Once upon a time... [You can't start a book that way!]

[The wonderful thing about a word processor is that you can put random passages into it and sort them out later.]

Hypertext jumps

I would like to tell you about an invention that pre-dates HTML. It's called hypertext and was an Apple based database of connected text. It allowed you to link the text phrases or thoughts from inside your writing. Much like the human brain functions. When we read passages from a book our brains take certain phrases and 'jump' to other related thoughts or directions. I would like to place links at various places to related thoughts or points of interest. In the 'old days' these would be called footnotes and would require a magnifying glass to read them.

NOTES/Contents

1. **"Understand This"** - The 'Barefoot' training manual
2. Where and when did it (computers) all start?
3. It's NOT impossible and it can be done with [insert item here] - when asked by someone whether [xxx] can be done.
4. Of Mice and (wo)men
5. Of Cats and keyboards or monitors
6. On practical uses for the computer system
7. On specific South African problems with computers.
8. What paperless office? [Can you take a computer to bed to read?]
9. The over the top "User friendly" system
10. The use of humour in training & "Sense of humour failure"
11. It didn't do that before!
12. [Computers can do strange and puzzling things but they can't tell jokes!]
13. Starting right with a 'reliable' computer
14. Booting from a network. Thin / Thick / medium-sized clients.
15. Will the real owner of the English language please stand up?
16. Data Catalogue NOT search engine - the new/real "Killer Application"

History

Many years ago, I was the proud owner of a small microcomputer system called a Sinclair ZX81. I had bought this microcomputer in the UK and had 'played' with it many nights into the small hours. I was young and didn't know what I was getting into. Having brought it back to South Africa, I told everyone I came across about this amazing piece of technology. Some were impressed and some were downright reactionary. I had been a "HAM" (Radio Amateur) for some time and when I told all and sundry on the 'Joburg' repeater that this micro could do radio teletype. Well I was told that it was impossible and I was out of my mind or on drugs. How was I going to make it work, as it only had 1 kilobyte (1024) of memory and no serial data input or output?

The challenge was down. (In the olden days knights would challenge each other to duels by taking a metal studded glove and smacking the other across the face with it. Not quite what I felt like but close! I suppose the knight that could get up of the floor after that would accept the challenge of the 'throwing down of the gauntlet') About three months later I demonstrated a ZX81 to my friends in the radio club printing a taped radio-teletype message onto the TV screen. Those three months were an incredible learning experience for me.

During those months I learnt how to connect peripheral devices to the main microcomputer. These small systems did not have the serial or parallel ports or disk drives that the later computer systems had installed. Mind you the original IBM PC didn't have these either! I learnt about the architecture of the microprocessor, the busses and memory. I also learnt to program in machine code not assembler straight away. I carefully wrote my test programs in an exercise book and inserted them into a called routine 'poked' into memory by the built-in BASIC interpreter. I learnt about "endless loops" in a most efficient manner as my loops would take microseconds to process! [Unlike the one in Windows 95 that takes 49.7 days to complete. Most users, of course, couldn't manage to run their Windows 95 machines more than 24 hours at a time, preferring to switch off and go home at five o'clock.]

The long road ahead to [insert your destination here]

During the years following the ZX81, I produced Z80 based systems for time clock applications and data capture. When I got one of the first IBM PC XT's, I was quietly overjoyed as I was now going to be programming and designing hardware based on a 16 bit microprocessor that was going to be much faster than the 8 bit Z80. Imagine my disappointment when I discovered that the CP/M machines could do things faster than the IBM PC. However the compilers for the IBM PC were arriving and I could write in BASIC and translate these programs to assembler automatically. The translated program ran very fast and could keep up (just) with a 300 Baud serial data link! I very quickly learnt the assembler for the 8088/8086. This relatively new microprocessor wasn't as straightforward as the Z80. It was a complex design with a lot more instructions than the Z80. It also had some wrinkles with the hardware interface as well. It was definitely not going to be substitute for the Z80 in small micro systems. I acquired a 'cross assembler' for the Z80 that ran on MS-DOS. [I never found out what it was cross about!]

Later processors had many more pin connections that smoothed these wrinkles and made the use of the micro in a control type application quite reasonable. E.g.: the Hubble Space Telescope with an 80386.

