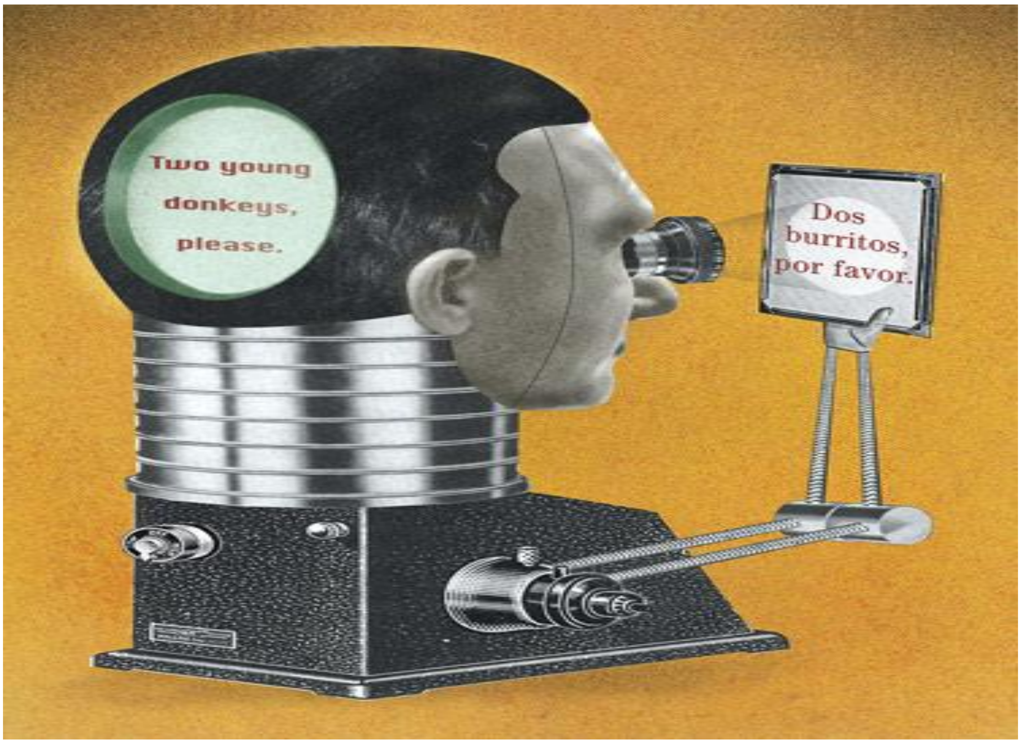
Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the Heart of South Leinster

