Interface - To Simplify Low Level Language Programming

Shailendra Vidhate[1], Diksha Bhusal[2], Mohini Nemade[3], Rutuja Lohokare[4]

[1] Professor, Department of Computer Engineering, MET's Bhujbal Knowledge City, Maharashtra, India
[2] Student, Department of Computer Engineering, MET's Bhujbal Knowledge City, Maharashtra, India
[3] Student, Department of Computer Engineering, MET's Bhujbal Knowledge City, Maharashtra, India
[4] Student, Department of Computer Engineering, MET's Bhujbal Knowledge City, Maharashtra, India

---***---

Abstract - *As we all are aware that assembly level language programming being hardware dependent is not easy to work in with for engineering/non-engineering students. Due to user friendly high level programming language, hands-on with low level assembly language is difficult. This limitation supports exploring new avenues to provide platform in order to simply fast executing hardware dependent assembly language programming. The existing system is efficient but is suitable only for the technicians as it is not convenient for the beginner's level programmer to get acquainted with such systems. First, this paper briefly looks at ways to provide simplified platform for assembly level language programming [8]. Hence our proposed system aims to make low level language programming easy to beginner's level user. Next, some possibilities to simplify input/output are considered and ways to structure assembly language are examined. Lastly this paper reports that existing system editor find itself a much better shape and quite closer to informal students with appropriate suggestions. So the proposed system would give the user a user friendly platform through an interface. And this interface would give enhanced features like predefined key-words differentiated with colors, dynamic support to instruction set, insertion of procedures, help on one click etc.*

Key Words: **Ubuntu 14.04, Eclipse, Java, Intellisense, NASM, Database, Assembly Language programming (ALP).**

1.INTRODUCTION

Although the market for assembly programmers is much smaller than it was in the past, there still appears to be a need to learn assembly language programming in order to closely understand the microprocessor hardware. Probably the most important reason to learn assembly language programming is that it is a good educational tool that allows students to learn about the inner working of computer microprocessor hardware [8]. Given the one to one relationship between assembly language and machine language, students can gain an understanding of topics such as registers, addressing schemes, instruction format, and machine language in a computer system course.

If we accept the above, that assembly language programming is probably a good way for computer science students as well as non-technical users to learn about computer system and it is also difficult for some students to learn assembly language programming, then methods to make assembly language programming simpler for students should be explored. For this here we are introducing user friendly editor. This is supported by Eclipse GUI for user programming. Features are provided like intellisense which can provide suggestions to user while working with instructions of programming, Predefined keywords with differentiated colors will help user for easy understanding of program, help is provided regarding ALP. Very few editors are available that provide a smart prompt for assembly language programming. The existing assembler', simulators or emulators implements a complicated user interface which creates ambiguity for new users in its usage. Also lack of smart prediction feature sometimes slowdowns the programming development process. Hence a system is being proposed that will help the programmer in speeding up low level programming, reducing errors and assist in the same. The proposed system would provide user with appropriate suggestions in accordance and in context of the code being implemented.

1.1 Motivation of Project

The Proposed system would provide the user Smart interface for Assembly Language Programming which can be speed up by suggested internal features. Hence the new user or students will use this system more efficiently. As, the time required is less this is also beneficial for the technicians who could cover more number of projects in less number of time. But mainly this system would create more interest for beginners.

1.2 Literature Survey

While studying the Assembly level language programming we come to know that there are Software tools used for editing, assembling, linking and debugging 16/32/64-bit assembly language programming. Basically we need an assembler to perform this assembly programming language and an assembler is a software utility that converts source-code programs written in assembly language into object files in machine language. The Intel family of processors supports assemblers like MASM (Macro from Borland), NASM (Net wide Assembler) TASM (Turbo) and YASM for both Windows and open source operating system.

As we have gone through the survey, we have found various kinds of inadequacy that the users or programmers face while working with the existing assemblers i.e. platform for low level assembly language programming. Drawbacks for existing system if taken into consideration is the existing system is a bit complicated for new users in its usage and this most of the time results in slow development processes. The developers who are very well sound about low level assembly language are only able to work quite efficiently. But new users either need to have thorough knowledge then only he could make up to using the existing system.

1.3 Existing System

We have existing simulators and emulators for ALP like simple 8-bit assembler simulator for 8-bit programs, x86 emulator with DOS, visual DSP++ 3.5 for 16-bit programs, PUP microprocessor simulators and emu8086 for 32 and 64-bit programs. Also x86 emulator which is a software program that is designed to replicate the function of x86 machine code instruction's in another or the same type of processing environment [9].

Also we have existing assemblers like Microsoft Macro Assembler (MASM) an x86 assembler that uses the Intel syntax for MS-DOS and Microsoft Windows for 16-bit programs, the Netwide Assembler (NASM) is an assembler and disassembler for the Intel x86. It can be used to write 16-bit, 32-bit (IA-32) and 64-bit (x86-64) programs [7]. NASM is considered to be one of the most popular assemblers for Linux and TASM itself is a 16-bit program; it will run on 16- and 32-bit versions of Windows, and produce code for the same versions. If u narrow your search and look for x86 assembler emulators, you will find commercial DOS Emulators [7]. But all these existing systems are hardware independent that allows user to perform programming only in 16 or 32 or 64-bit programs each at a time and these systems are efficient only for the technicians.