As I was taught as an electronics engineer, I could pick up a circuit diagram very quickly. Designing and developing was in my blood as my work had been up until the middle 1980's. Designing a microprocessor system was easy, much easier than designing a special VHF/UHF receiver, and was a matter of putting the 'building blocks' together. [BTW - keeping the RFI (Radio Frequency Interference) out of a receiver generated by a microprocessor is quite another story]

How I got over my 'shyness'

Much later on I was persuaded to train support personnel in the black arts of the computer systems. This was not easy, as many had no experience of technology or even electricity. It seemed that even basic science in schools had been neglected or ignored. Even a basic understanding of the computer could help these unfortunate persons solve the day-to-day problems of IT support. [At this time the number of support persons per users was reckoned to be 1 to 12, most of my trainees were at 1 to 200 or higher!] Some had been told that IT support was a glamorous and well-paid field of employment. They quickly learnt that their best future was elsewhere. Many defected to other countries. The ones that persisted either grew skins of leather or used ear muffs (later iPods) to ignore the whining [complaints] of the computer 'users'.

I attempted to teach these poor suckers [more about 'blowers' later] about the practical side of diagnosing computer problems. I gave them my diagnostic programs and demonstrated the use of the operating systems programs to enhance and speed up their working. I tried to teach the use of their mind in solving the problem rather than the 'parrot fashion' or diagnostic chart method. This last was totally out of the ballpark when Windows 3.0 came along as the possible combinations of hardware and software issues (bugs!) were nearly infinite in number. Many took my BASIC programs to heart and used variations later on to automate support tasks. The choice of BASIC proved to be quite a good one, as all the operating systems up to XP, have a 'BASIC' provided with the system.

NOTE: BASIC has been around a long time and will continue to be around for much longer. Public Domain and Shareware BASIC's and compilers abound. In Windows its been mutated into Visual Basic, which wasn't free or cheap. But later operating systems have VB Script, which has most of the essentials of BASIC and can talk to the system using the Objects provided by the system. Visual Basic has now departed for another realm in the form (pun intended!) of VB.NET. VBScript and scripting in general remain part of the supplied operating system software.

Why too k?

In 'Bowling for Columbine', Michael Moore says that Americans in general were sure that civilization was about to cease when the year 2000 arrived. He said it was part of the paranoia of the American people to be scared of things and other peoples. He said that Y2K was nothing to be concerned about in retrospect. I don't think he could have been one of the thousands attempting to use certain ATM's that had 'died' that day.

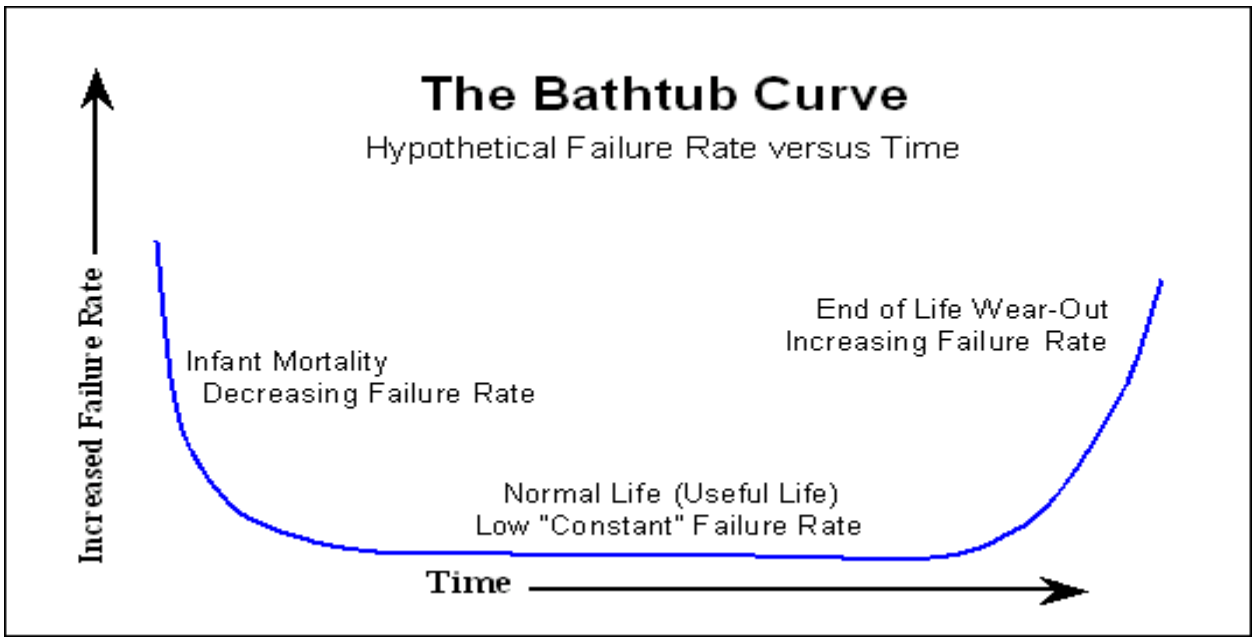
In planning for the future we have to take into account the possible failure of the system to cope with all sorts of issues.

