Online Judge Information System Modernization*

Semen V. Teploukhov [1] [0000-0003-0099-0369], Alexander A.Chernenko [1][0000-0002-8182-0005] and Antonida A.Weber [1][0000-0001-9303-0011]

[1] Adyghe State University, Maykop, Russia
mentory@mail.ru
spiritfov@yandex.ru
toshamath@mail.ru

Abstract. Currently, the issue of informatization of education is relevant, therefore, online judge class systems are widely used in the field of online testing of students' knowledge. However, the problem of the complexity of the compiler messages for understanding by students remains relevant. The article discusses an information system for online checking of programming tasks, for which it is planned to implement a compiler message processing module. An algorithm for processing compiler messages is proposed, taking into account the need to modify and translate these messages. To achieve this goal, the open-source system "OnlineJudge" was chosen, which is written in the Python programming language and supports some other languages. On its basis, many error handling templates and corresponding alternative response templates have been developed. For this, a statistical analysis was carried out and the most common errors were identified, after which translation and auxiliary information were added for each of them. Such messages are processed by the compiler using regular expressions. As a result, the resulting system was introduced into the educational process to test a set of laboratory work on programming. The conclusion is made about the possibility of using this system in the educational process, besides, the open architecture of the software module allows you to additionally modify it, for example, to check the students' answers to plagiarism.

Keywords: Program Code Automated Testing, Online Judge, Compiler Errors.

1 Introduction

One of the main problems in the initial stages of learning to program is the complexity of the compiler messages. Students find it difficult to independently understand these messages since they are given in English and often contain terms and formulations that are difficult for a new programmer to understand.

Currently, the issue of the informatization of education is relevant. Particular attention is paid to the development and implementation of new solutions to improve the efficiency of the educational process with the help of modern information technologies.

However, the informatization process is still not completed [1, 2], so many teachers check students' programming skills mainly manually and are forced to view a large number of similar works in a limited time.

Automation of routine work when checking programming assignments will allow teachers to pay attention to more complex and interesting aspects of learning. Students, in turn, will have the opportunity to learn more effectively [3].

The goal of online judge systems [4, 5] is to provide a secure, reliable, cloud-based evaluation of user-uploaded algorithms. Stages of operation of the system shown in Figure 1.

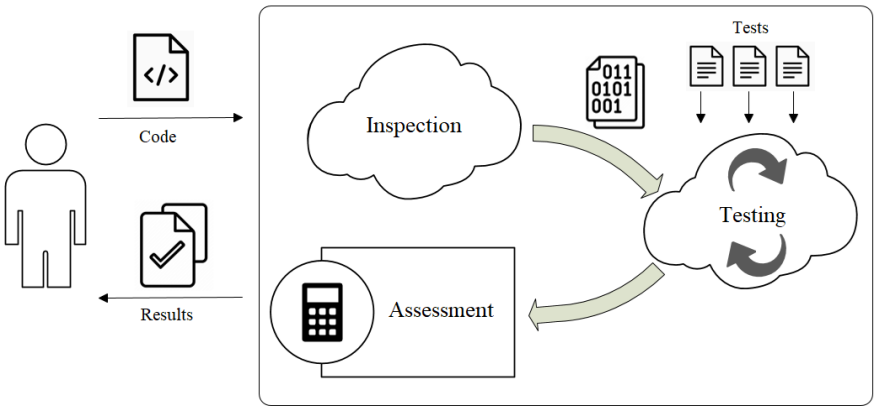

Fig. 1. Stages of operation of the online judge system.

To solve the indicated problems and increase the effectiveness of training, it is proposed to introduce into the educational process an information system for automated testing of programming tasks, which includes the function of processing compiler messages.

2 Theoretical part

Online judge class systems are widely used in the field of student knowledge assessment. An online judge is an online platform that real-time fully automated verification of the program code provided by the user [6]. This class of systems contains basic functionality, which can be modified to achieve the set goals with minimal labor and time.

As a result of the analysis of existing open-source online judge class systems, the OnlineJudge system was chosen for subsequent modernization, since it is based on a component approach, which facilitates the process of its modernization, the system also has a free license and supports several programming languages.

