RESEARCH ARTICLE OPEN ACCESS

New Approach to Develop a Bilingual Compiler

Shampa Banik*, Md Monjurul Islam**, Md. Azher Uddin***
* (Department of CSE, Chittagong University of Engineering & Technology, Bangladesh)
** (Lecturer, Department of CSE, Chittagong University of Engineering & Technology, Bangladesh)
*** (Lecturer, Department of CSE, International Islamic University Chittagong, Bangladesh)

ABSTRACT
This research work presents a development of a Bangla programming language along with its compiler with an aim to introduce the programming language to the beginner through mother tongue. The syntax and construction of the programming language has been kept similar to BASIC language by considering the fact that BASIC is very easier in terms of its syntax, which is reasonably applicable as an introductory language for new programmer. A compiler has been developed for the proposed programming language that compile the source code into an intermediate code which is optimized.
We have developed our system in Java. Our software is an efficient translation engine which can translate English source code to Bangla source code. We have implemented the system with a lot of test cases to identify what aspects of the system best explain their relative performance.
Keywords **–** Bilingual, Compiler, Lexical, Semantic, Syntax

I. INTRODUCTION

The role of computers in daily life is growing each year. Modern microprocessors are found in cars, microwave ovens, dishwashers, mobile telephones, GPSS navigation systems, video games and personal computers. Each of these devices must be programmed to perform its job. Those programs are written in a formal way with mathematical properties[1] and well-defined meanings known as programming language- rather than a natural language. Programming languages are designed for expressiveness, conciseness and clarity. A program written in a programming language must be translated before it can execute directly on a computer, this translation is accomplished by a software system known as compiler[3]. The task of a compiler is shown in Figure 1.

Figure 1: Compiler input and output

In a traditional compiler, the input is a specified programming language code and the output language is either assembly code or machine code for some computer system. Bilingual Compiler Design has become a highly specialized topic. Bilingual compilers are capable of mixing the written symbols from two or more natural languages in the same document; this includes transition from one symbol set to another in the same paragraph, line or word. The term Bilingual then does not apply to a processor that requires a chip modification to make them speak another language. The idea is to consider compilers as just one instance of translators, broadly, from (almost) any arbitrary source language to (almost) any arbitrary target language.

Target program

II. MOTIVATION OF THE BANGLA LANGUAGE COMPILER

There are many non-English based programming language available worldwide such as Lexico(Spanish based),Aheui(Korean), Sako(Polish), Rapira(Russian) etc. Such languages not only makes available an alternative performance for use by software development community, but also help immensely in popularizing software development among students. Logo and Basic have been converted in many non-English languages.

Lexico is a Spanish language based object-oriented, educational programming language based on .NET Framework. Created to facilitate the programming education, specially object-oriented programming techniques, Lexico has been shown to be successful in introducing Spanish-speaking students to the design of algorithms and data structures. Lexico was created by Pretends and other researchers, who verified that it does indeed help students learn faster and keeps them motivated. They hypothesize that this is due to the fact that students can experiment with algorithm design from the start of their journey in the programming world[4]. However a Kolkata based developer Abhishek Chowdhury has achieved remarkable success by developing a Hindi software development platform named ‘Hindawi’. It can be used to teach children

programming language in their mother tongue, and can be used for serious programming tasks, including systems programming of C, C++, lex, yacc, Java(TM) and assembly. A significant number of non-English languages have been developed to fulfill the aim of learning programming through mother tongue. At the same time, however, compiler design has become a highly specialized topic, and it is not clear that a significant number of Computer Science students will find themselves designing compilers professionally. For developing Bilingual Compiler there are very few research works has been conducted in Bangla programming language field in C programming with its compilers.

III. PROPOSED METHODOLOGY

In this Bangla programming Language every program will be started with the keyword শুরু followed by a colon ':' which declares the main routine. The main routine will be ended by the keyword শেষ followed by a semicolon';'.

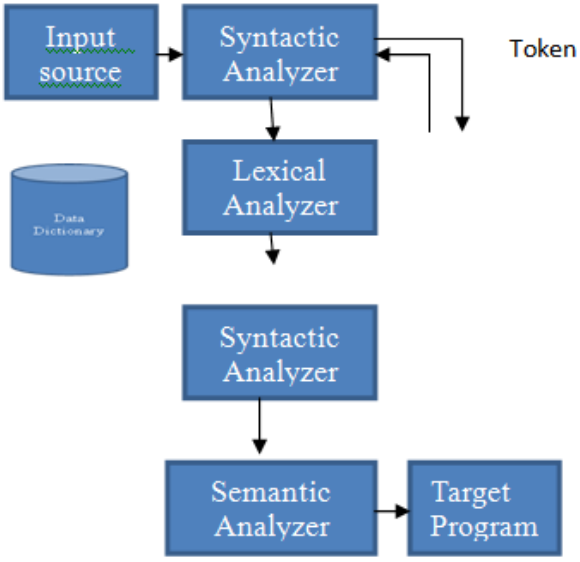


Figure 2: The Overall System Procedure

Every program must have the main routine. Other routines (sub routine) can be written as necessary. No statement can be written outside of any routine. Every subroutine must have a user defined name followed by a colon':'. Subroutines will be ended by the keyword ফের followed by semicolon';'. A sub routine will be called by the keyword ডাক followed by the name of routine. Any other statements should be ended by a semicolon ':'. Multiple statements can be written in one line.