Fail Safe?

Some years ago, I worked for a Railway Signalling company. This company was an agent for an overseas company, which had designed electronic equipment for the railways to ensure the safety of people travelling on, or working on the railways. The equipment would always "Fail Safe", so that signals [those signal lights on the railways - NOT Robots!] would always switch to red in the event of equipment failing. This company had just designed a microprocessor-based system for the railways overseas, which would fail-safe. An interesting aspect of this was the triple processor based system, that in the event of one of the three microprocessors failing or "going off the rails", the two 'sane' processors would blow the fuse of the faulty processor.

Hardware Reliability

Statistically it is possible to produce computers that are extremely reliable with a very low probability of hardware failure. By the use of the “Bathtub Curve” seen in any book on electronic equipment reliability, you can “soak test” equipment such that the item is delivered to the customer at point on the curve where it will be the most reliable.



The Dark Side of Computers

Nuisance stuff

It would seem that everyone else has your email address. Judging by the 'spam' you get. The term comes from a Monty Python sketch on an LP. [You may not know what an LP is, so here is a link to an explanation.]

If these perpetrators could be traced and nailed to wood, it would stop spam dead in its tracks. Unfortunately the nature of the Internet does not allow for the easy tracking down of the senders of these spurious emails. The only real way to stop spam is to ignore it. If it doesn't get to you, you won't be worried by it.

Most email clients can be 'programmed' to ignore emails from email addresses that don't exist in your address book. Unfortunately most don't delete them at the mail server, preferring to download them to your PC before deleting them. This is not the way to do it but it will cut down a lot of your distraction. If this became onerous to the ISP's because of mail server slowdown, they would also act to stop incoming spam. As it is at present, ISP's do nothing to protect their users from spam apart from making token attempts to install server based screening agents. These don't work 100% and can lead to incidents themselves as users claim that emails are going missing and it is the ISP's fault.

Email is NOT 100% reliable anyway and is totally insecure. It should never be used for confidential information transfer. I understand that the "inventor" of email is working on an identity system to provide for secure/private email. If you are considering sending confidential or private information via email, make use of Pretty Good Privacy. The product has proven itself as secure and a sufficient barrier to casual crackers. A sufficiently resourced government department or organised crime unit won't take too long to crack it. But you aren't too worried about them are you?

“Zombie” computers
[Yes they do exist!]

Quote: “Analysis of the net addresses where the e-mail messages originated showed that more than 100,000 hijacked home computers spread across 119 nations had been used to despatch the junk mail”.

Nowadays “BotNet” is used to describe the PC's that have been hijacked for use by the black hats for spamming. There have recently been cases of these “Bots” being counted in the millions!

Disaster Area

"Disaster Recovery" has a lovely tone to it. It describes the state of some organisations after some catastrophic incident has occurred. In some companies it means Monday morning. This is usually when a computer robbery is discovered. The realisation that the data on all the PC's is lost forever is usually followed by management going through a Biblical process, known as "A wailing and gnashing of teeth". [By the way - It's quite possible to have your computer 'call you' from its new home and announce to all and sundry that it has been stolen!]

