WRITING FOR THE SCREEN

Many home programmers dream of writing a best seller. But they rarely realise what they are up against. Professional software companies have enormous resources behind them to help produce their chart-toppers. One UK software house has over £250,000 worth of minicomputers dedicated to creating packages for home micros.

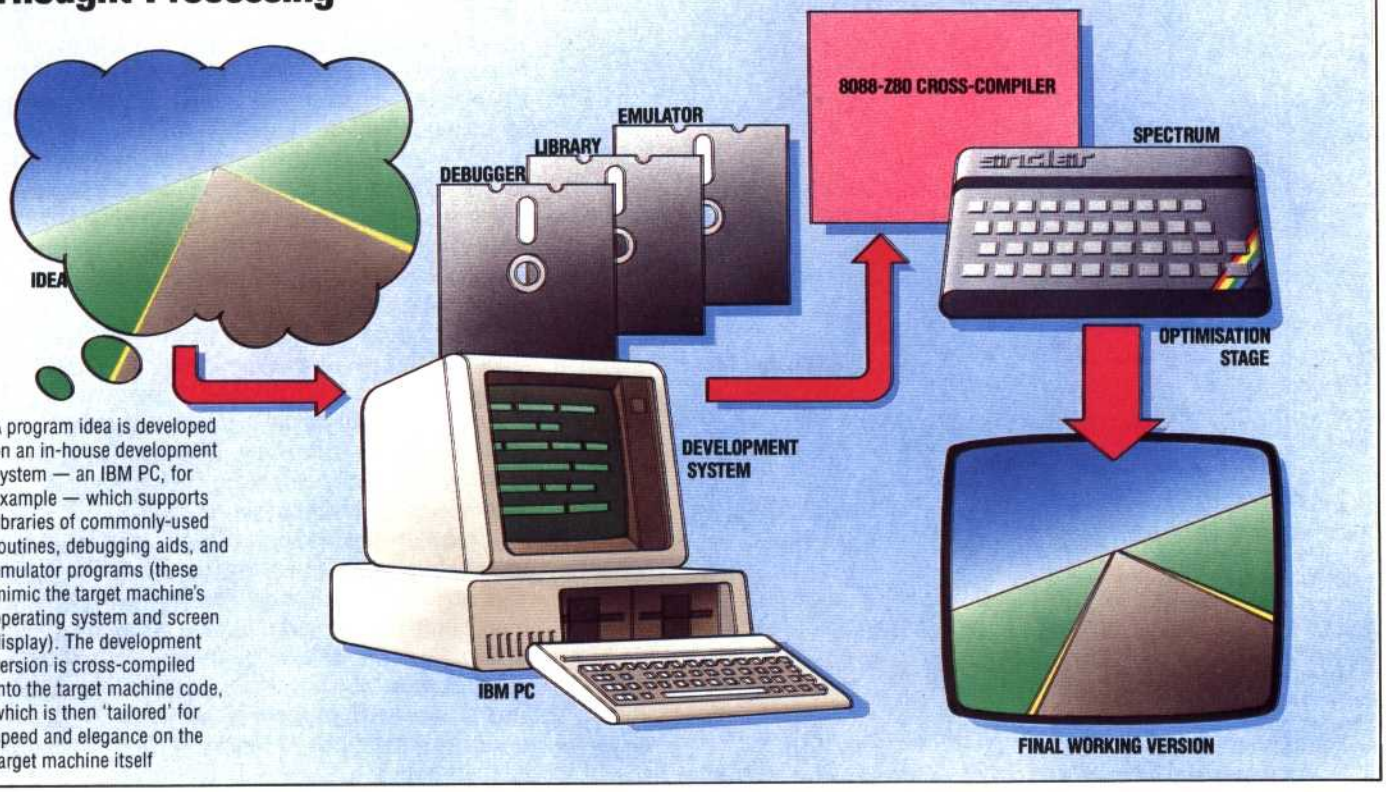
Expensive equipment doesn't necessarily mean successful programs; some amateur writers have managed to make a small fortune with software they've produced on a Spectrum at home. All the same, home programming whizzkids are becoming an endangered species, especially with the development of the big software houses over the past few years. Their powerful computers and sophisticated programming aids give them a real advantage over the home computer owner and allow their programmers to be more productive.

One of the most important attributes of serious software for home machines is the speed of operation; and this means that programs need to be written (at least in part) in machine code. But machine code is extremely difficult to work with — in particular, machine code programmers need other pieces of software to help write their programs. At the very least, an assembler program will be required to translate the programmer's source code into the object code that the machine understands, and this can be quite a challenging job if a big program is involved. Many software writers work in this way. To write a program for the Spectrum, for example, they will use an assembler program running on the same machine. This method has its limitations.

Primarily, the quality of assembler programs available for home machines is poor. Even the simplest of these packages will use up considerable amounts of memory, and therefore limit the size of the programs that can be written with it. Many home machines are also extremely unpleasant to work with for long periods of time: poor keyboards, poor displays and, in some cases, a lack of disk drives, can make using such equipment a tortuous task.

For these reasons most professional companies don't use the micro that the program is intended for (called the 'target machine'), but use business computers with special software (known as 'development systems') instead. Programmers using these machines often write in languages such as PASCAL and C. They use versions of these

Thought Processing



A program idea is developed on an in-house development system — an IBM PC, for example — which supports libraries of commonly-used routines, debugging aids, and emulator programs (these mimic the target machine's operating system and screen display). The development version is cross-compiled into the target machine code, which is then 'tailored' for speed and elegance on the target machine itself

KEVIN JONES