ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013

Design of Dynamic Error Handling Tool

Komal R. Patel, Harshita C. Kanani

Abstract— **In any programming language or environment, error messages perform role of intermediate between system (program development tool) and programmer. So it is important point of interaction in system. This point becomes critical when system is used by novices or beginner programmers, who have not enough knowledge of grammars or syntax of languages. Solution to this, paper present new tool which work as post compiler for GCC. Goal of tool is to enhance the error handling phase of compiler by dynamic error handling and making error messages more users friendly also provide hints to solve the errors. Heart of the tool use GCC output file as input file and produce new output file which have the more prescribed form of error messages with hints to solve errors. Also it provides dynamic error handling by keeping track of errors generated by compiler and updating predefined error hints file based on observing error correction.**

Index Terms— **Compiler, Error Handling, GCC, Post Compiler**

I. Introduction

Compilation is a process that translates a program in one language (the source language) into an equivalent program in another language (the object or target language)[8].

Programs submitted to a compiler often have errors of various kinds. With regard to errors, most compilers are not very effective at communication since they deliver their comments to the user without any knowledge of the user's "intent" for a sentence. Since no user has a perfect knowledge of syntax and semantics, errors are inevitable. The compiler detects an error, and then attempts to recover from the error so that it can detect more errors. That is, even in the presence of errors, the compiler tries to parse the entire the program in order to detect as many errors as possible.

Error Representation is different for the different compilers. Some compilers show error messages and line numbers, where some show line number, position in line and error message. In Research we follow GCC complier for C language.GCC produce output file which have line number, error message and sometimes with position in line at which error occurred. Beginner might find difficulty to solve the error just seeing error message and line number of error.

In GCC, generated error messages show line number at which they occur and small description of error. For novices or beginner might be difficult to find, represent, understand, and correct error in programs just seeing compiler error messages [2][9][10]. Solution to this problem is to enhancing representation of the error messages more user-friendly and provides hints to solve it. So Goal of this dissertation is to increase the scope of error handling in GCC. To accomplish the goal paper proposing a tool which work as post compiler. Heart of tool use GCC output file as input file and produce new output file which have the more prescribed form of error messages with hints to solve errors. Also it provides dynamic error handling by keeping track of errors generated by compiler and updating predefined error hints file based on observing error correction.

II. Previous Research Work

Most of the compilers development is performed years ago, so they haven't any special technique to handle novice or beginner users. They face various problems [1][2][3][13]. Some of them are:

- Difficult to find, represent, understand, and recover error in programs just seeing compiler messages
- Difficult to correct the error
- For novice, error messages are hard to understand
- Some bugs (pitfall) are not detect by compiler but create serious error, for ex: -----, +++
- Lack of program syntax knowledge to write code

Tool verificator [1] is developed for c++ and explain all causes of incorrectness of program more clearly and more understandable way than error messages and warning do. It maintains relation to the error messages produced and actually error in program. Another tool is fall in C [13] which is developed for C language. It detects some pitfalls by these three techniques: Regular expression searching, Structural pattern matching, and Message analysis of external programs. Another is online programming development tool [2] which help programmer to debug the program and Based on GCC. Gauntlet [10] is made for java. It works for pre compiler for java. It explains each error in layman's term. Hic [12] tool is based on identifying a subset of C++ which is appropriate for introductory students. Our proposed tool solve novices problem easily and provide better error representation then other tools also provide dynamic error handling.

III. PROPOSED SYSTEM

Proposed system (tool) take input as .c file and produce output file with error messages and hints. Tool is written in C language and Linux Commands. It works as post compiler; take input as a GCC output file and produce new modified output file. This output file have more described error messages than GCC's output file, hints and fixes of error if it have. In this system, we focus only on top 50 errors first and then think about other errors.

***Komal R. Patel**, M.E Student, Department of Computer Engineering, LDRP Institute of Technology & Research,, Gandhinagar, Gujarat, INDIA.*

***Harshita C. Kanani**, Asst. Professor, Department of Computer Engineering, LDRP Institute of Technology & Research, Gandhinagar, Gujarat, INDIA.*

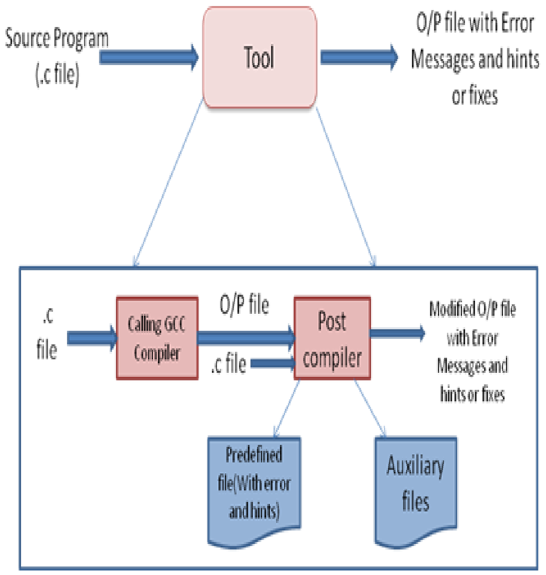
Source Program (.c file)
Tool
O/P file with Error Messages and hints or fixes
.c file
Calling GCC Compiler
O/P file
.c file
Post compiler
Modified O/P file with Error Messages and hints or fixes
Predefined file(With error and hints)
Auxiliary files