Every grammar should have some lexical specification, list of token using regular expression. In general, there is a set of strings in the input for which the same token is produced as output. This set of strings is described by a rule called a pattern associated with the token.

A lexeme is a sequence of character in the source program that is matched by the pattern for a token. For example, in the Pascal statement

Const pi=3.1416;

The substring pi is a lexeme for the token "identifier".

Again a pattern is a rule describing the set of lexemes that can represent a particular token in source programs. Some examples of tokens with associated patterns and lexemes.

The Lexicon specification of the Bangla programming language by using regular expression is shown in Figure 3.6:

Keywords: শুরু, শেষ, ফের, লিখ, পড়, যদি, বা, যখন, ডাক, ভাঙ
Operators: +, -, *, /, ^, <, >, =
Special operators: ;, // , :
Other token: id, number, str_literal

Digit->[0-9]
Letter-> ক, খ, গ.....
Op->: +, -, *, /, ^
Relop-> <,>
Sign-> +,-
Number->sign?(digit+, digit*.digit+)
Id->letter+(letter, digit,-)*
Str_ltr->"" [^"\n]""
Routine_name -> letter+(letter, digit,_)*

Figure 3: List of regular expression.

THE FRAMEWORK IN JAVA APPLICATION

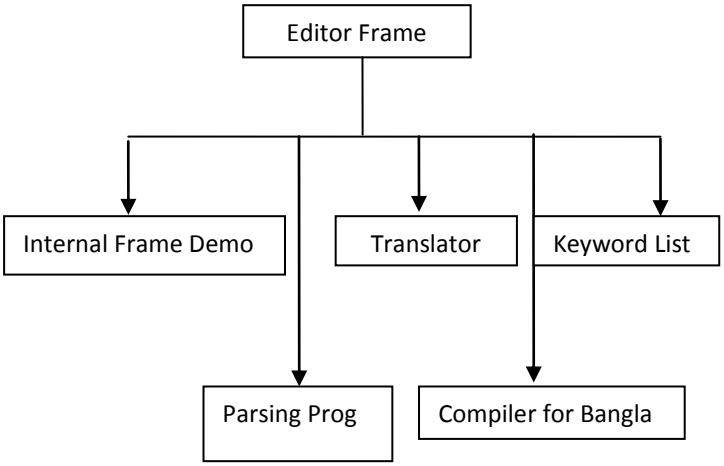
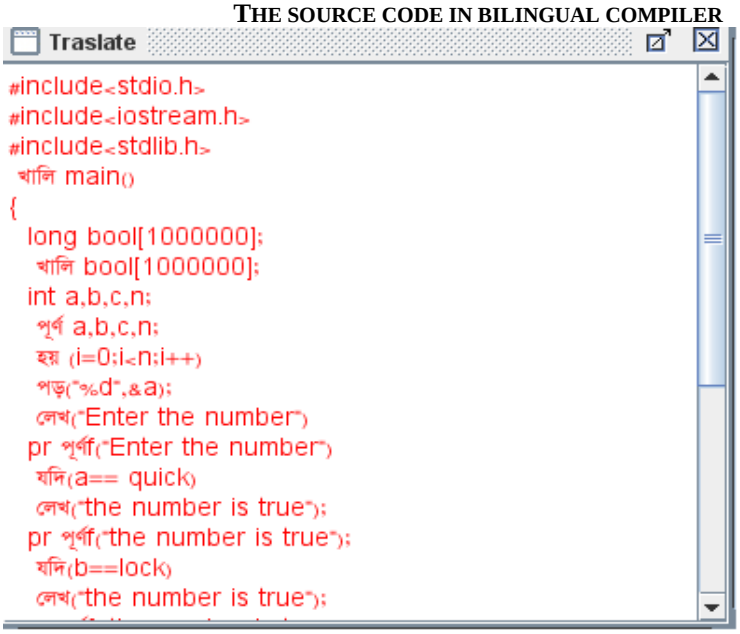


Figure 4.2: Frame implementation in Java

THE SOURCE CODE IN BILINGUAL COMPILER



IV. PERFORMANCE EVALUATION

We have experimented the system over lists of two types named as data type and keywords. We collected various data type and keyword mostly found in our programming code. In our experiments we have used curve for performance measurement of developed Bilingual Compiler. Here, source code with standard translated keywords is compared with other source codes similar translated keywords. To calculate the percentage, the result is then multiplied by 100.

$$\text{Percentage of accuracy} = \left[\frac{\text{Normal translated}}{\text{Standard translated code}} \right] \times 100$$

We have experimented on some source code files and shown below the experimental data in tabular form.