1.4 Proposed System

Considering the literature survey we are trying to provide solutions to few drawbacks of the existing system like difficult to identify and interpret the errors, difficult to debug in order to identify logical errors, switching between multiple windows for overall execution. Due to all these drawbacks the existing system resist beginners to explore the ALP effectively. Hence we are proposing an Interface where user could code for any of the choices provided like 16-bit, 32-bit and 64-bit programs. Because as per the survey this was not possible using the existing tools. Hence the proposed system will be providing an interface for user where he/she can work in with as per their choice and get the programming needs fulfilled. Also user could interact with low level assembly language with no hesitation even if he does not have any prior knowledge about this instruction set of ALP as we are providing programming details though help window. This Interface facilitate user like predefined key-words differentiated with colors, dynamic support to instruction set, autosuggestion, help on one click, PDF generation. Due to this new user and Technician people will surely be able to complete their work with less efforts in less time and effectively.

2. ARCHITECTURE

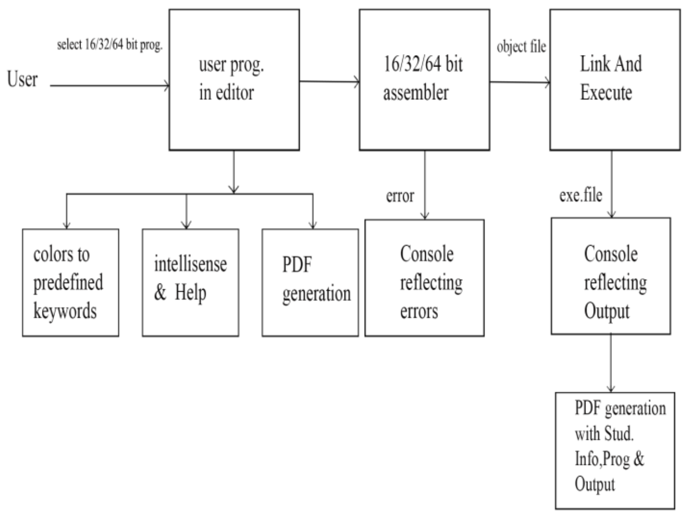
User
select 16/32/64 bit prog.
user prog. in editor
16/32/64 bit assembler
object file
Link And Execute
error
exe.file
colors to predefined keywords
intellisense & Help
PDF generation
Console reflecting errors
Console reflecting Output
PDF generation with Stud. Info,Prog & Output

Fig -1: Architecture

User program will be provided to eclipse front end GUI which is currently using 16/32/64-bit assembler program for open source operating system. The eclipse front end GUI will provide intellisense help, instruction set help, assembly programming font help. After running the program, the control will be passed to output generator as shown in Fig-1. Now this output generator will collect user's information for example name, roll no etc. Further the PDF will be generated for desired output.

3. GOALS AND OBJECTIVES

It facilitates the user to perform user friendly, Assembly Language Programming through Interface. It provides intellisense, Help, Insertion of procedures, predefined keywords differentiated with colors, dynamic support to instruction set etc. for user.

4. APPLICATIONS

It can be used in colleges for students. As assembly level language programming is little bit difficult because of its complexity and unfriendly nature and if any student wish to do programming he would need thorough knowledge and takes lots of efforts to deal with syntax each instruction set. But this system is providing user/student many features like intellisense and all that makes it very user friendly. this system can also be used by technicians. technicians or people in this field would definitely know about low level language programming but this system would help them to do their work in less time to achieve new goals and can be used in offices for training purpose.

5. ADVANTAGES

- It is very much user friendly.
- New users having very less knowledge can also be very comfortable with the assembly programming.
- The proposed system provides smart prompt for assembly language programming using intellisense and many more features.

6. DISADVANTAGES

- This system would be hardware dependent. So the code needs to be upgraded for higher versions of the processors and the current system code would not be supported for coprocessors. The operating system dependency can also be resolved as future scope.

7. CONCLUSION

With all the accumulated effort invested in designing an interface for ALP, there are reasons to believe that at the end of the project Interface: To Simplify Low Level Language Programming finds itself in a much better shape and quite closer to new users with appropriate suggestion in accordance and in context of the code being implemented than it was. We summarize the progress with respect to the main objectives of the project namely, smart prompt for assembly language programming, predefined keywords differentiated with colors, dynamic support to instruction set, insertion of procedures.

REFERENCES

[1] Intel 80386 Reference Programmer's Manual.
[2] "Intel Fellow - John H. Crawford". Intel.com. 2010-08-16. Retrieved 2010-09-17.
[3] A K Ray, K M Bhurchandi, Advanced microprocessors and peripherals

[4] INTEL 80386 programmer's reference manual 1986

[5] "Chronology of microprocessors (1990-1992)". Islandnet.com Retrieved 2010-09-17.

[6] A Tiny Guide to Programming in 32-bit x86 Assembly Language by Adam Ferrari.

[7] Netwide Assembler-Wikipedia, the free encyclopedia.

[8] Simplified Assembly Language Programming, James T. Streib.

[9] X86 Emulator-Wikipedia, the free encyclopedia.

[10] Evans, David (2006) "X86 Assembly Guide" University of Virginia. Retrieved 18 November 2010.