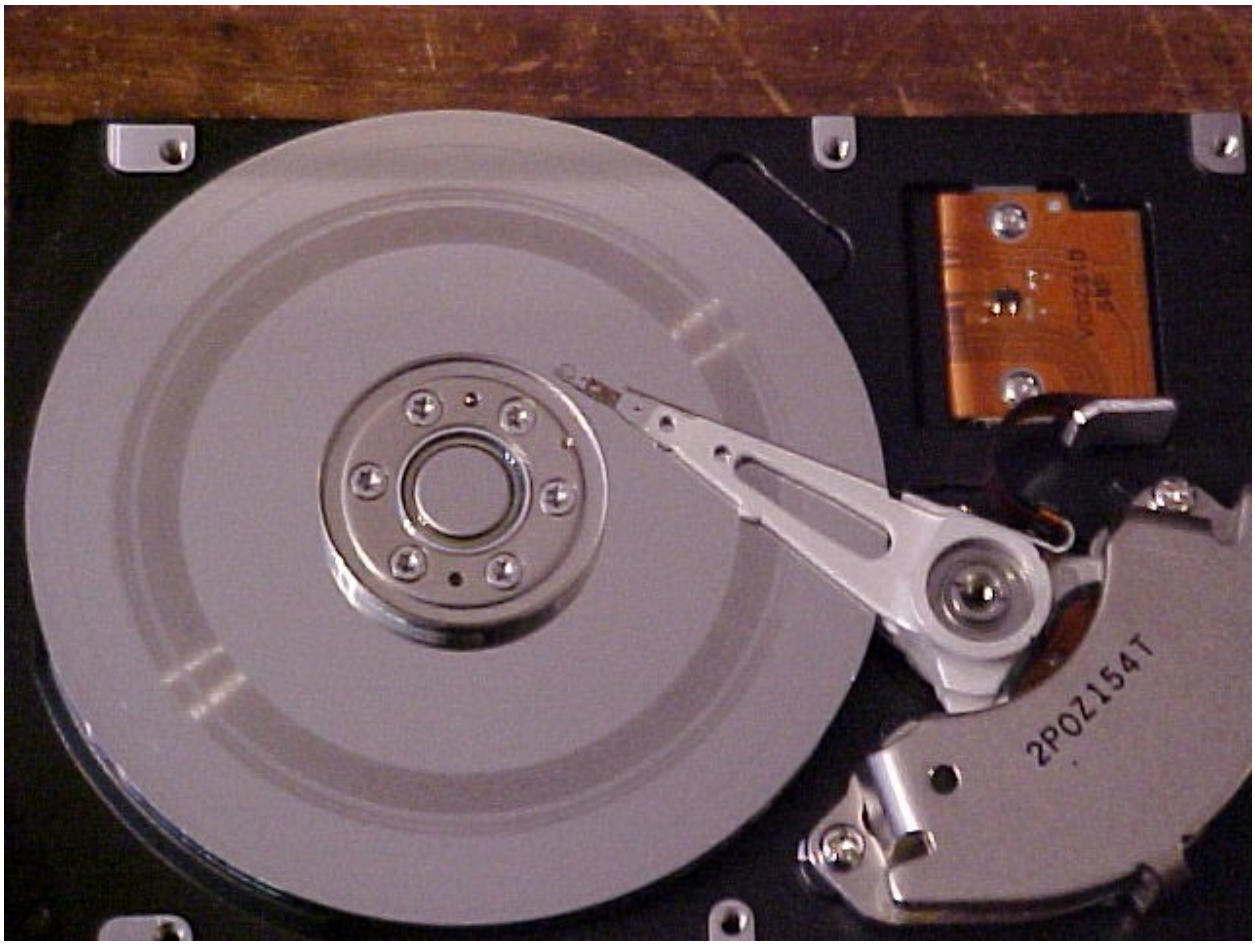
If its not been stolen, it has been struck by lightning via the phone line. Or mains power has been switched on and off rapidly to the building/home/suburb and the power supply in the computer has 'given up' or the hard disk has 'bounced the heads'.

Other largely physical incidents have been noted, such as kicking the PC when it didn't start up fast enough. We don't have too many earth tremors or earthquakes in South Africa, as the region is geologically stable. In LA an earthquake was responsible for a vast number of hard disk failures. Bomb blasts have also been responsible for hard disk crashes with one client telling me of an incident in London, where they couldn't get to the backup tapes, because the building had been structurally condemned.

What to do after your hard disk has crashed [First steps in data recovery]

1 Assess the damage. Include the cost of replacing the data.
2 Can the disk be accessed by the bios? Try auto-detecting the drive.
3 Can the drive be read by another pc as a slave drive?
4 Have you another hard disk of the same model and make? Try swapping the electronics card from the known good drive onto the broken drive. This may require that you get a special screwdriver attachment. Exercise care and do not drop the drive. If you can access the drive without error messages, copy data across to another drive.
5 If the drive persists in being stubbornly dead, consult a data recovery expert or company. The cost of recovery may be more than the value of the data.

Don’t open the drive like this until all hope of data recovery has been exhausted.



Damaged Disk

Virus, Worms and other malware

Virii (plural of virus) of all sorts abound. Its only just dawned on MS (see Windows XP SP2) that you can mark data as non-executable. Intel designed their processors to cope with this, as has AMD. So why did MS take so long to realise this?