To introduce this system into the educational process, it is necessary to perform some improvements. First, you need to implement the compiler message handling module. Secondly, given that the "OnlineJudge" system is intended for holding programming

competitions, it is necessary to modify the system in such a way as to ensure the possibility of its integration into the educational process.

As a result of the analysis of the system, a block diagram is built, shown in Figure 2.

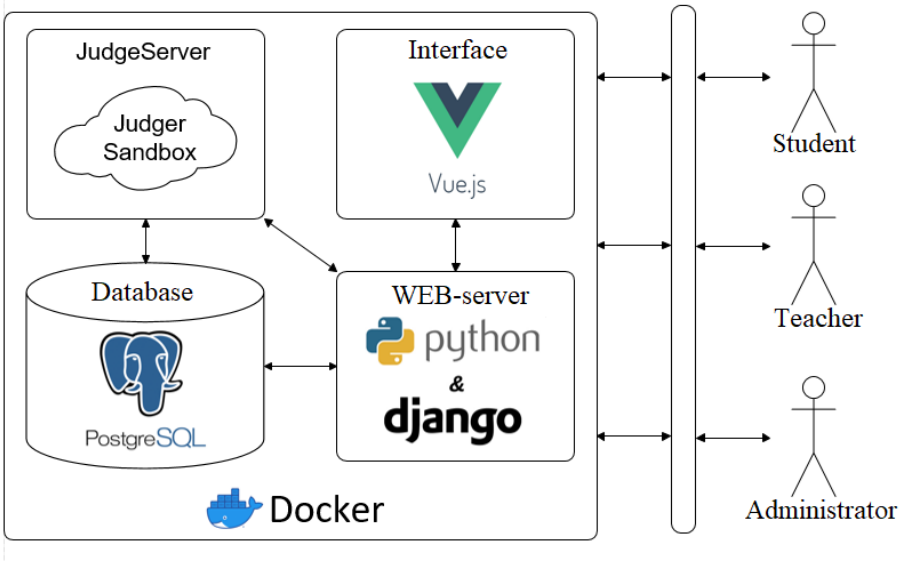

Fig. 2. Block diagram of the "OnlineJudge" system.

Compilation and testing of the code are performed using the JudgeServer and Judger-Sandbox modules. The result of the assessment, including the compilation errors that we must handle, is transmitted to the web server, which writes the necessary information to the database and communicates with the web interface, which carries out the output for the user.

3 Practical part

It is supposed to handle compiler errors for the C ++ language, in the selected system it is g ++ 5.4. It is not possible to process the entire list of compiler errors [7]. First, it is necessary to conduct statistical analysis and identify a list of the most common mistakes, after which each of them should be translated into Russian or another language (with the greatest possible simplification) and additional information on it should be added. The most convenient way to process messages is in a web server since the Django framework on which it is written provides convenient tools for working with a database where information about errors is recorded.

Docker is used to deploying this application. Docker [8] is an open platform for the development and operation of applications, the connecting link between all of the above components. As it was already found out at the stage of analyzing the subject area, one of the key factors in the development and support of online judge systems is security,

since the user code is executed inside the system and there is a high probability of a wide range of attacks. Besides, deployment is an urgent issue for the project. The system contains several modules that must function in one environment in harmony with each other. The process of configuring and deploying without the use of third-party applications is complex and requires additional steps, which increases the time spent on solving the tasks. Docker was chosen by the system developers to solve the above problems. The advantages of this tool:

- It is possible to separate the application from the infrastructure and work with the infrastructure as a managed application, which makes it easier and faster to launch and test when changes are made to the system.
- Isolation of applications when they run in a container provides the required level of system security.
- The ability to "package" the application in a container will simplify the process of handing over the product to its destination at the end of work on it. This will ensure ease of deployment and system support.

The algorithm for processing compiler messages is presented in Figure 3.

