

Debugging Overview

What is debugger?

A **debugger** is a program for finding errors or bugs in the code, which automates the debugging process. It allows the programmer to observe the execution of the program for detecting the exact location of the error.

Earlier debuggers had a command line interface, where a programmer had to enter special commands to start working. Modern debuggers already have a graphical interface, which makes working with them a lot easier.

How debuggers work

The main principle of debugging is that it can freeze the program execution, i.e. suspend it. This way one can examine the program in this frozen state.

There are two ways to suspend the running code on debugging:

- set breakpoints in the code that you want to examine and wait until one of them is hit.
- break program execution by pressing hotkeys or the stop button if you are working in a graphical interface. The debugger will finish the statement that is executing at the moment and then will stop the program.

Breakpoints

A **breakpoint** stands for a deliberate interruption of program execution. It is set at the point in the program at which you want to check its operation. After the program reaches this point, it will pause.

Then you will have the opportunity to examine the state of the program, for example, logs, memory state, processor registers, stack, and so on.

In contrast to a complete stop, with the help of a break, in the debugger the program can be terminated or continued from the same place where the break occurred.

It is also worth mentioning that for the line of code with the breakpoint the result of this line operation is still unknown. So after the code is broken at the breakpoint, you can see only the results of all the previous lines before the breakpoint.