

Jasm: A short story

The computer programming languages we use haven't changed much since the 1970s. Most languages are what one to two people can implement in the available time frame, and the rest are designed by committee. So, the language user has to do extra work:

- The user must type code into a text editor and try to imagine what it will do.
- Bug prevention is hard.
- Memory and multitasking issues cause crashes.
- The engineer has to input stuff here to match what was typed there (boilerplate).
- Fundamental programming structures must be hand-coded.

It is hard to program this way.

Programmers accept this because it has always been this way. There is no other way.

Third-party software packages check code, supplying specialized functionality, and help users visualize code. If used, they help a lot. Most programmers don't use them, largely because they cost extra and yield only incremental improvements.

It has been my career-long project to find the best tools, structures, and practices for preventing bugs and making code easier to understand. I became very productive by using them. It's usually an uphill battle to get co-workers to use them. Most programmers want to code and don't want to be bothered with doing extra work upfront. So, they spend half their time debugging.

Because of this, I created a software language called **Jasm** to do the heavy lifting for me. The coding problem collapsed into something simple. The computer does a lot of the work for me. The best way to do things is the easiest way to do them. The code is easier to understand. Jasm became what the other languages should have been but are not.

Jasm will take a small team of programmers to write, but it will do much more than existing software programs. A medium or large project can be completed in half the time.

admin (at) Jasm Tech (dot) com