF	EXE	7924 ...A 3/01/87				
LD	EXE	7652 ...A 3/01/87				

23 files LISTed =	186915 bytes.	23 files in sub-dir =	186915 bytes.
0 files SELECTed =	0 bytes.	Available on volume =	5120 bytes.

A screen from PC Tools V2.03, running under Ant's PC Emulator, printed with GPRINT_PRT on the QL... the odd 'colours' and box graphics are accurate

Continued from page 113

prise, as the old Apricot was not PC compatible. We tried to emulate Digital Research's alternative Dos-Plus operating system, but it locked up after writing a few spaces on the screen.

Once the emulator has started, you are presented with the standard MsDos prompt, A>. You can make this more informative with a command like PROMPT pg. The standard MsDos shell is pretty tacky, whether you run it on a PC or an emulator. The left arrow key deletes to the left, so you can't move the cursor inside the input line. Files can be 'secret' but you can still read their contents if you know their names.

There are well over 10,000 PC programs available; we tried as many as possible on the *P C Emulator*, aiming to give a general feel for the level of compatibility. Results may differ depending on PC program versions and configurations, so it's best not to assume that particular packages will work till you've tried them.

Ant says that it has tested the emulator with *MBasic, XTree , Rapid File, Perfect Writer, Perfect Filer* and *Perfect Calc*. Standard MsDos commands and simple disk utilities worked fine, including TYPE, DIR (/W), CLS, CD, MD, RD, RECOVER, CHKDSK, DISKFREE, MEMFREE, ARC, PATH, LABEL, DATE and TIME.

The emulator took 45 seconds to display a full directory of 24 MsDos files. FORMAT B: worked, even on a single drive system, but it was very slow indeed, taking around 260 seconds to format a 360K disk.

The emulator got in a mess when asked to format a system disk, with FORMAT/S. It failed to write the system files because it ended up thinking the disk in question had four sides! We had to create system disks by copying an original with *Diskcopy*, which took the best part of nine minutes to copy 360K to an unformatted disk.

Turbo Pascal ran, but hardly at 'turbo' speed – the compiler took over two minutes to compile a 322-line program into 3K of code and 1K of data. The Pascal editor was very slow, particularly when scrolling, but usable in small doses. *Mallard Basic* 'Run Only' version 1.26 seemed to work, but we were not able to try version 2 because it is tied in with *GEM*, which won't run under the emulator.

The *ZBasic* language 'interpiler' worked OK as long as the screen was cleared regularly. It managed 1,000 iterations of a floating-point IF loop in 54 seconds – or 86 seconds to go round a FOR loop 1,000 times. ZBasic floating point maths is very software-intensive, and integer loops

were much faster – the emulator managed 30,000 integer iterations in a relatively speedy 18 seconds.

We have not been able to run tests with Microsoft's *GWBasic*, which is said to work. The ancient *MSBasic* interpreter did not work properly and *Apricot GWBasic* failed to load, in the understandable absence of Apricot ROMs.

Version 3.10 of *Norton Utilities* (58K) loaded in around 40 seconds and ran OK – if very

slowly is OK. Other programs, like DS (directory sort), CHATT (CHange file ATTributes), VL and QU also worked. The Norton SI 'System Index' program ran, but didn't give much information as it rounded the speed index of the emulator down to '0.0 times that of a PC XT'!

Version 4 of *Norton* would not load, giving a spurious 'packed file is corrupt' message, Ant Computing says this is caused by the same obscure bug that clobbers FORMAT/S. The same message appeared with Amstrad's RPED.EXE.

PC Tools 2.03 was the only program that did not seem sluggish – some of the display colours were a bit odd, but everything was readable and the code ran at quite an acceptable speed. *WordPerfect V4* seemed to run reasonabley well, but was too slow for serious use.

We were less lucky with demo versions of *Supercalc 5* and *Shopper*'s *Volkswriter* freebie, both of which got stuck during initialisation. The same happened with TurboPower's *MAPMEM*.

Hardware support

Unlike DP's *Solution*, Ant's *PC Emulator* runs as a single task, so character output is handled by the same program as emulation. This means that entries appear on the screen at much the same rate as with a CP/M emulator – although processing is slower.

With *PC Emulator*, characters are displayed one at a time as the

PC program outputs them, whereas *Solution* uses a second task to write characters, so output may lag behind the cursor, with characters appearing intermittently in small groups.

The biggest fault of Ant's emulator is the lack of support for graphics. It can output ragged box graphics and graphic characters, like the PC's happy faces and card symbols, but attempts to display CGA graphics invariably crashed the emulator. Despite its slow output, *Solution* is much

---

PC Emulator Patch Program

```
100 COPY flp1_pc2,flp1_pc2a
110 OPEN #3,"flp1_pc2a"
120 SCROLL #3,18686,42
130 a$=INKEY$(#3) & INKEY$(#3)
140 IF a$<>CHR$(34) & CHR$(60)
150   PRINT #0;"Bad PC2 file."
160 ELSE
170   SCROLL #3,18686,42
180   PRINT #3;"$<";
190 END IF
200 CLOSE #3
210 STOP
```

---

better in this respect as it can cope with four-colour CGA graphics emulation in a 320-pixel wide QL window.