There is quite a history of this, so expect a deep and thorough examination.

Processor architecture

Write-only memory - should be read-only but it makes you think doesn't it?

Memory protection - why UNIX is MUCH BETTER than Windows

Programs or applications in Unix cannot 'touch' another program or user's memory. The kernel just won't allow it. The isolation in windows is pathetic and some programs actually re-write blocks of memory or disk without being checked.

Contributors to my computer life: -

- Douglas Adams - the author of 'The Hitchhikers guide to the Galaxy'
- Bill Gates - why not? I have read several of his books. They are all good.
- The Jargon File - see the Internet http://catb.org/~esr/jargon/
- My students (Who do you think gave me some of the funniest anecdotes?)

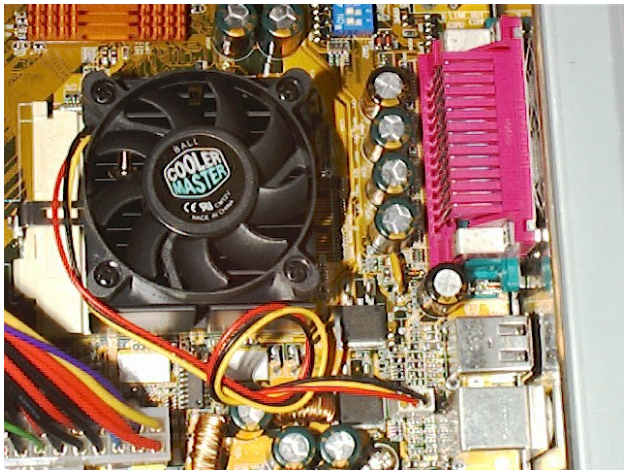
BOOKS [lots of them!]

8080/Z80 Assembly Language - Alan R. Miller ISBN 0-471-08124-8
Z80 Assembly Language Programming - Lance A. Leventhal

More to come...

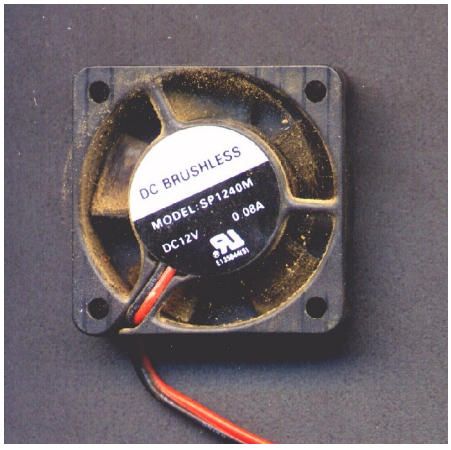
CPU Fans

Ever since Intel and others decided to internally multiply the clock for the processor, it has been necessary to keep the CPU cool. Heat is generated in the silicon on rising and falling edges of a square wave. The faster you drive the chip the closer these edges get together and the more heat is generated.



Note the screw stuck in the fan!

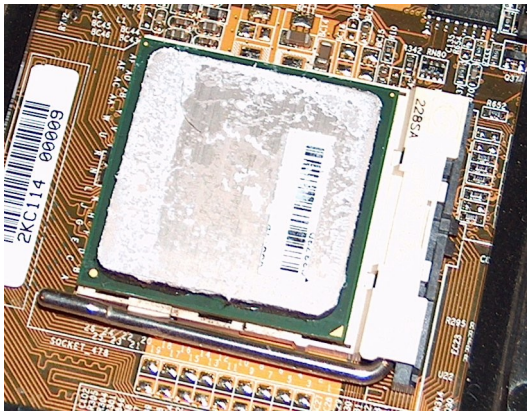
This CPU got so hot the plastic label melted.



Dirty fans cost CPU's

Further problems with fans and cpu's

When people are not trained in hardware assembly, look what happens...



The processor overheated because the heat transfer junction was insulated by the bar-code label. This is one of several pictures of different cpu's that either overheated or 'died' as a result of this practice.

Cross Assemblers

Assemblers that run on one computer that generates code for a different processor. Nowadays there are cross-compilers and cross-assemblers galore. With the recent open-source movement comes all the software tools that we used to pay for with serious money. Most work on both major platforms, Linux and MS Windows. The big jump in the last few years has been to the 'managed' variety of object-oriented code. Unfortunately this has dropped the classic Visual Basic that I was very conversant with.

