IJETMR
International Journal of Engineering Technologies and Management Research
A Knowledge Repository

INDEPENDENT INTEGRATED E-TRAINING ENVIRONMENT

Asmaa Yasin *[1], **Bana Ahmed** *[2], **Reem El Hadary** *[3], **Yahia Hasan Jazyah** *[4]
*[1], *[2], *[3] Undergraduate students, ITC Department, Arab Open University, Kuwait
*[4] Assistant Professor, ITC Department, Arab Open University, Kuwait

Abstract:

Several programming contests are being held every year searching for the best team in programming. To achieve the first place in the context, training is a crucial factor for participants; where meeting regularly and face to face is important issue. In some cases, meeting face to face is not available for competent, so this research develops an algorithm for training and examining the source code of user. The algorithm is based on designing an additional thread to monitor and control the process of testing the source code. The algorithm is efficient and achieves high percentage of success.

Keywords: Compilation; Multi – Threading; String Comparison.

Cite This Article: Asmaa Yasin, Bana Ahmed, Reem El Hadary, and Yahia Hasan Jazyah. (2018). "INDEPENDENT INTEGRATED E-TRAINING ENVIRONMENT." *International Journal of Engineering Technologies and Management Research,* 5(12), 146-152. DOI: https://doi.org/10.29121/ijetmr.v5.i12.2018.336.

1. Introduction

Computer programs are going through several stages to be ready for release. Many programs are designed nowadays that target the same output, but still there are tangible differences in their performance and accuracy. And several researches aim to handle the problem of code efficiency. One of the problems that students face in programming training provided by universities is that they have to attend meeting on campus. Some students are not always available during the times these meetings take place. To overcome this problem, we decided to develop a program that works independently, supports E-Learning and helps in training of beginner students using online prospective. Our program supports data synchronization among different users and machines by implementing a database server.

Several applications software is existing to judge the source code of competent that try to find the optimum code and best timing.

The proposed software allows the students to develop and improve their programming skills so that they can be qualified to participate in programming contests in the future. Programming contests is about dividing a number of students into teams then giving each team a set of problems to solve and then the team that solved the highest number of problems correctly in shortest time gets the highest score. Then each team takes a rank.

The application is based on the same idea, where the problem sets test students' understanding. The proposed software compares the output obtained from the code submitted by the student and the ideal output that the code should have given which is stored in database table. Then the application can inform the user his code is correct or not. Students can view their progress among other fellow students via the rank provided in our software, which revives the spirit of competition. The rest of this article is organized as follow: section 2 shows the related work, section 3 describes the compilation and execution process of program, section 4 illustrates the multi-threading process in the proposed application, section 5 shows the string comparisons of the user and the one stored in the database, section 6 discuss the results and section 7 is the conclusion.

2. Related Work

In this section, a summary related to the topic is displayed. Radhakrishnan *et al*. [1] examine architectural issues from both the hardware and Java Virtual Machine (JVM) implementation perspectives; it starts by identifying the important execution characteristics of Java applications from a bytecode perspective. It then explores the potential of a smart Just in Time (JIT) compiler strategy that can dynamically interpret or compile based on associated costs and investigates the Central Processing Unit (CPU) and cache architectural support that would benefit JVM implementations.

Morimoto *et al*. [2] develop a system to support teaching computer programming to a group of students who have common questions and make common mistakes on practice computer programs. The system extrapolates the causes and syntaxes of students' compilation errors by analyzing the trends of past compilation errors and presents the extrapolated result to the teacher in real time.
Kiran and Moudgalya [3] attempt to predict the failures in programming comprehension and debugging skills based on programming errors generated by the learner.

Kanazawa *et al*. [4] develop trial product to evaluate students programming skill exactly. That product is named PEAS system. We analyzed examinee by PEAS system. As a result, we got not only the evaluation values but also student's good and weak points.

Dyvak and Taras [5] propose an algorithm of finding of groups of identical files, which contents are equal. Its low calculation complexity is showed. The software for realization of this algorithm is developed.

Guo *et al*. [6] present an innovative online instant assessment system for interactive teaching and student's active and collaborative learning introductory Java programming.

