



Any questions?

If you've got a PC problem, or think you could help out other readers, contact **Frank Leonhardt** who this month manages to upset IBM, Intel, Microsoft, most computer dealers and the entire academic establishment, in just two pages.



Triton is the codename for Intel's current Pentium support chipset. Older boards would have used sets like Neptune, which are not as efficient. The Endeavour is Intel's latest motherboard and it's rather good. It has built-in SoundBlaster-compatible hardware, and a version with integrated Diamond Stealth-like graphics is promised for spring '96. Endeavours can also be fitted with a 256Kb or 512Kb pipe-burst cache option which will allow the Pentium to run significantly faster than a standard cache. It has four ISA and PCI slots, although only seven cards in total can be fitted.

The Intel Zappa also has a Triton chipset with a standard cache option, one less PCI, one extra ISA slot and no integrated sound hardware. It is cheaper than the Endeavour.

Intel isn't the only company to make Pentium motherboards, however. The Triton chipset is available to third-party manufacturers whose products are noticeably cheaper than Intel's. For example, you can buy a non-Intel Triton board for 20 percent less than a Zappa or a board with pipe-burst cache for 13 percent less than an Endeavour. Not having a Triton chipset saves only about £15. The word on the street is that the non-Intel boards are just as fast, other things being equal.

As to the Windows 95 vs Windows NT question, I'm afraid you don't have the choice you were expecting. Windows 95 will run in 8Mb but prefers 16Mb, while Windows NT runs in 16Mb but prefers 32Mb. Windows NT doesn't have the compromised functionality from which Windows 95 suffers, but you have to pay

the price in RAM. Personally, I'd go for Windows NT if I weren't stuck with an 8Mb 486SX/25.

Real programmers do it themselves

As a student, I have been using computers to write essays for a few years now. I would really like to go a step further and learn to program properly (not in BBC Basic, as I did for Computer Studies GCSE). However, I can only find full-time degree or HND courses, or those aimed at the professional market costing upwards of £500.

Could you suggest somewhere that might do a course suited to learning programming at the hobby level? I did try a correspondence course once, but due to disorganisation and staff rudeness I didn't get very far. Any ideas?

Chris Liddell, York

There is no simple answer to this question — it depends a lot on what you have in mind by programming "properly". BBC Basic is actually a very capable programming language supporting block structuring, local data and indirection, and there is no reason why you can't apply proper programming techniques while using it. But for backward compatibility with earlier Basics, it does accept monolithic and unstructured code if that is what you choose to give it. I fear that a typical GCSE course does not have the time to explore the differences between the two coding styles to any useful level, as good style only comes into its own in larger projects.

If, by "proper programming", you have

Q My computer takes about two minutes to start up Windows 95. I don't think this is correct. I have a 486 DX250 VLB 128Kb cache with 8Mb RAM. I would be grateful for any advice. I enclose copies of my AUTOEXEC and CONFIG files.

Julie Musk

Windows 95 does take ages to start up on what Microsoft thinks of as a low-end machine. There is nothing wrong with your CONFIG.SYS or AUTOEXEC.BAT files, but you might be able to tune your BIOS setting to make the machine run faster. My solution to the slow startup was to put Windows 3.11 back onto my 486 with 8Mb RAM.

Spot the difference

With the release of Win95, I have decided to upgrade my PC. What are the differences between the Intel range of motherboards, e.g. Triton, Zappa, Endeavour? Also, it seems that to use Win95 efficiently you should use 16Mb of RAM. Should I go for Windows NT instead, as it's a true 32-bit OS?

Mark Antony Smith

SIMMs and the art of PC purchasing

The best way to wind up your friendly local PC dealer is to wander into his shop and try to order a machine to an exact specification. The chances are you will specify at least one item which has been discontinued or is incompatible with something else on your list. Back up your demands with an insistence that you read good reviews for all the bits you're after, and you'll have them ranting down the phone to me about how journalists in general cause them untold grief.

They do have a point. Just because a particular video board is the fastest in a group test, it does not mean that it is compatible with all motherboards past, present and future. An obscure sound board might sound great, but if there are no Windows 95 drives for it and the manufacturer has gone out of business, then that's that.

In the circumstances, you have two choices: you can either be guided by the experts, and if they are building several machines a week they probably have some idea about what works and what doesn't; or you can proceed by trial and error yourself. A dealer isn't going to be interested in building you a Frankenstein's monster because he suspects, quite rightly, that you'll be back a day later complaining that it doesn't work and probably blaming it all on him.