Fig. 1: Proposed Tool

Tool provides dynamic error representation. Means when you run your program more that once then error representation of same error become more error associated than the previous one. For example, when you have error of missing semicolon, then this error is given to the line more than one where its actually present. In the tool this error is shown at actual line number when this tool already faced this error and it have hint and fix to solve it. So directly it shows actual line number. Whenever user solved the error, tool store the change in file as hint or fix as in its predefined file. Thus tool work on the principal of Artificial Intelligence, but it is not it. Some part of it.

Proposed system uses two types of file to describe the error message. One is predefined file and another is auxiliary file. Predefined file have list of errors, its description, hints and fixes if available. It works as database of errors. Tool provides dynamic error handling by keeping track of errors generated by compiler and updating predefined error hints file based on observing error correction by user. For each program two auxiliary file created separately. One file stores input (.c file) and second one stores output file generated by tool. By using these files tool fulfill concept of the dynamic error representation. By examine the previous output file and current output file, find which errors are deleted and then compare the previous input file and current input file. This comparison may give the error fixes. This fixes store into the predefined file and error hints and fixes is changed.

IV. FLOW OF PROPOSED SYSTEM

Proposed System work on 4 stages:

1) Initial Stage
2) Error Checker Stage
3) Interpretation Stage
4) Copy stage

Diagram shows how each stage is connected to each other.

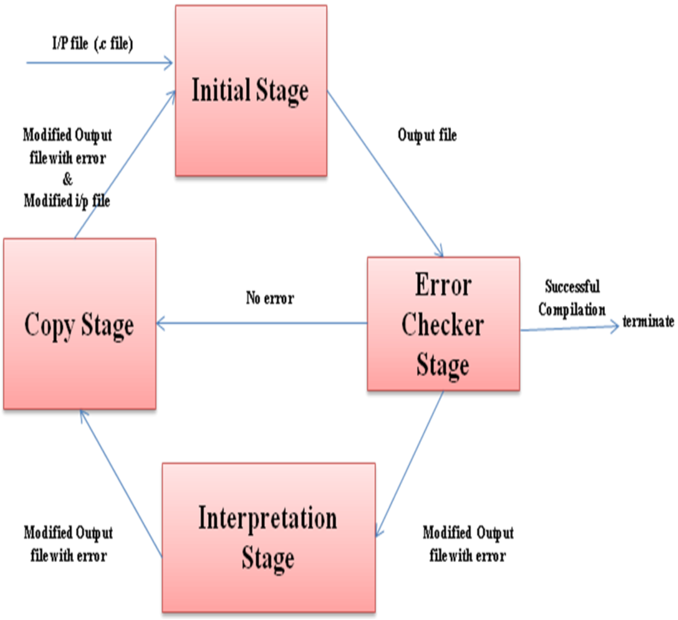

Fig. 2: Stages of Proposed Tool

Proposed system (tool) is work as post compiler. In Initial Stage, input file is must be .c file (written in C language). Here tool use GCC or calling GCC to compile and run program. GCC output is then used by the tool.

In Error Checker Stage, Proposed tool first check file has errors or not. If error is not present in file then successful compilation of program and go to the Copy Stage.

If file have errors then it's shown by GCC as output file with error line number and short description of error. Heart of tool gets this file as input file.GCC generated output file have error so now it check it is compile program first time??

It checks any hint or fix available for particular error in predefined file. If available then show hint or fixes with error messages else show only error message. The predefined file has Error massages, hints, fixes and brief description of error. This file updated during the Interpretation stage. Then it goes to the Interpretation stage.

If file compile more than once then we have to perform one additional step (Interpretation Stage). In This step, auxiliary output file and current output file are interpreted with each other. If file have change in it, then check any error is solved in current file. This Checking will be done by interpreting line number of error messages and others characteristics. Any line number missing from previous version of file then it means this error is solved. So hint or fixes is given by interpreting current and auxiliary input file line by line. This hints and fixes stored in predefined error file with its hints and messages. And then goes to the Copy Stage.

If file compile first time no need to perform interpretation with auxiliary files, directly it goes to Copy Stage. In Copy stage, both input and output file both is copied into separate auxiliary file. After that output is shown to user. Diagram below shows flow of tool.

