Cheap essay

cheapest essay writing services

Although the terms are important and are not important, the wrong choice of the term sometimes leads to the wrong choice of concept, the choice of an incorrect tool and the implementation of what is needed.

Since I am engaged in system programming (this is not the system, there is a speech about low-level programming), in conditions when it is impossible to simply take a more powerful computer, I often have to deal with real software structures that arose in the code when applied programmers tried to solve then your applied tasks. They think in terms of the subject area, but the characteristics of the language, and at times and equipment, are sometimes missed from the species.

One example is the case with a ring buffer, which I perceived as a specific term, namely, where the objects are placed on one side, and on the other are extracted in the same order. The developers who wrote this code actually required some finite storage with the replacement. The problem was that the implementation of this buffer was based on the original term, which did not meet the real scenario of using this buffer.

The buffer was implemented and enabled in the product. Subsystems working with data interfered with each other, so it was necessary to enter blocking to arrange access to the buffer. The blocking system turned out to be complex and regularly popped up problems that needed refinement and corrections.

Since we are talking about a low-level thing, then interested roles here are a little essentially other developers who should put in this storage and take the data. The requirement for the repository natural in the repository should be found what they want, and the revision appeal should not be blocked for several parts of the code for a long time. From the point of view of the buffer, the content of this data does not matter, only the record time mark.

From conversations with developers, it became clear that some data is needed only now while some others will be needed over the next ten seconds (in these conditions it is very long), so you cannot delete them. And about whether this data block is required to store for a long time, it is known immediately upon receipt of this block. Consequently, it makes no sense to handle all blocks equally. If you assign them a lifetime, then on the basis of this, you can determine when one or another buffer cell can be reusable.

As soon as the words of life appeared, the concept of buffer with obsolescence appeared, and after that the architecture of this buffer itself was formed.