If the dealer is friendly, he may supply you with the bits on the basis that he'll exchange what doesn't work.

However, from time to time a few things crop up which you should be asking about. This month it's SIMMs. A Pentium processor board normally has four SIMM slots for memory. These slots have to be filled in pairs. SIMMs come in 2Mb, 4Mb, 8Mb, 16Mb and 32Mb sizes at present, but 2Mb SIMMs cost rather a lot per megabyte. The 8Mb SIMMs have always been a bit cheaper per megabyte than the 4Mb ones.

If you ordered a machine with 8Mb in it you would probably receive two 4Mb SIMMs, leaving two slots free. A 16Mb machine would have two 8Mb SIMMs and two free slots for expansion. However, 8Mb SIMMs have just gone up in price (not a lot) but it is now cheaper for dealers to fit four 4Mb SIMMs to get the 16Mb, which leaves you with no free slots for expansion. This is one question you really should raise with your dealer.

something in mind which would be commercially useful, then BBC Basic does not impress when entered on a CV unless accompanied by something mainstream. Real programmers use Assembler, or these days C and C++. However, an awful lot of application programming is carried out in Visual Basic or Delphi, and database application languages like Oracle and dBase. Although such packages are used for large projects, they are not, in my opinion, particularly well suited for the purpose as they don't encourage, or even allow for, a maintainable programming style. Such languages are popular because you do not need highly skilled programmers to get results. They do, however, yield rapid results so they are a good choice for trivial applications and quick hacks.

In the mainframe world, languages like Cobol, PL1 and RPG still rule the roost and show little sign of being displaced.

There is a tendency for degree-type courses to concentrate on the academically pure languages like Pascal, Modula, Simula and a truckload of experimental application-specific systems of which no-one else has even heard. There are probably good reasons for doing this, but the fact remains that degree students generally require a year

or three after graduation to learn C properly before they can earn their keep in the real world.

So what should you learn? Visual Basic is nice and easy and does have a commercial application. C is the obvious choice if you want to get at the inner workings of the system, though not for seekers of instant algorithmic gratification.

There are plenty of books, videos and courses available for teaching yourself either of these languages. Books about C, in particular, tend to go into the art of program design as well as teaching the language. I can personally recommend "The C++ Programming Language" by Bjarne Stroustrup, as it has plenty of examples on how to use the language rather than just explaining how it works. It isn't intended for complete beginners, however. "The Design of the Unix Operating System" by Maurice J Bach should be considered required reading as well. It isn't a programming tutorial but explains how Unix was designed and written, with just the right level of detail necessary to understand the techniques used. Even if you never program Unix, the algorithms described can be applied to many types of programming problem.

By reading and practising, there is no reason why you can't learn to program to

any level you choose — many microcomputer programmers are entirely self-taught. Doing a course wouldn't hurt either because of the group support available while trying to get your mind around a tricky concept. Many local councils run evening classes in computer programming and these are probably worth a look. The course content might only be to GCSE level, but the access to like-minded individuals and knowledgeable tutors could be much more important to you.

There are local computer clubs and societies too, which you might consider joining. Some are obviously better than others, so it's worth looking in on more than one.

Give us a C

I have just bought a copy of Borland Turbo Assembler for the PC. Are there any good books to help me learn? I have been programming in C for a number of years now and need to write some fast routines

Paul Taggart

Osborne/McGraw-Hill publish a good selection of reference books for microprocessors and "The 8086 Book" by Rector and Alexy starts from the ground up. "Mastering Turbo Assembler" by Tom Swan (published by Hayden) covers the 80x86 series up to the 386, and also has chapters on working with C and Pascal alongside Assembler.

The extra 486/Pentium/PP6 instructions aren't necessary for a tutorial and once you've cracked programming the 386, all you will need is the latest Intel data book for the relatively minor additions.

PCW Contacts

Frank Leonhardt is an independent computer boffin who can sometimes be contacted on **0181 429 3047** or via email as **frank@dircon.co.uk** or **leo2@cix.clink.co.uk**. Letters may be sent to PCW at VNU House, 32-34 Broadwick Street, London W1A 2HG, but individual replies are not normally possible. Please do not ask about cover disks or CD-ROMs.

The C++ Programming Language:
Bjarne Stroustrup,
ISBN 0-201-12078-X
Addison-Wesley 01734 794000
The Design of the Unix Operating System. Maurice J Bach,
ISBN 0-13-201757-1
Prentice-Hall 01442 881900
Intel 01793 403000
Simply Computers 0181 523 4020