We've experimented the software for the performance test by taking five source code in first experiment for the translated keywords. Here we are representing the percentage of accuracy of the translated keywords of different source codes in a tabular record. Then we've also shown a graphical representation of the record of the first experiment.

Number of translated keywords	Percentage of accuracy
4	44.44%
9	100%
6	80%
5	55.55%
3	33.33%

Table1: Percentage of accuracy calculation for the translated keywords for five source codes

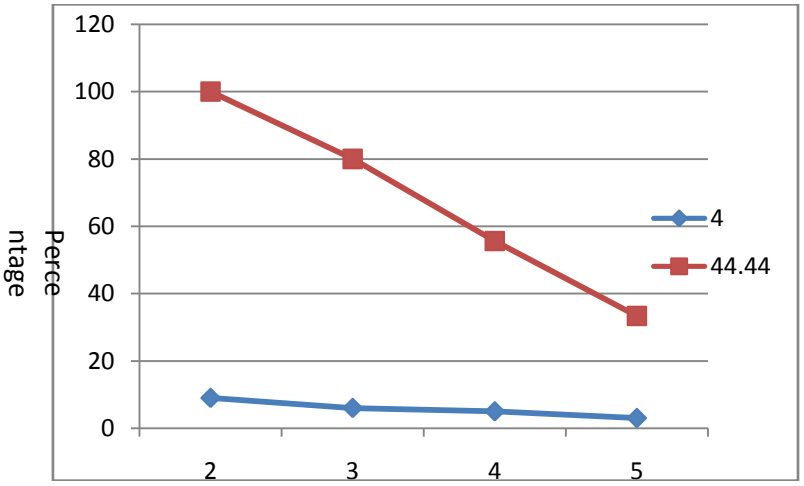


Figure 4: Number of translated keywords Vs percentage of accuracy

In this graphical representation the blue color is showing the percentage of accuracy of the translated form and the red color is showing the experimented translated keywords for five different source codes. We have implemented the system with five source codes containing: 4, 9, 6, 5 and 3 keywords. According to implementation, the percentages of accuracy are: 44.44%, 100%, 80%, 55.55% and 33.33%.

We've also experimented the software for the performance test by taking seven source in first experiment for the translated data types. Here we are representing the percentage of accuracy of the translated data types of different source codes in a tabular record. Then we've also shown a graphical representation of the record of the first experiment.

Number of translated data types	Percentage of accuracy
4	80%
5	100%
3	60%
1	20%
2	40%
1	10%
2	40%

Table 2: Percentage of accuracy calculation for data types of seven source codes

V. CONCLUSION

The idea is to consider compilers as just one instance of translators, broadly, from almost any arbitrary source language to almost any arbitrary target language**.** . We found out the performance of our software using two assumptions.

Professional Compiler Designing is nevertheless applicable to a wide variety of situations.

It turns out, however that many of the techniques and algorithms used by compilers are actually much more broadly applicable than just for translating high-level programming languages to assembly or machine code.

This research work presents a development of a Bangla programming language along with its compiler with an aim to introduce the programming language to the beginner through mother tongue.

The developed system is in a developing stage so, it has some limitations. It can only translate the Cpp files. It has also limitation in finding the Bangla meaning of the long sized keywords and data types. I hope these problems will be solved in near future.

REFERENCES

[1] Md. Abdus Sattar, Rajesh Palit, "*Design and Development of a Bangla Programming Language with Its Compile*r", In Proceedings of National Conference on Computer Processing of Bangla(NCCPB-2004), Independent University, Bangladesh(IUB),27 February, 2004; ISBN:984-32-1112-4 .pp. 72-81.

[2] Allen I.Holub,"*Compiler Design in C*", Prentice-Hall(November 1989).

[3] Alfred V.Aho,Ravi Sethi and Jeffrey D.Ullman, "*Compilers-Principles,Techniques and Tools*" AT&T Bell Laboratories,Murray Hill,New Jersey, 2004/2005.

[4] R. W. Quong, "*Ltoh: a customizable La-TeX to HTML converter*", April 2000

[5] Allen Kent, James G. Williams, "*Bilingual Processors*", user's guide and Reference Manual. Addison-Wesley,2004