International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

A Heterogeneous and Optimized System for Trainers for Analysis and Rectification of Compilation Errors

Badr Almutairi, Sultan Almotairi

Abstract*: *In the computing world where information is being gathered and analyzed it's important for programmers to convert raw data into meaningful information and this can only be achieved through programming. The art of programming is not easy as many programming languages are present in the current programming world. The first programming course for any new programmer is always difficult. New programmers always try to copy the code provided by their instructors and after compiling the program, gets the number of errors which they are not able to solve. In this paper, we present a solution which helps the instructors and the programmers to know about student's capacity of coding. This system provides a top few errors with their solutions. The programmers and the evaluators can also use the system to optimize the program which the students and the other stakeholders are using. As this system alerts the programmers and the other stakeholder about the compilation errors which occur and thus providing instant help for rectifications of the code when needed.

Index Terms*: *Programming, Code Compilations, Code Optimization, Rectification of real-time errors.

I. INTRODUCTION

The study of programming in today's world is one of the important factors. Therefore, number of skilled Computer or IT programmers and other stakeholders are needed as the programming environments are totally optimized. This indirectly pressures the educational institutes to train their students in a proper way. A new programmer often tries to copy the program or just by heart the programs without understanding the program logic. On compiling such program get the number of errors which they can't solve. Teaching to such type of students in limited time is a quite difficult job for faculty. Additionally, the faculty has to understand the student's grasping power and their ability to solve the errors. After compiling errors, the student can't solve those errors as all the compilers just indicate the outstanding error which and the line number for an occurrence of errors with error codes. To solve such problem, there is a need for such structure which provides faculty about student's capability of understanding programming logics also provide solutions to errors accordingly in real time. Many researchers did not analyze the problems which have been faced by many programming experts and students in general. In this paper, we have thus taken this research work to find better solutions to reduce the effort and time required by the programmers and other stakeholders in this project so that the errors can be minimized and can be understood in a more predictable way.

II. RELATED WORK

As this research work has been carried out by many research's in the past we would like to through light on their contribution and the work accomplished with these researches never the more we have selected some researchers and their research work details.

Yasuhiko Morimoto et. al.[1] emphases on the error which are displayed to the stakeholder and the information regarding the error is visible to the instructors this system provides information to the instructors about the compilation errors which the paper explains but does not provide a solution to cater to the error handling thus compilation error are visible but are not rectified using this solution also it does not provide the instructor about how to inform the stakeholder about the error. It only displays the error without any help. This paper talks about an only stationary error and not real-time errors

Kazuhiko Nagao et. al. [2] would like to bring to notice that. As the error which the compiler displays are of different kinds of syntactic error, symmetric errors, lexical errors, intermediate code generation errors and many more. But this research article only focuses on the error which occurs in providing with UNIX programming environment these errors are specific to the programming environment and also do not display information about the error they only display the error codes which are not sufficient of the programmers to understand this will not solve the problem of error handling.

Dean Sanders et. al. [3] focus on the analysis and bug fixing of *c* programs. The bugs which are reported to the students are in simple statements which can be understood by the developers easily and also error messages are displayed in plain English like statements which can be understood by the stakeholders of the program the only drawback in this approach is that this application works only in UNIX environment and does not

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
Badr Almutairi, Assistant Professor, Department of Information Technology, College of Computer Sciences and Information Technology College, Majmaah University, Al- Majmaah 11952, Saudi Arabia,
Sultan Almotairi, Assistant Professor, Department of Natural and Applied Sciences, Community College, Majmaah University, Al- Majmaah 11952, Saudi Arabia