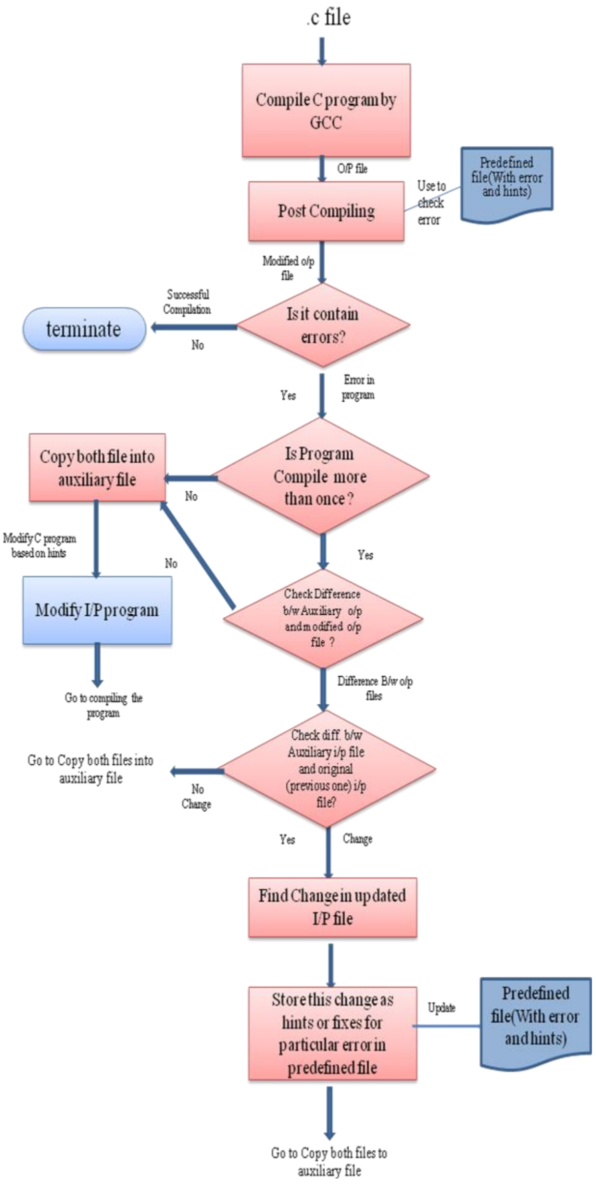
.c file
Compile C program by GCC
O/P file
Post Compiling
Use to check error
Predefined file(With error and hints)
Modified o/p file
Successful Compilation
terminate
No
Is it contain errors?
Yes
Error in program
Copy both file into auxiliary file
Is Program Compile more than once?
No
Modify C program based on hints
No
Yes
Modify I/P program
Check Difference b/w Auxiliary o/p and modified o/p file ?
Difference B/w o/p files
Go to compiling the program
Go to Copy both files into auxiliary file
No Change
Check diff. b/w Auxiliary i/p file and original (previous one) i/p file?
Yes
Change
Find Change in updated I/P file
Store this change as hints or fixes for particular error in predefined file
Update
Predefined file(With error and hints)
Go to Copy both files to auxiliary file

Fig. 3: flow of Tool

V. CONCLUSION

Programming is not about syntax – it is about problem solving. Tool allows novice programming to focus on the problem solving aspect of programming and gain confidence using technology to solve problems. Proposed tool enhanced error handling phase of GCC compiler of Linux. Dynamic error representation is help to update the error hints. Compiler messages interpretation become easier by using this tool. This Tool is helpful to novice programmer because it solved the novice's difficulties to handle the errors. Make programming easy for beginner.

REFERENCES

[1] Danijel Radosevic and Tihomir Orehovacki: "An Analysis of Novice Compilation Behavior using Verificator", IEEE/2011.

[2] Anmol More, Jitendra Kumar, Renumol V. G. "Web Based Programming Assistance Tool for Novices", IEEE/2011

[3] Guillaume Marceau, Kathi Fisler, Shriram Krishnamurthi , "Measuring the Effectiveness of Error Messages Designed for Novice Programmers",ACM/2011

[4] McLaren BM, DeLeeuw KE, Mayer RE: "A politeness effect in learning with web-based intelligent tutors", International Journal of Human-Computer Studies 2011, 69(1-2): 70–79.

[5] Traver VJ. "On Compiler Error Messages: What They Say and What They Mean", Advances in Human-Computer Interaction 2010; 2010: 1-26.

[6] Radoševi_ D, Orehova_ki T, Lovren_i_ A.Verificator: "Educational Tool for Learning Programming" Informatics in Education 2009, 8(2): 261-280.

[7] Masato KIYAMA1 and Hyo ASHIHARA: "Syntax Error Repair with Dynamic Valid Length in LR-based Parsers ", IEEE/2008.

[8] Dr. Matt Poole, Mr. Christopher Whyley, "Compilers", 2006/2007

[9] James Jackson, Michael Cobb , and Curtis Carver : "Identifying Top Java Errors for Novice Programmers", IEEE/2004.

[10] Thomas Flowers,Curtis A. Carver, and James Jackson :"Empowering Students and Building Confidence in Novice Programmers through Gauntlet " , IEEE/2004.

[11] Kazuhiko Nagao and Naohiro Ishiil : "A Concept of User Agent for User Interface of C Compiler " , IEEE/2002.

[12] Robert W. Hasker :" HiC: A C++ compiler for CS1",2002.

[13] Tetsuro Kakeshita , Mariko Oda ,Yoshihiro Imamura : "Fall-in C : A Software Tool for Pitfall Detection in C Programs ",IEEE/1994.