LECTURE/ STUDY NOTES ON C

PART – I (Overview of C Programming)

Prepared by: *Luther Debbarma, Asstt. Professor, Dept. of IT & Computer Sc, MBB College, Agartala, Tripura.*

Introduction of C Language

C is a general-purpose programming language. Initially, it was thought that C is an exclusively system programming language. But, one can write a wide range of application programs as well in C. Most of the code of operating systems, editors, compilers, word processors, etc is written in C language. C has been one of the most popular programming languages since its early days till today.

C is a structured and procedural programming language. Can you give an example of unstructured programming language? Although it is possible to write object oriented programs in C, it is very difficult to do so and hence we can say that C does not support object oriented programming.

C is a high-level programming language though some authors refer to it as “mid-level” programming language. It must be noted that the syntax, and hence the whole code, of C is somewhat disconcerting to new programmers. Moreover, C is highly flexible in that it is well-suited for system programming as well as application programming.

History of C

C was born at Bell Telephone Laboratories, Inc (popularly referred to as Bell Labs and now part of AT&T). C was developed by Dennis Ritchie in 1972. Interestingly, C was designed and developed to implement the Unix™ Operating System on a then new system, PDP-11 (of Digital Equipment Corporation or simply DEC).

The history of C is therefore very much tied with the history of Unix. In the computer world, the first machines (between the 1940s and the 1950s) ran only one program at a time and only one user could have access to the machine at a given time. In the 1960s, batch systems came into existance. In this scheme, programmers would submit their jobs on punched cards and a batch of many jobs would be read by the computer. This too was unsatisfactory to many users because after submitting a job the programmer had to wait for few hours to get the output. So, to come around this problem, the concept of timesharing was introduced at Darmouth College, ?????? and MIT. After the initial success a better timesharing system called MULTICS was being built by MIT, Bell Labs & General Electric together.

Due to some reasons Bell Labs pulled out of the MULTICS project. One of the researchers of Bell Labs, Ken Thompson wanted to continue working in this line and eventually wrote a simpler version of MULTICS (using assembler) for a discarded PDP-7 minicomputer. Another researcher at Bell Labs, Brian Kernighan, jokingly called the new operating system UNICS (Uniplexed Information and Computing Service). Did you get the joke? If not, UNICS and eunuchs rhyme well. What Brian Kernighan wanted to suggest is that UNICS was a stripped down form of MULTICS. The name, however, stuck. Only the spelling was later changed to Unix.

So, where is C? Well, UNIX was such a huge success that many researchers of Bell Labs joined to improve the Unix. The first problem occurred when Unix was to be ported to newer systems like PDP-11/45 & PDP-11/70. These systems are different and hence Unix had to be re-written for these systems. This is a great problem for the developers. A solution was to use a high-level language instead of assembly language. Ken Thompson did exactly that. The problem was that he used his own designed language called “B”. B was descended from BCPL, BCPL descended from CPL. Now, “B” had many shortcomings due to which Unix could not be rewritten.

At that time, Dennis Richie came to the rescue. He designed a successor to “B”, called “C”. A compiler was written for “C”. Now, Thompson and Ritchie together rewrote Unix in C. Since then, C was refined and presently it is ???????????

Another interesting fact is that Ritchie's compiler produced object code only for the PDP-11. It was Steve Johnson, also of Bell Labs, who improved the work of Ritchie and implemented the portable C compiler. It became easier to produce the object code for almost any machine. Slowly, C begun to be used in universities. But the real explosion of C occurred in 1978 after the publication of a book “The C Programming Language” by Brian Kernighan and Dennis Ritchie.

Since then, several versions of the C Compiler was wrote by different individuals or groups and it also led to problems in compatibility. So, it was required to standardise C. In 1983, ANSI begun its work and finally in 1989 the standard C known as ANSI C was approved. ISO also approved ANSI C in 1990. It was revised again in 1999.

Importance of C

Any college, institute or university has C language as on of the core programming language in its curriculum. Two other languages that find their places in Computer Science & Engineering courses are C++ and Assembly language programming. This is despite the fact that there are numerous languages available and are also being used by several software developers. Only few colleges teach languages such as Java, Visual Basic, etc.

Definitely, then, C must have its advantages over other languages? Well, C language is very well suited for writing system programs/ software like compilers, linkers, drivers and the operating system. On the other hand, C is also used for writing application programs, spreadsheet programs, database management programs, games, etc. In other words, C is highly flexible.

Other than that, C is relatively a small language. It has only 32 keywords. There are only a few basic data types like character , integer and float data types. In fact, there is no built-in input/output facilities. Those are done with the help of functions. Functions, whether library or user-defined, also extend the C language and they also help C programs to be more compact.

Learners and practitioners of C simply love it because it is, in the words of Kernighan and Ritchie, “pleasant, expressive & versatile”. They also write that C “wears well as one's experience with it grows”.

In my point of view, C is very important than any other language because it helps build system software. Both Unix and Windows operating systems are written mostly in C. For e.g., Windows NT 4.0 is written of 16 million lines of code, mostly of C. And, Unix Version 7 consisted of 18,800 lines of C & 2100 lines of assembly language program. In fact, the core software of any computer system is written in C.

Therefore, students are taught and encouraged to learn C. Moreover, once a person becomes familiar with C, learning other languages becomes quite easy. This is the very logic followed by the curriculum framers of Computer Science and Engineering courses. Moreover, reputed software companies also look for graduates with a solid knowledge of C, C++, & assembly language programming. After that, they train the new recruits for few months in the programming language that the respective companies uses for building software.

Demerits of C

I do not know about the demerits of C. It is left as a homework to you students. This is because you are in a better position to complain against C.

When I first learnt C, I disliked C because it seemed a bit cryptic. It is not like the language GW-BASIC I previously learnt in school. GW-BASIC was very much like English in that the commands were like: PRINT, LET, INPUT, END, etc. On the contrary, I was very discouraged at first to see things like: #include <stdio.h>, printf(“%6.2f”, temp), etc. It didn't make much sense at first. However, other than these challenges that every beginner faces, I find no other demerits of C.

Kernighan and Ritchie also points out some of the deficiencies of C: “some of the operators have the wrong precedence, some parts of the syntax could be better.”

Well, other than the ones pointed out above, you are invited to find out the demerits of C.