Scripting and breakpointing is also part of this process. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Following a consistent programming style often helps readability. Allen Downey, in his book How To Think Like A Computer Scientist, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. However, readability is more than just programming style. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Normally the first step in debugging is to attempt to reproduce the problem. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. There exist a lot of different approaches for each of those tasks. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. One approach popular for requirements analysis is Use Case analysis. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. There exist a lot of different approaches for each of those tasks.