provide cross-platform integration and bug detection. This research project only provides and reports error at the end of the compilation of the program and does not provide real-time error messages to the stakeholder writing the program. It also fails to capture different types of compilation errors reported at different stages of the compiler. Thus this research is good for users writing programs and projects in the UNIX environment and then tracking bugs in the project which they have written. Loana et. al. [4] focused on error detection and bug tracking in a specific programming language which is JAVA this system is more robust and more stable and detects and notifies bugs to the stakeholder at regular intervals. All the bugs that are notified to the end user and the stakeholders are all calculated and computed then all the bugs along with the bug ID number are displayed to the programmers who are writing the code and developing the projects. This system only detects bugs which belong to certain phases of a compiler and all phases of a compiler are covered in this project as the bugs are specific to JAVA programming language this system can be further enhanced to work for multiple programming languages. The research working which has been conducted from the research articles above state that the systems failed in providing

- One time solutions with correct information to stakeholders involved in programming
- Providing solutions which already existed in the repositories online
- Enhancing and elaborating on the code optimization and bug fixing
- Cross-platform programming and bug fixing solutions which can work for different programming languages.
- Facilitate faculty to provide suggestion to a particular program of a particular student.
- Facilitating tools and techniques of programming which can enhance the performance and reliability of code written.

The proposed system provides support for students by facilitating them with Real-Time solutions to the compilation errors which they encounter while compilation. Also, the system provides support to the faculty by facilitating them with analysis of compilation errors, code-verification and provide suggestion to a particular program of the particular student by which the student can improve their coding ability.

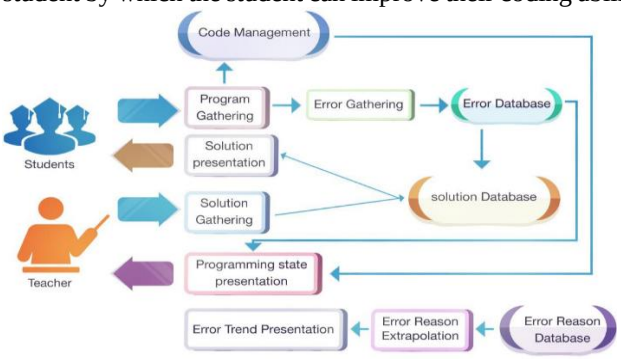

Fig. 1. System Architecture of the Proposed System

III. SYSTEM DESIGN

This research work helps different stakeholders in the programming environments to correctly write code and reduce a significant number of bugs encountered while programming. Fig. 1. summarized the proposed approach. Program content is stored in the separate file on the server and its entry is made in a database. All of the bugs the stakeholders come across are stored in the database. Never the less the information resides on the central repository where the stakeholders push the code or publish the codes online for execution. There are various repositories which will be used for detecting and managing the bugs which appear in programming these solutions will help in reporting and managing different bugs at various levels in programing environments which will be selected based in the programming environments which will be selected by the stakeholders who are writing projects and completing the code.

Bugs reporting being most significant part of programming languages based on the compilers which are written for different programming environments need specific understanding, see Fig. 2.

The central repositories that are required

1. Programming central database (MySQL)
2. Bug reporting Database (MySQL)
3. Corrections Database (MySQL)
4. Bug reasoning Database (MySQL)

The vertical subdivisions required are

1. Code Congregation (PHP)
2. Bug Congregation (PHP)
3. Elucidation Congregation (PHP)
4. Explanation Demonstration (PHP)
5. Program reasoning (PHP)
6. Bug reporting and analysis (PHP)
7. Bugs trends display (PHP)

A. DATABASES USED IN THE SYSTEM

1) CODE MANAGEMENT DATABASE: All the programs those are coded and saved by a registered student are stored on the server by creating a .C extension file in their respective directory and the entry is made in the Code Management Database.

2) ERROR DATABASE: All unique errors the system comes across are stored in the database. The solution is provided to each and every error by the faculty which is stored in the solution database.

3) SOLUTION DATABASE: This database holds solutions of all the errors which are provided by the teacher itself. This database is further used to by solution presentation module for presenting a solution to the student.

