Description of investigation

The issue I am trying to solve is the lack of an educational tool showing how compilation and execution of low-level code works. Students at A-level learn about compilers and interpreters, but how this process works is often complicated and difficult to understand, and it would be extremely useful if they could experiment with the process. Sadly, actual compilers are very opaque and you cannot “watch” them doing the compiling, as is the process of running the executable.

The aim of my project is to solve this, by providing a fully-fledged educational system to show the steps from high-level code through executing to running the code. It will not quite be a full compiler to actual machine code, but instead to my own instruction set and executable file format. There are overall three parts to this project:

* A compiler to turn a high-level language (based on C) into bytecode
* An assembler to turn a simply assembly language into bytecode
* A bytecode interpreter to run executables

All of these will be designed with education in mind, to be user friendly and allow careful experimentation and watching of what happens. I will be attempting to keep to the OCR guidelines for how the stages of compilation work, and my method will also be guided by the C compiler. The high-level language will be a stripped down version of C, and my iterations of improving my program will partially involve adding gradually more features.

Once these programs are created, they will be able to act as a set of classroom tools, either for a teacher to demonstrate how compilation, assembly or Von Neumann-style execution look like. The toolkit will then be available to the students, who can experiment with the process. The compiler will walk them through the steps in the compilation so they can see how their code is compiled, and they can then watch as the virtual machine executes their code as if they are watching the CPU run a program.

The bytecode interpreter will be similar to the Little Man Computer, except that it will have a much larger instruction set, access to a large amount of memory and also some other changes, such as having multiple registers.

If my project is successful, teachers will be able to not only explain how the process from the software build to its final execution work, but also let them see with, and experiment with, the process all along the way.