BASIC

[Beginner's All-purpose Symbolic Instruction Code]
[The programming language of choice.]

BASIC Beginnings

BASIC, which is undoubtedly the most popular computer language, is bundled with virtually every microcomputer sold. It was developed in 1964 by two Dartmouth professors, John Kemeny and Thomas Kurtz. They noticed that most decision makers in business and government are non-scientists and that their decisions dramatically affect the worlds of work and public affairs. They wondered how sensible decisions about computing and its use could be made by people essentially ignorant of the subject. They agreed that you could no more learn computing by listening only to lectures and having no computer than you can learn to drive a car by using only a manual and having no car.

Two problems needed to be overcome. First was the inaccessibility of computers, which ran in batch mode from punch cards; second was the need for a language that was easier to learn than the highly mathematical FORTRAN. BASIC, with its accompanying time-share system, solved both problems. It operated then, as now, as a desktop calculator, where you could enter

```
PRINT LOG(25)
```

and get an immediate answer, or as a programming language, with which you could write and save complete programs. Early BASIC was interpreted; now, newer versions can be compiled as well.

BASIC's detractors were powerful and vociferous, with perhaps the most damaging charge coming in 1975 from Edsger Dijkstra: "It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: As potential programmers, they are mentally mutilated beyond the hope of regeneration." It was the much-maligned GOTO as the principal control structure that led to such strong condemnation. And I must agree that such was my experience during the late 1970s. Converting a gung-ho high school BASIC hacker to structured programming when GOTO's work just fine is hard indeed.

Dartmouth BASIC was never copyrighted or standardised, so different versions that took advantage of various microcomputer features became available from almost every software developer. These included BASEX, Integer BASIC, MITS BASIC, Tiny BASIC, SWTP BASIC, RM BASIC, BAZIC, BASIC-09, Better BASIC, Professional BASIC, Macintosh BASIC, Microsoft BASIC, Applesoft BASIC, CBASIC, SBASIC, BASICA, GWBASIC, TBASIC, and a standardised ANSI Minimal BASIC in 1978 (so minimal that it was almost totally ignored by the industry).

What's New?

Kemeny and Kurtz found this proliferation appalling and dubbed these horrible dialects of such a beautiful language "Street BASIC."

They thus formed a corporation in the early 1980s and developed True BASIC, a version that meets the ANSI and ISO standard. A True BASIC program will run on any machine that supports an ANSI BASIC system. Its compiler has two windows: an editing window, where you can write and edit programs, and a history (or command) window, where you can run programs and interact with them.

Programming language concepts have developed since the 1970s, along with new languages that implement the new concepts as well as take advantage of improvements in hardware. BASIC has changed also, and the two leading implementations are True BASIC and QuickBasic from Microsoft.

When BASIC was first developed, notions of structured programming were mostly in the paper-and-pencil stage. Structured code reflects in appearance a program's organisation. Notions of structure include blocks, where data can be localised, and data structures, such as arrays, records, and lists, to name only a few. The new BASICs include structured features such as IF THEN ELSEIF ... ELSE ... END IF, FOR ... NEXT, DO WHILE LOOP, DO ... LOOP UNTIL, and various CASE statements. They also support graphics, matrix-handling functions, and internal and external functions and procedures. Line numbers are optional, and you can insert or delete them using the menu bar.

Another concept implemented in modem high-level languages is modularisation, where data and related procedures can be bundled together, with some features kept hidden from a user. True BASIC has added this feature in the form shown in listing 1. QuickBasic's version calls global variables COMMON, local variables STATIC, and shared variables SHARED.

You can save modules and collections of BASIC code in files and include them in other programs with the INCLUDE statement, or you can store procedures and functions in a library, from which you can bring them into a program with the CALL statement. You can save and distribute compiled code in EXE files, so developers can sell their software without source code.

But True BASIC has been kept lean in keeping with its designers' philosophy of simplicity. This is not so with Microsoft's candidates, QuickBasic and Visual Basic for Windows. Microsoft's president, Bill Gates, wants QuickBasic to replace Pascal as the language of choice for high schools. Thus it includes user-defined data types in the form of records. It makes no attempt to adhere to the ANSI/ISO standard but takes full advantage of the DOS environment. Visual Basic, an attempt to make Windows programming easy, includes icons that can be designed into the user interface as well as used while writing BASIC code.