4) BUG CONGREGATION DATABASE: This database manages all bugs which are reported by different programmers who are working in different programming environments. The bugs of each compilation is placed in this database.

Retrieval Number: J93030881019/19©BEIESP
DOI: 10.35940/ijitee.J9303.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

B. ALGORITHMS USED FOR THE EXECUTION OF VARIOUS SEGMENTS:

1) *Program Gathering Segment:*

- Create File/Select File.
- Type code in textbox.
- Save File. Server is updated.
- Database entry made.
- Save reports database.
- Database entry made.
- Creating Repository.
- Bugs message display tools.
- Bugs rectification and helps.

2) *Error Gathering Segment:*

- Fetch code from the text file
- Compile using GCC Compiler
- Fetch Errors and enter them into a database
- Display Errors
- Fetch Errors from a compilation output file
- Perform text matching
- Sort Errors
- Enter them into a database

3) *Solution Gathering Segment:*

- Display new errors to the faculty
- Faculty enters Generalized Error Solution(GES) for pattern matching
- Faculty enters Solution the respective error
- Make entry into the database.

4) *Solution Presentation Segment:*

- An error is encountered.
- Perform Pattern matching between error and Generalized Error Solution(GES)
- if solution present, fetch the solution
- else display "Solution unavailable" make entry into error database as "New Error".

5) *Programming State Segment:*

- Select student
- View codes
- Select Program and click on Compile

6) *Error Trend Presentation:*

- View Analysis by Student/Batch/Class
- Click on Top Most Errors

- Displays top 5 errors

IV. USER INTERFACE

1) Usage by Stakeholders:

- Stakeholders types code and compiles the code.
- Stakeholders views the errors and their respective solution and get responses to the bugs
- Stakeholders uses the solution provided to overcome the bugs from the central repository of programming.
- Stakeholders can also request for recommendation for the program from different stakeholders.
- Stakeholders can work in collaboration with each other and push code to the central server.
- Stakeholders can also report bugs which they have encountered while programming.
- Stakeholders can also suggest solutions for the bugs which have been reported and solutions found.

Fig. 2. User Interface For The Novice Programmer From Www.Jdoodle.Com

V. RESULT AND ANALYSIS

*A. **PERFORMANCE***

The proposed system uses Ajax-JavaScript to update numerous parts of the User Interface. Thus, in turn it avoids the unnecessary reloading of entire page and leads to less amount of network usage database usage. Time for compilation is 0.2 sec, Time for solution display is 0.2 sec and time complexity is 0.4 sec.

*B. **SCALABILITY***

The proposed system is a Client-Server type of system. To test load balancing and scalability we connected 15 clients to the main server and the system was perfectly performing its functionality. As soon as we connected client 16 and client 17 to the server, the system functionality started degrading and the system became slow which concludes that the main server can connect at the max 15 clients to provide all the functionality to the clients without degrading system functionality

*C. **AVAILABILITY***

Availability defines that data should be available only to legitimate users and should not be available to users who are not registered to access it. The proposed system enables this quality attribute of availability. The users are requested to login before using system functionality and only authorized users with valid login-id and password are allowed to access data and data is available only to authorized users

Retrieval Number: J93030881019/19©BEIESP
DOI: 10.35940/ijitee.J9303.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

D. EFFICIENCY

The program written using different programming languages should be efficient to run of cross compilers and without much changes and modifications which are required in the system. The code written should efficiently handle error and should be autocorrected with hints provided to the developers for correction and a watch window for the same. On the same user interface the system should generate Realtime errors and should provide help and correct codes which can be automatically replaced with correct ones the programming environment should provide Realtime help and suggestions about the correct code which can be replaced with the wrong syntax codes.

E. DURABILITY

The code written by the programmers should last for long time with correct input and outputs provided if the user inputs wrong data the system should be able to handle the errors and generate correct error codes with correct English translations of what this code means so that the user can understand the code and can rectify the code which is provided at the help windows of the system the code with correct syntax can be easily replaced from the help window which is provided in the same solution explorer. This will improve program execution and code running ability of the program designers.

