

International Journal of Advanced Research in Computer and Communication Engineering

Niral – A Tamil Programming Language

Maheswari M¹, Naveen Bharath P², Rokith K³, Nithish Sangili L⁴, Sriruban K⁵

Assistant Professor, Computer Science and Engineering, Anand Institute of Higher Technology, Chennai, India¹

Student, Computer science and Engineering, Anand Institute of Higher Technology, Chennai, India²⁻⁴

Abstract: In the ever-evolving landscape of technology, proficiency in programming has become a necessary skill, empowering individuals to innovate, problem-solve, and shape the digital world. However, for Tamil-speaking individuals embarking on their journey into the world of coding, the language barrier often poses a significant hurdle. The Languages and tools in the current industry are extraordinary with no doubt, but their main goal is not to be beginner friendly, or rather, in our case easier for Tamil Speakers. Recognizing this challenge, we present "Niral," a programming language crafted to cater to the unique linguistic needs of Tamil-speakers. The motto of this Language is not to replace or be better than the existing tools but to become a better option for beginners who are looking to get into programming. We aim to create a Spark of Curiosity and Ease the transition of Tamil Speakers into the world of Programming. Aligning with our goal, our initial survey of 200 students, showed that beginners who speak Tamil find our syntax easier and more understandable compared to the conventional programming languages.

Keywords: Technology, Students, Programming language, Tamil-speaking individuals, Language barrier, Tools, Beginner-friendly, Linguistic needs.

I. INTRODUCTION

In a world where technology is becoming increasingly ubiquitous, the ability to code has emerged as a crucial skill. However, for many Tamil-speaking individuals, the journey into the realm of programming has often been hindered by language barriers or rather a lower grasp of the concepts as compared to a person with English fluency. Recognising this gap, we present Niral, a programming language designed specifically for Tamil-speaking beginners.

Niral, derived from the Tamil word for "Program" embodies a vision of inclusivity and accessibility in the world of computer science. Its creation stems from a deep understanding of the linguistic and cultural nuances of the Tamil language, aiming to bridge the gap between traditional language barriers and the universal language of code.

At its core, Niral is not intended to compete with established programming languages but rather to serve as a facilitator for those embarking on their coding journey. By aligning its grammar structure with Tamil, Niral offers a familiar and intuitive environment for beginners to explore the fundamentals of programming. This linguistic familiarity is not merely a superficial feature but a fundamental aspect of Niral's design philosophy, enabling Tamil-speaking students to leverage their existing language skills as they dive into the world of coding.

One of the key principles guiding the development of Niral is simplicity. Recognising that many beginners may find the syntax of traditional programming languages daunting, Niral streamlines the coding experience with clear and concise syntax inspired by Tamil grammar. This simplicity not only lowers the barrier to entry but also promotes a deeper understanding of programming concepts, allowing users to focus on problem-solving rather than syntax intricacies.

Moreover, it is our goal to provide comprehensive Tamil documentation and learning resources tailored to the users. These resources provide a structured pathway for learners to progress from novice to proficient programmers. Additionally, community support plays a vital role in the Niral ecosystem. It is an open-source project open to accept contributions from the community, fostering collaboration and knowledge-sharing among Tamil-speaking developers worldwide.

In essence, Niral represents more than just a programming language; it embodies a vision of empowerment and inclusivity in the digital age. By providing a welcoming and accessible platform for Tamil-speaking beginners to learn and explore the art of coding, Niral paves the way for a new generation of diverse and skilled programmers to make their mark on the world stage.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 😤 Peer-reviewed & Refereed journal 😤 Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134206

II. RELATED WORKS

Swaram is a general-purpose, procedural programming language designed to enable programming in Tamil. It was developed as an academic project at PSG College of Technology, Coimbatore, India, by G.R. Prakash and K.K. Ravi Kumar. Swaram aims to be easy to use, platform-independent, and to support Tamil language fonts directly.

It is designed to have specialized support for various aspects of the Tamil language and to be similar to widely used programming languages like C/C++/Java, minimizing the learning curve for those familiar with these languages. Swaram includes a compiler, a virtual machine, and limited APIs for English and Tamil languages

Ezhil: Ezhil is a Tamil-based interpreted procedural programming language developed by Muthiah Annamalai. Ezhil is a Tamil programming language developed to cater to Tamil-speaking novice computer users, offering a unique approach by using Tamil keywords and grammar. It is an interpreted procedural language that simplifies programming for beginners, resembling logical constructs found in English-based programming languages. Ezhil supports imperative programming and a mix of Tamil and English identifiers, making it accessible to K-12 students for early exposure to computer science concepts. With a focus on simplicity and ease of use, Ezhil provides a platform for Tamil speakers to engage in programming, bridging the gap between native language and computer science.

These two are the primary works in this Domain. Swaram, although a full fledged language, it does not follow or resemble Tamil grammar in any way. It is heavily inspired by C syntax and uses Tamil keywords. Unlike Swaram, Ezhil takes a different approach in terms of syntax.

It follows the Tamil grammatical style and borrows the program structure from python with which it is implemented. The Syntax style and structure are extremely subjective but we feel like Ezhil is too far from the Industry standard and migrating from Ezhil to mainstream languages will be harder compared to Niral.

III. LANGUAGE FEATURES OF NIRAL

A. Use of Tamil as Primary Means

ΝM

Tamil language keywords for an Interpreted, dynamic programming language. These keywords can also be written with English characters like – "sol" – meaning: word

B. Allows Mixing English and Tamil language identifiers

The use of both Tamil and English characters is a decision that is taken to facilitate ease of use. We do not differentiate between "sol" and "சொல்" in the program, which stands for String Data Type.