Tamada *et al*. [7] propose a framework to support programming education intended for novice programmers at universities. This proposed framework entails a collection of compile errors and source codes of a student at the time such errors occur, as well as storing the learning history of the student Novel computable encryption algorithms are designed. Then, a collusion-resistant outsourcing is achieved using non-interactive padding, partition, and expansion mechanisms. Results of simulation experiments demonstrate that the client-side overhead is quadratic in the input number of salt values, along with an optimal server-side performance. *Wang et al*. [8]

3. Compilation and Execution

After designing the source code, compiling and executing is very important step, which is the first step to test the algorithm. User's submitted code is deposited into a .Java file. Then, the file is compiled using a jar library that acts as a compiler for the application. The output from this step is a .class file that can be executed later for retrieving the output stream. In the compilation process, the file either compiles successfully or fails due to a compilation error.

Once the compilation step is done successfully with no errors, the code is executed and run in a different thread than the main application thread. This step either results in a runtime error from user's code or the retrieval of an output stream.

4. Multi-Threading

The proposed application is targeting students, and so, all different possible inputs submitted by students should be taken considered. One main concern of a submitted code is when it fails to end its execution, input the application into an infinite loop; this leads to serious problem which cause the application to freeze and stop responding. To handle this issue, multi-threading is used. The execution of user's submitted code is executed and isolated in a different thread than the main application's thread. A third thread is created to monitor and observe the thread that runs user's code.

The third thread waits for one second and then checks if the thread of running user's code is still executing, if it is still executing, the third thread will end user's thread by destroying it, and so protecting the application against the previous problem of users' code.

The selection of the one second threshold is because the problems set offered by the proposed application is simple and does not require long processing codes. This conclusion is obtained by running different forms of algorithms targeted to solve those problems, the results is that all of them elapsed less than one second to obtain the correct answer.

5. String Comparison

When user's code is successfully compiled and executed, an output stream is retrieved from that code; this output stream is then compared to the correct output stored in database. The comparison is performed byte by byte. If any of the bytes does not equal to the correct output, user's answer is rejected and considered as wrong answer.

The following steps summarize the previous discussion:
Step 1. Compile the submitted code as shown in figure 1.

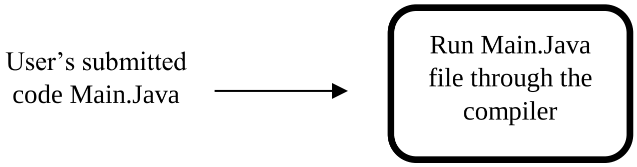
User's submitted code Main.Java
Run Main.Java file through the compiler

Figure 1: Compilation of source code.

ISSN: 2454-1907
DOI: 10.5281/zenodo.2532224

Step 2. If the result of compilation is successful, execute the source code as shown in figure 2.

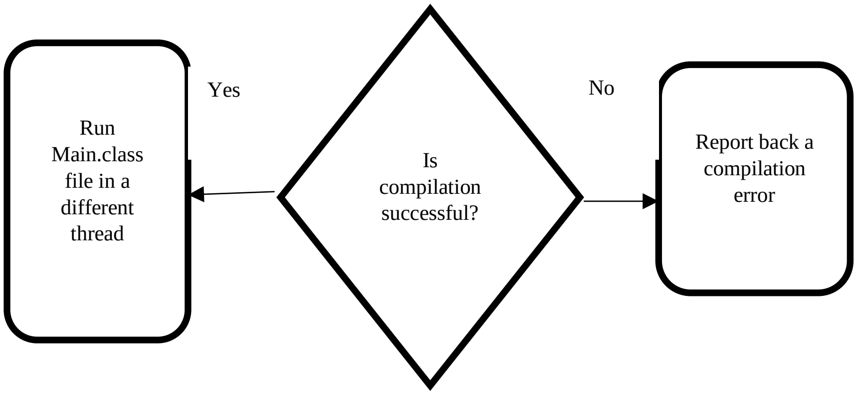
Run Main.class file in a different thread
Yes
Is compilation successful?
No
Report back a compilation error

Figure 2: Execution of source code.

Step 3. Create a separate thread (user Process) to run the submitted code's execution process
Step 4. If the result from executing the code is successful, an output stream is retrieved. Otherwise, there is a runtime error. As shown in figure 3.

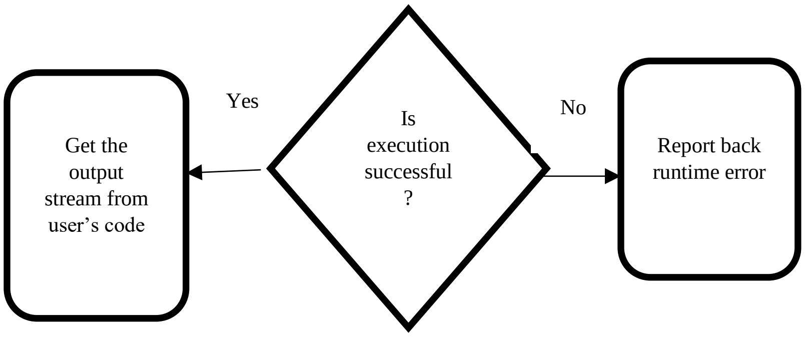
Get the output stream from user's code
Yes
Is execution successful ?
No
Report back runtime error

Figure 3: Handling the result of execution of source code.