As an example of the different capabilities of True BASIC and QuickBasic, the programs in listing 2 (True BASIC) and listing 3 (QuickBasic) sort a list of student names in descending order according to their grade-point average. There are several things to notice in these two versions for sorting an array of "pointers" to fixed array elements. First is that neither version has true pointers that contain memory addresses. An array, called List or Rank, keeps track of the array location instead. Second is the existence of record types in QuickBasic. Third is the built-in Swap function in QuickBasic, which has to be programmed in True BASIC.

You may wonder why I changed the name of the True BASIC List array to Rank in the QuickBasic version. Well, it turns out that List is a keyword in QuickBasic that lets you reassign function keys in the DOS system. Lots of things like that show up. But if you're wedded to DOS, you'll learn all these keywords quickly enough.

There are advantages to strong typing and variable declarations, as in Pascal and Ada. It is very hard for inexperienced programmers to detect errors in BASIC. Maybe BASIC is for experienced programmers like Art Rainirez (whom I will discuss in a moment) or those who want little more than a fancy calculator, and the strongly typed languages are better for beginning programmers.

Who Uses It?

You can customise operating-system commands with macros written in BASIC. Many laboratory instruments also have an understanding of BASIC. Art Ramirez, a low-temperature physicist at AT&T's Bell Laboratories, measures magnetic and thermal effects occurring during experiments on superconductivity. He writes controllers for various devices such as voltmeters, using either GWBASIC (QuickBasic's predecessor) or TBASIC (a version of True BASIC from a company called TransEra). His instruments "understand" BASIC, he finds it quick and easy to use, and he feels no need for anything fancier. The True BASIC company sponsors a group that distributes shareware among engineers and others who might be interested in scientific applications.

Thousands of commercial programs have been written in BASIC by people with good ideas but little programming experience. What do you do when a program needs upgrading but is written in a version of BASIC that's no longer supported by its distributor? Ah, there's the rub-in any non-standardised language, not just BASIC. The hodgepodge of BASICs appears to have settled down to two major contenders: the standardised, simple True BASIC and QuickBasic, which has many more features. Either one has enough structures to program some pretty serious applications.

[Stolen from BYTE magazine (which has since gone belly up and resurfaced as a web site.) - by Doris Appleby]

INTEL PROCESSORS/CPU's

The 8088/8086

..
..
..

IBM PC

The IBM Personal Computer

Probably the most sought after microprocessor-based computer of the twentieth century. Its "clones" numbered in their millions. Its still the base model for personal computers even though the machines produced today don't always look like the original.

[]

The original had 64k of memory, maybe one floppy drive or a cassette port (at the back), no hard disk and only a black & white (well green and black actually) display adapter.

:user: /n./ [from the Jargon File]
1. Someone doing `real work' with the computer, using it as a means rather than an end. Someone who pays to use a computer. See {real user}.
2. A programmer who will believe anything you tell him. One who asks silly questions. [GLS observes: This is slightly unfair. It is true that users ask questions (of necessity). Sometimes they are thoughtful or deep. Very often they are annoying or downright stupid, apparently because the user failed to think for two seconds or look in the documentation before bothering the maintainer.] See {luser}.
3. Someone who uses a program from the outside, however skilfully, without getting into the internals of the program. One who reports bugs instead of just going ahead and fixing them.

The general theory behind this term is that there are two classes of people who work with a program: there are implementers (hackers) and {luser}s. The users are looked down on by hackers to some extent because they don't understand the full ramifications of the system in all its glory. (The few users who do are known as `real winners'.) The term is a relative one: a skilled hacker may be a user with respect to some program he himself does not hack. A LISP hacker might be one who maintains LISP or one who uses LISP (but with the skill of a hacker). A LISP user is one who uses LISP, whether skilfully or not. Thus there is some overlap between the two terms; the subtle distinctions must be resolved by context.

PC
Personal Computer - for a single person. Computers used to be people! Computers used to be multi-user monstrosities that lived in air-conditioned environments.

Does NOT mean “Politically Correct”.

Radio Amateur

What is a HAM?