The first version of Ant's emulator could only cope with MDA text displays, but version 2 supports CGA text as well, with a resolution of 80 columns in 25 lines. Ant says that many bugs have been fixed since the first release, which was never sent out for review. "There have been some quite drastic changes", commented Ant programmer Guy Turley, "basically version 2 works!"

PC Emulator has trouble making sense of some CGA colours, which it converts into QL MODE 4 stipples. At times we found that text appeared in black on green and red stripes, or white on gold (actually a green and red chequerboard pattern). A couple of programs highlit the current option in black on black (!), so it was necessary to move the cursor up and down to see the text underneath.

Both *Mallard* and *ZBasic* left the last two characters at the right hand corner of the screen when scrolling. Ant blames this and the weird colours on the QL setup, which uses an Amstrad CPC monitor. Some programs needed CLS before loading, to ensure a tidy screen.

MsDos's internal clock worked, but ran slowly – at between 50 and 90 per cent of real time, depending on the amount of disk activity. The

times used to mark files are really only useful if you want to tell which of a group of files was created first.

PRINT, the MsDos spooling command, worked, albeit slowly. PRINT takes a list of files and sends them to the printer while you use the machine for other things. Spooling was sporadically directed to serial port 1. You can redirect this to a parallel printer with PAR_USE or a hardware protocol converter, but there's no obvious way for PC tasks to access QL devices other than the console, printer and disks.

The emulator multi-tasks happily under QDos, although it runs even more slowly if other QL tasks are busy, and hogs disk drives assigned for its use. Amstrad's PC RTC (Real Time Clock) code has no effect, but QL multi-tasking clocks run happily alongside the emulator.

You can't remove the emulator task safely once it is running, as it links itself into the QL's polled interrupt list. Standard commands to remove a job leave the interrupt connected, even if the code disappears. This is really a QDos fault, although it's a shame Ant does not provide a special keystroke to close the emulator down properly.

You must swap between tasks with Control V, rather than the adjacent QDos standard Control C, as MsDos uses Control C to break out of programs or commands. Some PC keys are not available on the QL keyboard, so the emulator recognises SHIFT ESC as a special code.

For instance, you type SHIFT ESC then F1 to get the effect of F6, which enters an MsDos end-of-file character. SHIFT ESC F5 gives F10. Similarly, SHIFT ESC and I stands for INS, with E for END, H for HOME, and up and down arrows to page up and down. A beep sounds after SHIFT ESC to warn that the next keystroke has a special meaning.

Commands and output are copied to SER1 if you type Control P – like Control B on a BBC Micro. Lines appear on the screen and printer, giving a hard copy of your dialogue with MsDos.

The QL ALTKEY extension works fine with the emulator, so you can swap out to SuperBasic, set up a sequence of characters to be associated with ALT and a letter, and type the sequence into the emulator with one keystroke. *SuperToolkit*'s ALT ENTER recalls previous lines, but adds a Control C at the end, so MsDos ignores the entry. This does not matter too much as you can still use MsDos's F3 key to recall the last line.

PC Emulator uses just under 110K of memory, including two 32K display areas. The supplied

version limited the PC program area to about 375K, regardless of the amount of memory on the QL. We wrote a tiny *SuperBasic* program (see the figure 'PC emulator patch program') to copy the task and patch it to use 5K less than the amount of free memory on the QL. It copies PC2 – the standard emulator – to PC2A on drive 1, reporting 'Bad PC2 file' if the code has changed since we found the patch. This boosts the maximum program area to around 490K on a 640K QL.

Verdict

Version 2 of Ant's *PC Emulator* works, but it is more of a toy than a serious tool. Dos commands work slowly, but at usable speed. Some PC programs worked without problems; others ran with bugs that could be worked around, but several programs wouldn't work at all, either because of incompatibilities, emulation bugs, or unacceptable sloth.

The instructions are just five and a half pages in a Quill document on the *PC Emulator* disk. You will probably need access to an MsDos manual if you're new to PCs, but Ant's documentation should be adequate if you're familiar with MsDos or PCDos.

Ant's emulator does not come with programs to transfer files from QL to PC format or vice versa, but these are widely available from other suppliers like PDQL, ABC, or Quanta's Public Domain library. We've had some fun trying out PC software on the QL, and feel that it was an inter-esting exercise – but we ended up with the impression that *PC Emulator* is best thought of as an adventure game for programmers and hobbyists.

If you want a PC, buy a real one. If you want to try out PC programs and freebie disks on your QL, £14.95 is a small price to pay – but don't expect PC performance from a software emulator, because you won't get it.