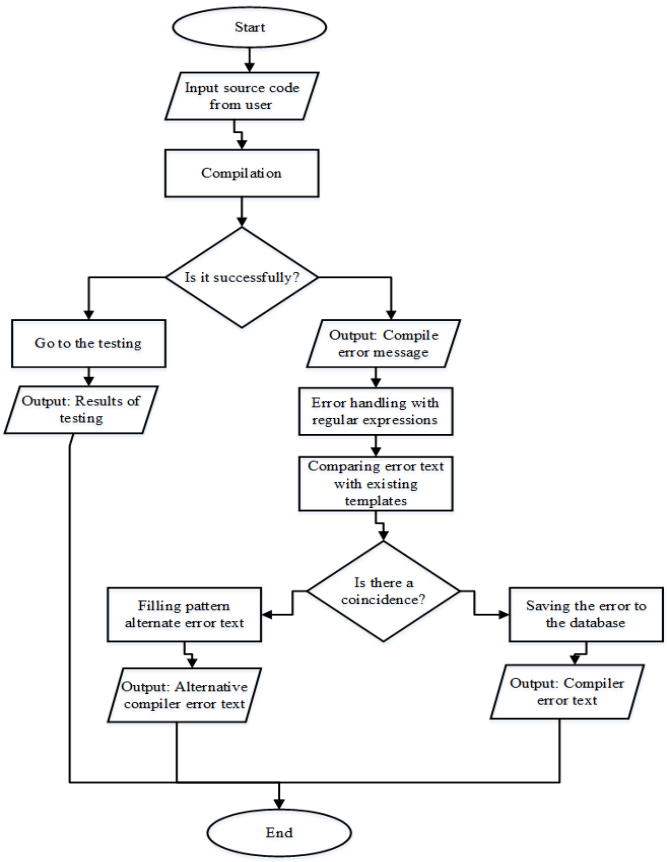

Fig. 3. Compiler message processing.

Before implementing the compilation error handling module [9], it is necessary to change the program code so that it contains the templates of the processed errors and the corresponding templates of alternative error texts. After the system detects a compilation error, the algorithm will try to find the required template in the list, if successful, it will match it with the alternative text, substituting the necessary data into it, such as the line number or the name of a variable, and transfer it to the interface for display on the screen.

Let's consider the operation of the algorithm using a specific example. The second line of Table 1 shows the code in which the variable is declared 2 times. This situation will cause an error, the output of which is shown in the third row of the table. The message is difficult to understand [10]. It is suggested to replace it with a simpler and more understandable message presented in the fourth line of Table 1.

Table 1. Example.

Brief description of the error	Re-declaring a variable

Cause of the error	int x;
	// some code
	int x = 10;

OnlineJudge output (g++ compiler)	/judger/run/ef81b41bd39b49fe8ff59f7cb20e6474/main.cpp: In function 'int main()': /judger/run/ef81b41bd39b49fe8ff59f7cb20e6474/main.cpp:9:7: error: redeclaration of 'int x' int x = 10; ^ /judger/run/ef81b41bd39b49fe8ff59f7cb20e6474/main.cpp:7:7: note: 'int x' previously declared here int x;
Alternative conclusion	On line {}, you tried to declare a variable that is already declared online {}. Two options: 1. If you needed to use a variable that you declared earlier, try removing the type name. The type only needs to be written when declaring. 2. If you needed to declare a new variable, try renaming it.
Brief description of the error	Condition not enclosed in parentheses
Cause of the error	if x == 10
OnlineJudge output (g++ compiler)	/judger/run/cf243a8e4bdf4cb9aa92614d2aceb824/main.cpp: In function 'int main()': /judger/run/cf243a8e4bdf4cb9aa92614d2aceb824/main.cpp:7:6: error: expected '(' before 'x' if x == 10 ^
Alternative conclusion	On the line {} (column {}) where you wrote the if, make sure the condition is enclosed in parentheses.

After the modules were implemented, they were tested and integrated into the system. Taking into account the changes made, the system works as follows.

Preparatory stage:

1. Deployment of the system.
2. The administrator adds several users to the system.
3. The teacher adds several practical tasks to the system.

At the next stage, the system functions as part of the educational process. Students solve problems, study, gain the necessary marks. In more detail, the process of solving a specific problem by a student follows the following scenario:

1. The student writes the code to solve the problem and sends it to the system.
2. The code is checked for originality. If the verification is successful, the code is compiled. In case of a compilation error, the compilation error handling module is activated (figure 4).
3. If the compilation is successful, testing takes place. During testing, some errors can also occur.
4. If the task is solved correctly and all the stages of the check have been successfully passed, then the system displays a message about success.