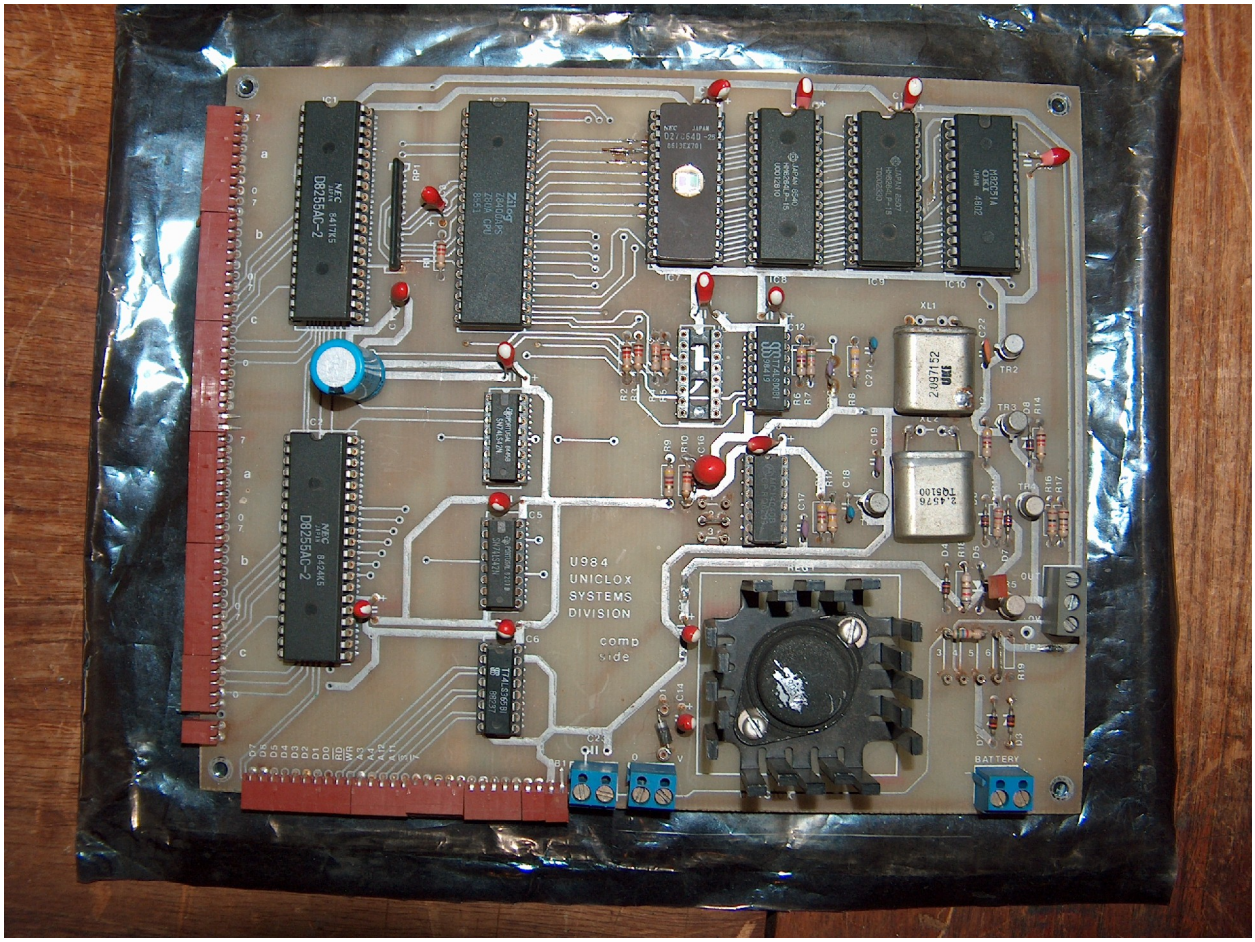
The Sinclair ZX81

Thank you Sir Clive!

Windows 3.0

ZILOG PROCESSORS/CPU's

Z80



http://www.zilog.com

This "chip" has been responsible for more "gadgets" and hardware innovation than any other. It was the second microprocessor to take a major market share pushing Intel off the lead position for quite a while. Mind you it was developed by 17 of Intel's former employees.

It was used in everything from toasters to GameBoy's, from Hayes Modems to door locks. The ever-loved TRS80 and Colour Genie processed using it. It ran CP/M [Control Program for Microcomputers].

The Long Playing record - LP

The LP resulted from the arrival of vinyl in the 1940s. Using vinyl as oppose to shellac it became possible to press records that played at 33 and a third RPM and could hold over 20 minutes of music on each side.

The LP format changed forever the way music was sold to the public, as it was now possible for artists to release collections of music rather than single songs. LPs quickly became a standard format for and remained so until the introduction of CDs 40 years later.

For a better explanation see:
http://www.kcmetro.cc.mo.us/pennvalley/Biology/lewis/crosby/lphist.htm