Occam & C++ Translator



Student Name: Shaoguang Miao

Student ID: C00131017

Supervisor: Joseph Kehoe

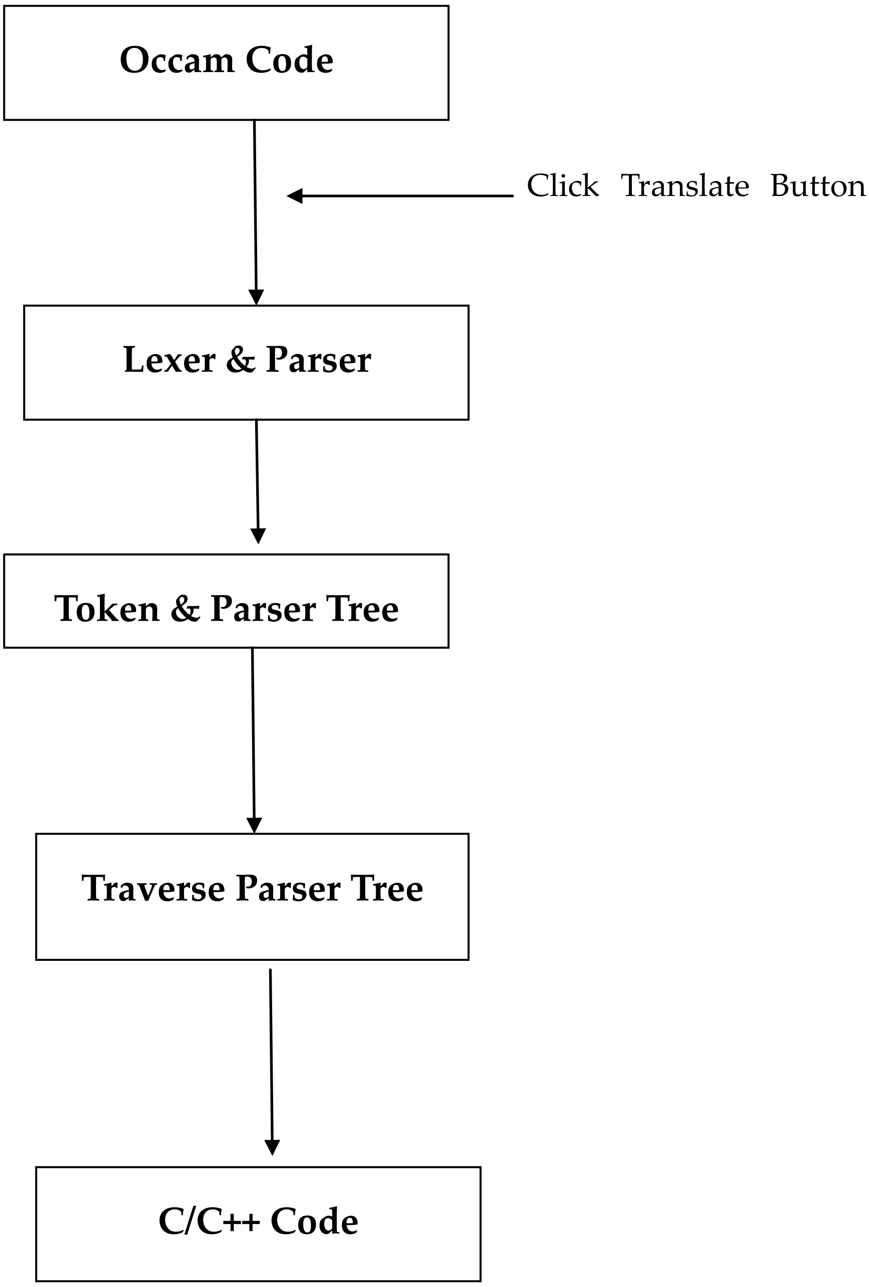
Table of Contents

1. What is the project idea comes from?

Parallel programming is becoming more and more important. However, it's very difficult to write parallel programs using normal programming language, like C, C++, and Java. Communication between two tow threads (processors) is also a big problem using those programming languages. There is an easy way to write a multi-threads program which allows the communication between threads in Occam. Occam is designed for multi-tasks working concurrency. If we can write multi-thread part using Occam and the other part using the other suitable programming language could improve the developing speed and quality. It will give programmer sets of troubles if they make an API by themselves. Thus, building a translator which could translate Occam to target language becomes a suitable idea for all parallel programs. What we expected is that just input some simple Occam code, the software will output the complex C/C++ code for us to implement parallel part.

2. Functionalities

The final project could translate Occam to C/C++ programming language. The most important part is PRA (parallel) and threads communication using channel. How to make the translated code working efficiency is the point of the project. What the project could do will show in Dig.1 below:



Explanation of Functionalities

This project doesn't have lots of essential functionalities, just need an input stream and click translator button, the input stream will be translated to C/C++ code.

The Lexer will recognize the key words which have been defined as tokens. The parser will use those tokens to organize own programmer.

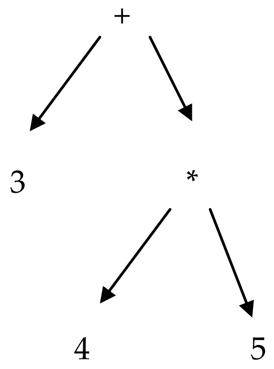
For example:

Example1

a. We defined *'Occam'* as *'C'*.
b. Input a stream *'The Occam is a kind of language'*
c. The Lexer will catch *Occam* from this sentence.
d. The Parser will output *'The C is a kind of language'*.

Example2:

a. We defined the *'number'* as *[0-9]*, '+' as '*operator*', '*' as *'high priority operator'* in lexer
b. Input a stream *'3+4*5'*
c. The Lexer will catch all numbers and operator
d. We defined high priority operator will be execute first.
e. The parser will build a parser tree shown following:



f. The parser tree will be traversed using LNR (left child, node, right child) way.

g. Output will be 23

3. Modular

a. Lexer & Parser

This modular will find the keywords from the program. Before check all the source code, we will define the keywords as token using C\C++ programming language. The program will separate the Occam code to different parts first and then using special way to get the key words of Occam. Like SEQ, FOR, IF, PROC, etc. Each key words will become a small module in the project.

b. Token & Parser Tree

This modular will translate the token to C\C++ code. If it is necessary, we will translate it to Parser Tree which is kind of data type store the token in special ways.

c. Traverse Parser Tree

Traverse the Parser Tree and get the code translated.

For example:

Occam code:

```
IF
    a > b
        c := 3
    a < b
        c : = 4
    TRUE
        SKIP
```

Translate it to C++:

```
    if (a>b)
        c = 3;
    else if a<b
        c = 4;
```

4. Potential user

The C/C++ programmers are always in trouble of parallel programming. Too much complex grammars could make the programmer confusing. However, Occam gets some easier ways to solve this kind of problems and it's very easy to learn. The programmer who wants to use C/C++ to write a huge complex program could learn a little bit Occam and download TBB environment, it could write Occam program first, then translate it into target language. It will get a correct and efficiency target language code instead of writing a complex target language program.