C. Lightweight and Fast interpreter

Implementation of the Interpreter in Go is a choice that is made in-order to create a single lightweight executable that does not have any dependency. This will greatly reduce the time it takes to set up the language in the user's machine.

D. Niral is Open Sourced

Free-software license, which allows wide accessibility and development of the language. Niral will grow as a Language and Community with Open Source contributions.

IV. NIRAL LANGUAGE SYSTEM

The Keywords and Syntax of the Language are primarily designed on the basis of Tamil Grammar. It follows the chain of reasoning followed by the Language. The reason-chain will be the only contrasting difference between Niral and other industry standard programming languages, such that the expression almost always precedes the action or the keyword.

Other than that, the structure of the control flows and loop statements follow the same pattern as C, C++, Java, Go, etc. There are IF-ELSEIF-ELSE, FOR, WHILE, SELECT for the control statements which are standard. The FUNCTION declarations and calls are different from other languages though, details are given below.



International Journal of Advanced Research in Computer and Communication Engineering

```
1 <expression> endral {
2   // code here
3 } illana <expression> {
4   // code here
5 } illana {
6   // code here
7 }
```

Fig. 1 IF-ELSEIF-ELSE control Statement Syntax

| 1 | <expression> endral {</expression> |
|---|---|
| 2 | // code here |
| 3 | <pre>} illana <expression> {</expression></pre> |
| 4 | // code here |
| 5 | } illana { |
| 6 | // code here |
| 7 | } |

Fig. 2 FOR and WHILE loop Statement Syntax

```
1
   <expression> endral {
2
       // code here
3
   } illana <expression> {
       // code here
4
   } illana {
5
       // code here
6
7
   }
   }
8
```

Fig. 3 SELECT Statement Syntax



International Journal of Advanced Research in Computer and Communication Engineering

Fig. 4. FUNCTION Declaration and Call Statement Syntax

Keywords list of Niral is shown in Table 1 below.

TABLE I KEYWORDS LIST

| जळ्ज : yen : int | சொல் : sol : string | என்றால் : endral : if |
|------------------------|---------------------------|-------------------------------|
| இல்லனா : illana : else | முறை : murai : for | வரைக்கும் : varaikkum : while |
| இந்த : indha : switch | செயல் : seiyal : function | சொல்லு : sollu : print |
| கொடு : kodu : input | நீளம் : neelam : length | |

V. SAMPLE PROGRAMS

```
1
   yen n kodu;
2
   yen evenSum = 0;
 3
   yen oddSum = 0;
 4
5
   yen i = 1;
6
   n murai {
7
        i % 2 = 1 endral {
            oddSum = oddSum + i;
8
9
        } illana {
            evenSum = evenSum + i;
10
11
        }
        i = i + 1;
12
13
   }
14
   "Odd Sum: " + oddSum sollu;
15
    "Even Sum: " + evenSum sollu;
16
```

Fig. 5. Odd and Even Sum of Numbers from 1 to N (user input)

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

```
1
   sol name kodu;
2
 3
   yen i = name neelam - 1;
 4
   sol result = "";
 5
   i > 0 varaikkum {
 6
7
        result = result + name[i];
        i = i - 1;
8
9
   }
10
11 result sollu;
```

Fig. 6. Reversing the String given by the User

```
1
   yen prev = 0;
    yen cur = 1;
 2
 3
 4
    yen end kodu;
 5
 6
    end murai {
 7
        cur sollu;
 8
         yen temp = cur;
 9
         cur = cur + prev;
10
        prev = temp;
11
    }
```

Fig. 6. Print the Fibonacci Series up to N numbers

VI. IMPLEMENTATION

The Components of the Niral Interpreter is nothing different from other Language implementations out there, but there are few paradigms that Niral adopts in the Interpretation process that are not the general norms.

Niral Interpreter is written in Go, this choice was made due to the simplistic nature of the Language and the performance of the Language. It is quite easy to implement concurrency with Go, so we took advantage of this aspect of the language. The Diagram of the Architecture of the Niral Interpreter is given below,



The Consumer consumes the program file and produces a stream of Pieces (a.k.a. Tokens). There are these concepts of states in the consumer. Each state consumes some part of the program, throws the piece to the channel for the Generator and then calls the next state. This process of consume and call happens concurrently to the parsing process until the entire program is consumed.

The Generator consumes the stream of Pieces from the channel and then creates the AST object for further evaluation. The concurrent design of these processes are different from other language implementations. Niral currently uses Recursive Descent parsing technique for the parsing process which is fairly simple and powerful.

Niral currently supports all basic programming constructs except for the functions which will soon be implemented too. The Source code for the interpreter along with the binaries are all open sourced and will be available in Github[4]. Niral has a long way to go as a Language. It is currently lacking Documentation and some concepts of programming such as Custom types, Generics, etc, are not implemented. As mentioned earlier the goal of Niral is not to compete with the current arena of Programming languages but to assist beginners trying to learn programming.

VII. CONCLUSION

The potential of Niral as a general purpose programming language in Tamil for anyone trying to learn programming is high and the future is promising. Niral will continue to grow and better documentation and language features that assist beginners will be supported. Niral is an open source project and will stay the same in the future as well. Community contributions are encouraged for the benefit and growth of Niral.

REFERENCES

- [1]. Thorsten Ball (2016). "Writing an Interpreter in Go".
- [2]. Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman (1985). "Compilers: Principles, Techniques, and Tools".
- [3]. Rob Pike (2011). "Go Lexical Scanning".
- [4]. Link to the Implementation https://github.com/iam-naveen/interpreter.