

From software to fatware to nanosoftware



The first programmable electronic computer is considered the ENIAC, built as a general-purpose system (that is, capable of carrying out different tasks depending on the program provided). The operators of the system had to be extremely careful with the programs they developed, as an error resulted in a lot of lost time revising the perforated cards that provided the information. And the programs had to be very small, as its memory was small, it had 17,468 vacuum tubes capable of storing numbers: each digit used 36 tubes.

How must have the people who handled data from this system have felt? I would imagine that they felt like a biologist examining a microscope, inspecting each piece of data, each number, each instruction, so that it fitted into the tiny memory of the system.

The capacity and speed of computers started to increase rapidly and the first PC (the IBM 5150) could store 16,384 numbers in its transistor-based memory. The small company (at the time) Microsoft had developed a Basic language interpreter that only used four kilobytes and was included in the ROM of this microcomputer. Nowadays, it seems almost unbelievable. A programming language interpreter that fits in such a small space seems like an urban legend.

The progress of Information Technology has allowed faster microprocessors and more reliable storage devices to be developed and cheaper. Who could have imagined a computer like those around today in 1981, when the first IBM PC was presented? A microprocessor with a clock rate measured in gigahertz, disk and memory storage measured in gigabytes, and all of this for a quarter of what it cost then.

But, what have users gained from all of this? Does it take any less time to finish a spreadsheet on any current computer than a few years ago with systems designed for working with screens with bright green text? Not much. They are more attractive, with more fonts, many effects... but they are not much more effective. The large majority of users do not know what 80 percent of the functions of a word processing program do, which helps the office "experts" to come across as heroes when they teach someone how to put a word in bold without needing to take their fingers off the keyboard and using the mouse.

Software has grown and become more complex, as systems allowed it. This growth has resulted in higher resource usage: memory, disk, processor, graphics card, etc. Remember the famous dBaseIII, which went from occupying a couple of 5.4 inch floppy disks to dBase IV, which was distributed on eleven floppy disks!!!! Many were shocked by this waste of floppy disks and space. When this program was only used, in many cases, to register and query a simple database. How many users used the SQL query system embedded in dBase IV?

That's not to mention operating systems. MS-DOS 3.3, for example occupied two 360KB floppy disks, MS-DOS 6, four 1.44 MB floppy disks (16 times more),

From software to fatware to nanosoftware



Windows 95, 13 floppy disks with a special format that achieved a bit more space and then switched to CD-ROM to install it. Windows Vista is distributed on DVD, a media that can store 4.7GB, that is, over 13,000 floppy disks like those used to distribute MS-DOS 3.3.

What have been the repercussions of this growth? Many gadgets, three-dimensional graphic interfaces, photorealistic images, but at the cost of using resources as if there were no tomorrow.

And what about malware? How much space did Friday 13th virus use? Just 2KB of memory, and infected files grew by 1,813 bytes. And the Brontok.FT worm? More than a worm, it seemed like an anaconda, a python. It used 12 megabytes!

Everything grows: disks, memories, operating system functions... Can't applications be designed to shrink instead of growing? A digital watch has a larger information processing capacity than the Apollo XI and that reached the moon!

Yes, it is possible. On the mechanics side, the nanotechnologies field is being investigated so that computers are being built on an atomic scale. For now, they are only experiments, wheels in which the teeth are no more than atoms or tubes through which only a molecule can pass. Nanotechnology systems are starting to bud in the IT field. Programs that, in spite of the current trend to use more resources, more memory and more functions; are extremely light and rapid.

Nanoprograms can be designed for very specific functions, such as displaying a small clock on screen, or for a small but addictive game. Or even for incredibly complex functions, such as Panda Software's Nanoscan program.

Nanoscan is a system for seeking out active malware on computers, which is capable of finding hundreds of thousands of malicious programs, without needing to occupy megabytes or gigabytes on the system. Through an incredibly careful development system and with the objective of achieving maximum functions with minimum size, the market can finally forget about the tendency to fatten up software.

And how is this possible? Simple, forget that a system is going to be able to offer more resources without limit. Traditional programs have been developed thinking that they are going to be installed on a system full of different APIs, which are very useful and offer a wide range of functions, but must be loaded in memory to use them. And each program does the same. Therefore, resource usage grows out of control.

If a practically self-contained program is developed, without opting for fatware, and with truly investigative and evolutionary R&D, impressive results can be achieved in the software industry.

From software to fatware to nanosoftware



A new era is dawning, the nanosoftware era. Maybe, before very long, we will have to revert to 3.5 floppy disks to install a word processing program. Why not? It's just a matter of treating software development as a science, and not only as a set of interlinked files that eat up resources.

Fernando de la Cuadra
International Technical Editor
Panda Software (<http://www.pandasoftware.com>)
E-mail: Fdelacuadra@pandasoftware.com