The Infinitely Profitable Program

Ξ **March 11th, 2008** | → 17 Comments | ∇ Coding, Geeky, Software, Technical |

A recent article on Slashdot about assembler games-programming on an Atari [Donkey Kong and Me] got me reminiscing about writing assembler apps in my own early days – and about the machines we had way back when.

Having earlier cut my teeth on a DEC PDP-8, the biggest kick I got was when CP/M came out. CP/M was initially a 'business operating system' – but – it was also a system that one could possibly afford to have at home, on a **P**ersonal **C**omputer – serious stuff for an up-and-coming nerd!

I was working for Tatung at the time, and as they built computers, I got to play with some pretty expensive kit – twin five and a quarter inch floppy drives and *everything*! I mostly worked on debugging the ERSO BIOS, and also worked on the Tatung Einstein computer; which used a compatible but beefed-up and expanded version of CP/M called Xtal DOS.

> The Einstein was a great little machine for a while [before MS-DOS machines really came into their own], as well as being able to run CP/M programs like VisiCalc and WordStar, it could also run great games programs as it had in-built sound and graphics capabilities [colour and sprites!]. And all for £499 [that's in 1984 remember] – pretty pricey for the day!

Peet's Utilities:

I wrote articles for Tatung's Einstein magazine, which I still have kicking around somewhere, and some commercial programs too. The most successful group of these was a suite of programs called, creatively enough, 'Peet's Utilities' [1986].

Written in Z80 Assembler [I didn't know about languages like C then], the utilities included an undelete, hex-editor, function-key programmer (a TSR; if the reader knows what such a thing is!), program auto-boot (like 'run on start-up' today), printer-controller, typewriter-emulator [yup I used **typewriter** there] and a whole host of other things. I sometimes wonder, if I'd have ported these to MS-DOS, I might have become Peter Norton [I assume Norton's Utilities made heaps, whereas I didn't]! I recently found the contract I signed for Peet's Utilities – I got the princely sum of £1 per copy.

For completeness, I should say how Peet's Utilities was written – the mechanics of it; as it's quite interesting, and indicative of the times.

It was written on a CP/M business machine – another Tatung machine as it happens [a TPC-2000 – the rightmost machine in this picture].

The left most machines in the picture are Tatung Einsteins.

The TPC-2000 was faster and had more memory than the Einstein – and so could run my macro-assembler more efficiently. It was also a **pure** CP/M machine – so, by writing it on that I could be sure to not use some special Einstein-only feature or subroutine (as I could then market it more widely).

After I built the programs on the TPC-2000 I ported them to an Einstein using a program/protocol called Kermit – where I'd test/debug it proper! Quite a code/build/test/debug cycle for the times!

GO.COM:

User feedback [on both Tatung's TPC-2000 and Einstein lines] repeatedly mentioned an irritation: that users often found they had to exit their current application [VisiCalc, WordStar, …] to perform simple disk operations, like finding a file on a floppy. It was a real and frustrating problem. For example, say they were running the popular WordStar word-processor and wanted to find an existing file for editing. Let's also say they're not sure which of a dozen floppy disks the file is on – they needed to use CP/M's DIR command to locate it. But, in order to use DIR they'd first have to exit WordStar. Of course, once they'd found their document they'd have to re-run WordStar, i.e., they had to load WordStar off a floppy again – a real pain, esp. as we're talking very slow floppy speeds here [for those that remember the sound – chunk chunk chunk is fairly onomatopoeic I think]!

To solve this problem I came up with the idea of GO.COM. **The most profitable program ever written**?

When a CP/M program loaded into memory, it was always located at the same address, at 0100h – the start of the so called 'Transient Program Area' [TPA]. CP/M's own mini-programs like DIR loaded, or were defined as resident subroutines elsewhere.

Well, it occurred to me that, as WordStar was still in memory [although the user had exited the program, the memory containing it (the TPA) was still intact] it would be rather useful to somehow re-execute the program in the TPA directly; rather than reloading it, i.e., why *reload it* from disk when it was already in memory? However, to do that you'd have to execute whatever code lay at the 0100h address once you'd finished using DIR etc. But, how to make that happen – you couldn't write a conventional program to do it, as it would have its 'jump to 0100h' commands/code loaded into the TPA itself – at 0100h – thus, if run, it would just run itself!

That's where GO.COM came in.

GO.COM contained no program bytes at all – it was entirely empty. However, because GO.COM was empty, but still a valid program file as far as CP/M was concerned (it had a directory entry and file-name ending with .com), the CP/M loader, the part of the OS whose job it is to pull programs off disk and slap them into the TPA, would still load it!

So, how does this help? Well, using the scenario above:

- the user exited WordStar
- the user ran DIR (or whatever else they needed) and at some future point would be ready to re-run Wordstar
- the user now 'loaded' and ran GO.COM
- the loader would *load* zero bytes of the GO.COM program off disk into the TPA – starting at address 0100h – and then jump to 0100h – to run the program it just loaded [GO.COM]!
- result – it simply re-ran whatever was in the TPA when the user last exited to DOS – instantly [WordStar in this example]!

So, GO.COM, which consisted of zero bytes of code – and sold for £5 a copy is, I figure, the most profitable program ever written (as any other program will return mathematically fewer £s per byte than GO.COM did)!

Is it really infinitely profitable? Well, in terms of what I made out of it obviously NO – I'm not infinitely wealthy. However, GO.COM could truthfully be a 'money for nothing' case. For example, priced per byte, it would look this way

Cost £5
_____ = ___ = ∞
Bytes 0

I actually had some funny (as in not so) phone-calls and letters over GO.COM [no email back then]: Some purchasers – who were obviously into computers – rang up Tatung to speak to me and to ask how – and why – I'd disguised the size of the program (DIR reported that is was zero bytes remember). When I told them that it actually WAS zero bytes long, some of them became a little annoyed! "How dare you charge me £5 for nothing!" I told them that I hadn't, I'd given them something useful and if nothing else, I charged them £5 for an implementation of the idea!

It's a pity CP/M's CCP didn't have more in-built commands; the existing ones were limited to a small set of useful disk-based commands:

* ERA erases specified files.
* DIR lists filenames in the directory.
* REN renames the specified file.
* SAVE saves memory contents in a file.
* TYPE types the contents of a file on the logged disk.

I later ported GO.COM to early versions of MS-DOS, i.e., before the EXE file format existed – and it worked just as well there too! Happy days!