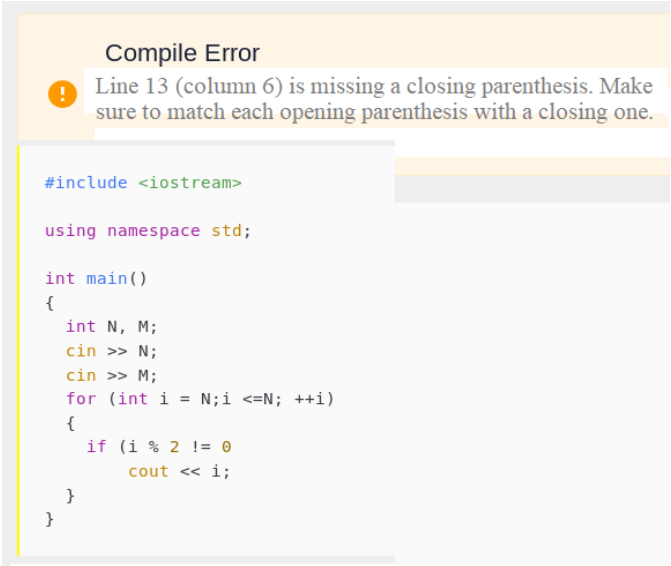

Fig. 4. Error example.

4 Conclusion

Thus, in this article, an information system for online verification of programming tasks was considered. The "OnlineJudge" system, which has an open-source code and is implemented in the Python programming language, was chosen as a basis. For additional processing of the compiler's messages, an algorithm for processing its messages was developed. The algorithm is based on the selection of templates of the most popular messages, their translation into another language, and their simplification to make it easier for students to understand. The work of the implemented software module is demonstrated by an example. Processed messages look simpler and clearer than the original ones, which allows for a more effective learning process. Besides, the open architecture of the software module allows for additional modification, for example, to check students' responses to plagiarism [11].

References

1. Woolcott, G.: Reconceptualising Information Processing for Education. Springer Singapore, pp. 168. (2020)
2. Shopin, A.V., Buchatskiy, P.Yu.: Electronic Information and Education Environment of the Adyghe State University. The Bulletin of the Adyghe State University, the series Natural-Mathematical and Technical Sciences, vol. 1(154), pp. 147-150. (2015)
3. Arakelov, A.V., Buchatsky, P.Y., Arakelova, U.A.: Preparation of Professional Personnel in the Process of Realization of the All-Russian Program "Digital Economy of the Russian Federation". In: 2nd All-Russian Proceedings on Proceedings Remote educational Technologies, pp. 20-24, Yalta, (2017).
4. Pham, M. T., Nguyen, T. B.: The DOMJudge Based Online Judge System with Plagiarism Detection, 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), Danang, Vietnam, pp. 1-6. (2019)
5. Wu, H., Liu, Y., Qiu, L., Liu, Y.: Online Judge System and Its Applications in C Language Teaching, 2016 International Symposium on Educational Technology (ISET), Beijing, pp. 57-60. (2016)
6. Wasik, S., Antczak, M., Badura, J., Laskowski, A., Sternal T.: A Survey on Online Judge Systems and Their Applications. Poznan University of Technology, ACM Computing Surveys, 35 p. (2018) URL: https://arxiv.org/pdf/1710.05913.pdf.
7. Rabin, Md R. I., Alipour M. A.: Configuring Test Generators using Bug Reports: A Case Study of GCC Compiler and Csmith, The 36th ACM/SIGAPP Symposium on Applied Computing, Software Verification and Testing Track (SAC-SVT'21), 9 p. (2020) URL: https://arxiv.org/pdf/2012.10662.
8. Azab, A.: Enabling Docker Containers for High-Performance and Many-Task Computing, 2017 IEEE International Conference on Cloud Engineering (IC2E), pp. 279-285, Vancouver, (2017).
9. Sharp, Ch., Assema, J., Yu, B., Zidane, K., Malan, D.J.: An Open-Source, API-Based Framework for Assessing the Correctness of Code in CS50. 25th annual conference on Innovation and Technology in Computer Science Education (ITiCSE), 6 p., (2020).
10. Stallman, R.M.: GCC 5.2 Manual 2/2, Samurai Media Limited, 442 p. (2015)
11. Asuni, N.: Quality Features of TCExam, an Open-Source Computer-Based Assessment Software, pp. 58-63. (2008).