VI. CONCLUSION AND FUTURE SCOPE

We provide a solution that can be used for identifying bugs and reporting them to the centralized system. Which can be further taken into consideration by the programmers that such bugs can occur in different programming environments thus helping systems to work more efficiently and effectively. Also our approach can be used for rectification of the bugs which have been reported by different stakeholders during programming projects for different vendors. Rectification of the bugs can also be reported using our approach to help both students and instructors.

Different compilation bugs which occur while compiling programs under different environments can be reported and also solutions to such problems can be easily discovered by stakeholder who are writing code under such environment.

REFERENCES

1. Yasuhiko Morimoto,Kunimi Kurusawa, Setsuo Yokoyama, MaomiUeno, Youzou Miyadera, A Support System forTeaching Computer Programming Based on the Analysis of Compilation Errors, Sixth International Conference on Advanced Learning Technologies (ICALT'06).
2. T. Moreau, M. Wyse, J. Nelson, A. Sampson, H. Esmaeilzadeh, L. Ceze, M. Oskin, "Snnap: Approximate computing on programmable socs via neural acceleration", 21st International Symposium on High Performance Computer Architecture (HPCA), 2015.
3. S. Mazahir, O. Hasan, R. Hafiz, M. Shafique, J. Henkel, "An area-efficient consolidated configurable error correction for approximate hardware accelerators", 53rd Design Automation Conference (DAC), 2016.
4. M.K. Ayub, O. Hasan, M. Shafique, "Statistical error analysis for low power approximate adders", 54rd Design Automation Conference (DAC), 2017.
5. C. Li, W. Luo, S.S. Sapatnekar, J. Hu, "Joint precision optimization and high level synthesis for approximate computing", 52nd Design Automation Conference (DAC), 2015.
6. J. Huang, J. Lach, G. Robins, "Analytic error modeling for imprecise arithmetic circuits", Silicon Errors in Logic - System Effects, 2011.
7. A. Yaganteeswarudu, " The speaking compiler-A compiler with audio for immediate error correction", 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pg. 980-983, 2016.
8. W.-T. J. Chan, A.B. Kahng, S. Kang, R. Kumar, J. Sartori, "Statistical analysis and modeling for error composition in approximate computation circuits", 31st International Conference on Computer Design (ICCD), 2013.
9. J.S. Hu ; F. Li ; V. Degalahal ; M. Kandemir ; N. Vijaykrishnan ; M.J. Irwin, " Compiler-directed instruction duplication for soft error detection ", Design, Automation and Test in Europe, Page s: 1056 - 1057 Vol. 2, 2005.
10. Moslem Didehban ; Aviral Shrivastava ; Sai Ram Dheeraj Lokam, " NEMESIS: A software approach for computing in presence of soft errors ", IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pg. 297 – 304, 2017.
11. Jorge Castro-Godínez ; Sven Esser ; Muhammad Shafique ; Santiago Pagani ; Jörg Henkel, " Compiler-driven error analysis for designing approximate accelerators ", Design, Automation & Test in Europe Conference & Exhibition (DATE), Page s: 1027 – 1032, 2018.
12. Kazuhiko Nagao and Naohiri Ishii, A Concept of Agent based learning system for C Programming, 2nd International Conference on Information Technology Based Higher Education and Training, Kumamoto, Japan, July 4-6, 2001.
13. Dean Sanders and Brain Dorn, Classroom Experience with Jeroo, Central Plains Conference (CCSC) JCSC 18, 4(April 2003).
14. S. Lee, D. Lee, K. Han, E. Shriver, L.K. John, A. Gerstlauer, "Statistical quality modeling of approximate hardware", 17th Intl. Symposium on Quality Electronic Design (ISQED), 2016.
