Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

C: Evolution and Features

Vishakha Kapoor, Karanbir Singh Pannu
Dept. of Computer Science, Chandigarh Engineering College,
Punjab Technical University, India

Abstract- This research paper is a concise presentation of the history, evolution and the features of the language 'C'. With the modern day reductions in the hardware costs and corresponding shoot ups in the software costs , we find ourselves in a dilemma that, are the traditional programming languages at par with the modern programming tools . The answer to this question lies in the very brief insight into the features and specifications of a not so old programming language that is "THE LANGUAGE C". 'C' language gives its users an unlimited freedom to develop the code the way they want to provided, they have a little know-how of the specific system for which the code is meant for. So this strange sounding language has come up as the most popular computer languages today because of it being structured ,high-level and machine independent language.

Keywords- Portability, gamma-abstraction, operator precedence, subprograms, dynamic memory management.

I. INTRODUCTION

One of the most powerful, agile, portable and elegantly structured programming language which combines the features of both high level languages and the functional elements of an assembler. YES, we are talking about the mother of all the modern languages, "THE C LANGUAGE ". 'C' provides a decent set of abilities and functional aspects which comprises of the different data types, the structured approach, the concept of pointers, strings, the control structures and the run time libraries which are immensely helpful in dealing with the applications. The 'C' programs are quite efficient and there is very little semantic gap between the language and the computer hardware. Another important feature of this language is the portability across different systems. The popularity of this language shows itself evidently from the many to mention desirable qualities like the set of built in functions and operators which can be used to write any complex program. Its compiler comprises and joins the capabilities of an assembly language with the features of a high level language therefore it is perfectly apt for writing both system software as well as application packages. Some of the programming language features which are going to be discussed in this paper in lieu of the 'C' language are:-

- Procedures: evolved from the notation gamma-abstraction and prevailing in C.
- Recursion: primitive recursion like for loops and general recursion like while loops which evolved from the recursion theory and prevails in C.
- Exceptions: first introduced in PL/1 and formalised in ML. and later similar concepts appeared in C language.

The main power of the C language resides in the variety of data types and the operators that the language provides which makes it considerably faster than other languages of its time like BASIC.
To this date C remains a simple and small language, translatable with simple and small compilers. And for the people with a little knowledge of how the computer works, it is not difficult to develop time and space efficient programs on this coding platform.

II. HISTORY

Developed mainly along the UNIX operating system and the fact that UNIX is entirely coded in C makes it closely related to this OS. It actually came into being in the years 1969-1973 i.e. the early development stages of the UNIX. Then a major breakthrough in the development of this language is marked in the years 1977-1979 concurrent with the testing of portability of the UNIX system. So 'C' shares a close bond with UNIX. C was not completely developed but has gradually evolved from its ancestors like ALGOL, BCPL and B. And this evolution took place, all under DENNIS RITCHIE at the Bell Laboratories. The structured programming legacy once started by ALGOL is carried forward by C in a manner of its own. In the mid 70s a language named BCPL (Basic Combined Programming Language) was developed by Martin Richards which was primarily used for writing the system software. From BCPL, using many features of the aforesaid language, another language called B was developed by Ken Thompson in the year 1970. B was also used to create the primitive versions of the UNIX OS at the Bell Laboratories. Hence the language C as known to us today owes its evolution to BCPL and B. BCPL, B and C , All three of these strongly fit in the typical procedural family typified by FORTRAN and ALGOL. These languages differ syntactically in many details but are similar widely. The programs comprise of a sequence of global declarations and function declarations. BCPL allows nested procedures, but may not refer to non-static objects defined in containing procedures. This practice is altogether banned in B and C. To make sure that C remains a standard, in 1983, AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI) set up a specific technical committee to define a particular standard for this language which was much needed for its existence in the coming years. A version of C was approved by this committee in the year 1989 which is now known as ANSI C.