Step 5. Create another thread (terminator) responsible for monitoring user Process thread's execution time that wait for TLE (Time Limit Exceed) then kills the user Process. See figure 4.

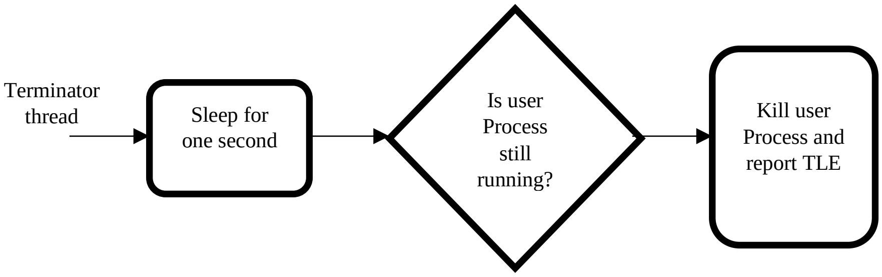
Terminator thread
Sleep for one second
Is user Process still running?
Kill user Process and report TLE

Figure 4: Terminator thread.

Step 6. The main thread checks whether both of the other threads still running. If yes, then it waits for one millisecond until one of the two threads stop. See figure 5

ISSN: 2454-1907
DOI: 10.5281/zenodo.2532224

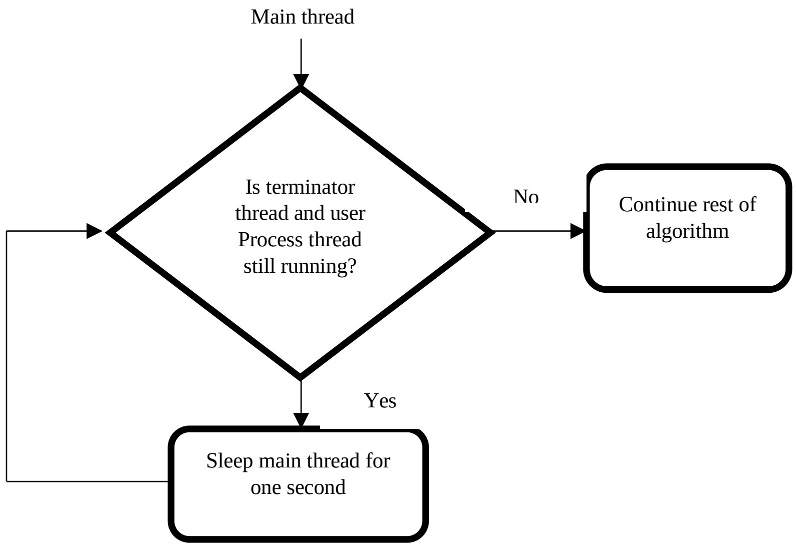
Main thread
Is terminator thread and user Process thread still running?
No
Continue rest of algorithm
Yes
Sleep main thread for one second

Figure 5: Processing terminator and user threads.

Step 7. If there is no compilation error, runtime error, or TLE, the output stream is retrieved from user’s code and compared to the correct output from the database.

Step 8. If the outputs are equal, then user’s code is accepted as a correct answer. Otherwise, it is a wrong answer. See figure 6.

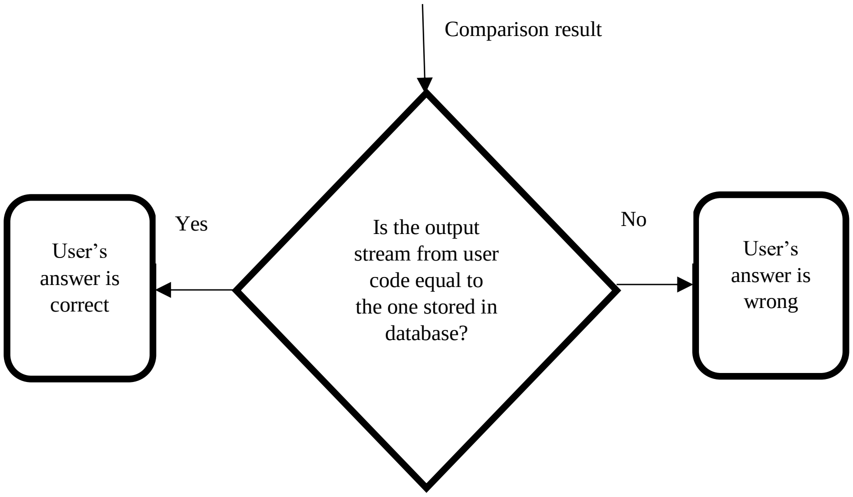
Comparison result
Yes
User’s answer is correct
Is the output stream from user code equal to the one stored in database?
No
User’s answer is wrong