15. H. Jiang, C. Liu, N. Maheshwari, F. Lombardi, J. Han, "A comparative evaluation of approximate multipliers", Intl. Symposium on Nanoscale Architectures (NANOARCH), 2016.
16. A.K. Verma, P. Brisk, P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design", Design Automation and Test in Europe (DATE), 2008.
17. N. Zhu, W.L. Goh, K.S. Yeo, "An enhanced low-power highspeed adder for error-tolerant application", 12th International Symposium on Integrated Circuits (ISIC), 2009.
18. R. Ye, T. Wang, F. Yuan, R. Kumar, Q. Xu, "On reconfiguration-oriented approximate adder design and its application", Intl. Conference on Computer-Aided Design (ICCAD), 2013.
19. Ioana Tuugalei Chan Mow, Analysis of Student Programming Errors in Java Programming Courses, Journal of Emerging Trends in Computer and Information Sciences, Vol. 3, No. 5, May 2012.
20. S. Liu, K. Pattabiraman, T. Moscibroda, B. G. Zorn, "Flikker: saving dram refresh-power through critical data partitioning", ASPLOS 2011, pp. 213-224.
21. M. de Kruijf, S. Nomura, K. Sankaralingam, "Relax: an architectural framework for software recovery of hardware faults", ISCA, pp. 497-508, 2010.
22. D. T. Stott, B. Floering, D. Burke, Z. Kalbarczpk, R. K. Iyer, "NFTAPE: a framework for assessing dependability in distributed systems with lightweight fault injectors", DSN, pp. 91-100, 2000.
23. R. Maia, L. Henriques, D. Costa, H. Madeira, "XceptionTM - enhanced automated fault-injection environment", DSN, pp. 547-550, 2002.
24. V. C. Sharma, A. Haran, Z. Rakamaric, G. Gopalakrishnan, "Towards Formal Approaches to System Resilience", PRDC, 2013.
25. J. Wei, A. Thomas, G. Li, K. Pattabiraman, "Quantifying the accuracy of high-level fault injection techniques for hardware faults", DSN, 2014.
26. J. A. Stratton, C. Rodrigues, I.-J. Sung, N. Obeid, L.-W. Chang, N. Anssari, G. D. Liu, W.-m. Hwu, "Parboil: A revised benchmark suite for scientific and commercial throughput computing", Center for Reliable and High-Performance Computing, 2012.
27. H. Cho, S. Mirkhani, C.- Y. Cher, J. A. Abraham, S. Mitra, "Quantitative evaluation of soft error injection techniques for robust system design", DAC, 2013.
28. B. Sangchoolie, F. Ayatolahi, R. Barbosa, R. Johansson, J. Karlsson, "Benchmarking the hardware error sensitivity of machine instructions", Proceedings of the 2013 IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE Workshop), 2013.
29. The PHP Group, May. 2019, http://www.php.net,
30. Free Software Foundation, Inc. , May. 2019, http://gcc.gnu.org.

Retrieval Number: J93030881019/19©BEIESP
DOI: 10.35940/ijitee.J9303.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

AUTHORS PROFILE

Badr Almutairi received his PhD. degrees in computer science from De Montfort University, UK, in 2014. Currently, he is an assistant professor in the Computer Sciences and Information Technology College of Majmaah University. His research interests include deep learning, usability and improve the performance of communication, e-learning systems, and education Technology.

Sultan Almotairi received his B.Sc., M.Sc., and PhD. degrees in computer science from Florida Institute of Technology, Melbourne, USA, in 2010, 2012, and 2014 respectively. Currently, he is an assistant professor at Department of Natural and Applied Sciences in Community College of Majmaah University. He has acted as the dean of Community College at Majmaah University since June 2015. In 2016, he was elected as the Chairman of the Municipality Council of Majmaah. His research interests include neural networks, deep learning, pattern recognition, machine learning, image processing, and computer vision.