III. FEATURES (Composition)

- Main aim of any programming language is to help process certain kinds of raw data into meaningful information. Same is the aim of 'C'. Data processing task is accomplished by executing a code made up of a sequence of instructions i.e. the program. These instructions are made up of certain symbols and words governed by grammar or syntax rules specific to C. It has its own vocabulary and grammar. The character set of C language consist of letters, digits, special characters and white spaces. So these variables, constants and keywords made from the legit character sett of C go a long way in forming the instructions of this language which are later processed to obtain the useful information.
- Pre-processor is yet another unique feature of this language which processes the source code before it is passed on to the compiler. It is governed by the pre-processor directives. These tools help make the program easy to read, easy to modify, improves portability and makes them more efficient.
- C also supports a wide variety of built-in operators such as +, -, *, & and < etc. The manipulation of data and variables is done with these very operators and they may be classified as:-

• Arithmetic operators
• Relational Operators
• Logical Operators
• Assignment Operators
• Increment and Decrement
• Conditional Operators
• Bitwise Operators
• Special Operators

All these operators together with the type conversion provided in the C language clubbed with operator precedence and associativity provide a firm foundation for writing good quality code.

- Another important aspect of a language is the reading, processing and writing of data. C manages it pretty well with 'scanf' and 'printf' statements or the direct assignment to variables. The header file <stdio.h> is included in the program to enable all the input/output functions.
- The decision making and branching with the help of IF, IF-ELSE, SWITCH and GOTO statements provide an excellent mechanism to test the conditions and run specific code thereafter. These statements render the work of the user easier by helping making a specific choice and jumping to a specific part of the code.
- Iterations are equally important in any programming language to eliminate the use of writing the same code again and again for a specific task to be performed. For that C allows the use of WHILE, DO and FOR statements.
- 1-D and 2-D arrays provide a strong user-defined data type to store more than one element on it. This powerful data type facilitates efficient storing, accessing and manipulation of data items.
- The division of a C program into functional parts help these independently coded subprograms for subsequent combination to a main program. This makes the task of coding very easy on the whole. Too enable this feature, in C language we have the library functions which need not be written by the user and then, there are the user-defined functions like the main () function.
- For more efficient handling of the arrays and data tables, increasing the execution speed of the code, to support dynamic memory management and to provide an efficient tool to manage dynamic data structures C provides another derived data type known as POINTERS. They are undoubtedly the most powerful feature of C and add a lot of flexibility to the language.
- The console oriented input/output functions always use the terminal (keyboard and screen) as the target place. But this is a problematic approach when large volumes of data are involved. So then comes a more flexible approach of storing data on the disks and restoring whenever necessary. This method uses the concept of files. Some of the basic file operations supported are:-

Naming a file
Opening a file
Reading data from a file
Writing data to a file
Closing a file

The two distinct ways to perform file operations in C are:-

1. Low level I/O using UNIX system calls.
2. High level I/O using functions in C's standard library.

- Dynamic data causes a lot of problem when not properly dealt with. This is that kind of data which keeps on changing even during the execution of the program. Such situations are tackled using the dynamic data

structures in conjunction with dynamic memory management techniques. Linked lists are extensively used for this purpose.

Some of the dynamic storage management functions are malloc (), calloc (), free () and realoc ().

IV. CONCLUSION

This fact is beyond doubt that that UNIX has contributed a lot for the development of this language, as it made the language available to thousands of people. On the other hand, the systems use of C and as a result its portability to a wide variety of machines was a major factor in the systems success. Although the language C, being used by more than a million users on a wide variety of compilers, yet remain stable and unified. The project oriented nature of this compact language could not form a hurdle for its usage ranging from small systems to large supercomputers. This system implementation language is remarkably efficient enough to displace the assembly language and at the same time represents only the essential details and full of fluency to describe the algorithms and its relationship with the various environments.

ACKNOWLEDGEMENT

First of all we would like to thank God Almighty for equipping us with strength and determination to complete this research paper. Faith has been a constant force behind this. Then we would like to show our heartfelt gratitude to our teacher Mrs. Geetkiran Kaur, Assistant professor, Computer Science Department, Chandigarh Engineering College, Landran who has always been there as a guide, mentor, teacher and most of all as a motivator in the entire duration. Her support is always welcome and gracefully acknowledged.

REFERENCES

[1] E.Balagurusamy Programming in ANSI C, 4th ed., 2004.

[2] Pamela Bhattacharya, Lulian Ne Amtiu, "Assessing programming language impact on development and maintenance", department of computer science and engineering, University of California, Riverside, CA, USA.

[3] Feiyi Wang, C Language: Review notes, October 2008.

[4] Dennis M. Ritchie, "The development of C language", Bell Labs/lucent technologies, Murray hill, USA.

[5] Chris Hankin, Hanne Riis Nielson, Jens Palsberg, "Strategic Directions for research on programming languages", ACM workshop on strategic directions in computing research MIT, June 14-15, 1996.