Figure 6: Final step after comparison the two strings.

6. Result and Analysis

The algorithm is tested about 30 times using different user programs written in Java; each tested file has a size of less than 1KB saved in a text document. Each file is tested independently and separately.

The size of source code is not huge and should be optimized because it is dedicated for programming contest and so it should be as small in size as possible. The files include different types of programs that include if-statements, loops, nested loops, etc…

After testing the algorithm, all tested codes do not cause the system to freeze and give the output as necessary i.e. match or non match. The success rate is perfect; the reason for this high percentage of success is mainly that the algorithm is designed following a defensive programming technique. Having created a thread dedicated its sole purpose to "watch" and "monitor" the Main Thread that prevents from falling into a lot of several problems. Moreover, the designed application is purging and checking the input variables entered by the user in a way that it can prevent wrong inputs entered by users.

7. Conclusion

Application software is designed and implemented using Java programming language in order to enhance the efficiency of code comparison process used in programming contests. The testing result of the application stated perfect success; the application is able to overcome several problems such as infinite loops of tested Java programs and long time of execution by defining additional thread to monitor the application process.

Acknowledgment

This research was supported and funded by the research sector - Arab Open University – Kuwait Branch under decision number 18201. We thank everybody who assisted us to improve the study.

Ethics

We testify that this research article submitted to the International Journal of Engineering Technologies and Management Research, title: "Independent Integrated E-Training Environment" has not been published in whole or in part elsewhere.

This research project was conducted with full compliance of research ethic norms of Arab Open University - Kuwait.

References

[1] Radhakrishnan R et al. "Java runtime systems: Characterization and architectural implications", IEEE Transactions on Computers. 2001 Feb; 50(2):131-46. https://doi.org/10.1109/12.908989

[2] Morimoto Y et al. "A support system for teaching computer programming based on the analysis of compilation errors", Sixth International Conference on Advanced Learning Technologies, 2006 Jul 5 (pp. 103-105). IEEE. https://doi.org/10.1109/ICALT.2006.1652378

[3] Kiran EL and Moudgalya KM. "Evaluation of programming competency using student error patterns", 2015 International Conference on Learning and Teaching in Computing and Engineering (LaTiCE), 2015 Apr 9 (pp. 34-41). IEEE. https://doi.org/10.1109/LaTiCE.2015.16

[4] Kanazawa A et al. "Analysis of programming competency for PEAS system innovation: PEAS System: Programming experience assessment scale system". 12th International Conference on Computer and Information Science (ICIS), 2013 IEEE/ACIS 2013 Jun 16 (pp. 53-58). IEEE. https://doi.org/10.1109/ICIS.2013.6607816

[5] Dyvak and Taras. "The rapid algorithm of the files comparison with the hash functions usage.", In Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2008 Proceedings of International Conference on, pp. 99-100. IEEE, 2008.

[6] Guo, et al. "Design of Online Runtime and Testing Environment for Instant Java Programming Assessment." In Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, pp. 1102-1106. IEEE, 2010. https://doi.org/10.1109/ITNG.2010.227

[7] Tamada et al. "A framework for programming process measurement and compiling error interpretation for novice programmers." In Software Measurement, 2011 Joint Conference of the 21st Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 233-238. IEEE, 2011. https://doi.org/10.1109/IWSM-MENSURA.2011.23

[8] Wang et al. "E-SC: Collusion-Resistant Secure Outsourcing of Sequence Comparison Algorithm." IEEE ACCESS 6 (2018): 3358-3375. https://doi.org/10.1109/ACCESS.2017.2780129

*Corresponding author.
E-mail address: soso.mardan@ gmail.com/bana.ahmed93@ gmail.com/
r.elhadary@ hotmail.com/yahia@ aou.edu.kw