Sid and Tim are Sinclair aficionados

MacSporran's Lament

THE LATEST adventure from CGH Services is *MacSporran's Lament*, "a Highland adventure for the bonny QL", as the four-page instructions put it. You find yourself in the desolate Highlands, outside an eerie Scottish castle, armed only with a tin of Golden Syrup. Your rather cliched objective is to rescue a maiden from the castle, avoiding sulky sea lizards, goblins, Scotsmen and r tame spiders.

The game was written in Scotland by Dave, Anne and Kathy Watson, and it's packed with dialect and local references. The program explains 'I dinna ken' unrecognised words, and responses to 'HELP' include brief bursts of monotonic music and quotations from Robert Burns. We didn't find 'Wi datchie sesames, and names for nameless things', or 'They gang in stirks and come out asses' very helpful, but we assume they mean some-thing North of the Border.

During play, the top third of the screen shows simple filled line graphics. The graphics add a little atmosphere to the game, but they are crude – like the graphics on the Spectrum game *The Hobbit*, but not as well-drawn.

The game is a 64K task, which leaves around 20K free for *SuperBasic* on a standard 128K QL. It displays a cursor, so you can run it with EXEC and swap in and out of the game at any time with Control C. The version we tested came on microdrive, but we converted it to run from disk by patching the file name 'MDV1_ SPORRAN_ SAVE', near the end of the task.

Picture information is read from a 33K packed file. This technique works acceptably quickly, even from microdrive; the file soon finds its way into Slave Blocks on an expanded QL. Pictures are not redrawn when you return to a location

you have previously visited, unless you type LOOK.

Multi-coloured commands and responses appear in a four-teen line window at the bottom of the screen. The game vocabu-

Mode 4 title screen

lary runs to about 440 words, including synonyms and a rich variety of swear-words. You can see the vocabulary if you COPY the task file to the screen, but the game messages are encrypt-ed to save memory, so it's hard to cheat by examining the code.

The display works in MODE 4 and MODE 8 – you can even swap back and forth during play – but the first one or two charac-ters at the left hand of the screen may be lost on some TVs. Text uses an angular fount, rather than the normal QL chararac-ter-shapes, so mes-sages can be hard to read until you get used to the style of lettering.

The MODE 4 display uses 78 char-acters per line, in the QL's smallest char-acter size. Your entries appear in

green on black, with the normal QL line-editing features; you can move the cursor left and right in the entry, deleting either way. You confirm commands with Up arrow or Enter, while down arrow repeats the last entry.

MacSporran's Lament costs £8 on disk or microdrive. We have not had time to play it all the way through – PC emula-tion has taken a big toll this month – but *Lament* seems worth a look if you're keen on QL adventures and find the scenario appealing.

Automated adventures

MacSporran's Lam-ent was written with version 1.55 of *Adventure Creation Tool. ACT* is a utility rather like Incentive's *GAC* (*Graphic Adventure Creator*) or Gilsoft's *PAW* (*Professional Adventure Writer*) on other computers. It allows game designers to design text and graphic adventures without programming.

The designer supplies the

Mode 8 screen

utility with a list of messages, words and actions that can be used to solve the game, along with a map describing the links between locations in the game and optional graphics for each location.

The utility compiles this information into a task which plays the game, accepting input, recognising words, and display-ing text and pictures as neces-sary. The designer can concen-trate on the puzzles and player information without having to worry about nitty-gritty pro-gramming details.

As far as we know, *ACT* is the only graphic adventure utili-ty for the QL. The package was born a couple of years ago, as the *Adventure Programming Tool* from Ron Massey's Shadow Games, but it has since been taken over, renamed and enhanced by Digital Precision. *MacSporran's Lament* is the first *ACT* game we have received for review.

The only other QL adven-ture-writing tool we have encountered is Gilsoft's *The Quill*, which was adapted for the QL from an old Spectrum prod-uct. This works, although it has a few rough edges and can only handle text adventures. It was used to write the PD game *Ye Classical Adventure*, also avail-able from CGH Services.

The most common language for QL adventure-writ-ers is still compiled *SuperBasic*, which was used to write *Starplod* and *Tower of Valagon*. The advantage of Basic is that it gives the designer freedom to design an original user-interface for the game.
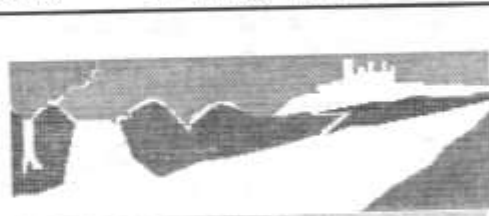
For instance, *Starplod*, reviewed here in March, uses joystick control with several text and graph-ics windows, plus an array of icons. You can't do that sort of thing without program-ming, because standard input and output rou-tines are predefined by adventure-writing utili-ties.