Helena Huang  
Dr. Nelson  
ATCS: Programming Languages  
10 December, 2013

History of C

Primarily developed between 1969 and 1973 by Dennis M. Ritchie at what was then AT&T Bell Laboratories, C was an imperative structure-oriented programming language designed to be used for the newly created UNIX operating system. It eventually became a widely used language for both professional and leisure programmers.

C’s genealogy traces directly back to the Combined Programming Language. CPL was influenced by ALGOL 60, but because it aimed to allow users to work with both high level programming and individual bits, it was so large and complex that it was unwieldy and never even fully implemented. As a result, Martin Richards developed the typeless language BCPL in the mid-1960s: Basic CPL. It simplified CPL to, as its name implies, the basic features, allowing it to be used for more applications. In 1970, Ken Thompson of Bell Laboratories cut down the language even further, creating the language B (probably a similarly cut down form of the name BCPL), which was specifically for systems programming. Finally, Dennis Ritchie, also working at Bell Laboratories, developed C. The name does not stand for anything, only representing the sequel to B.

Though BCPL, B, and C differ in many aspects, especially syntactically, the basis of each is the same. Programs in each language are made up of global declarations and function declarations. They are all compact, can be translated by simple compilers, need library routines to interact with an operating system (eg. Input-ourput), and are “close to the machine” yet high level enough to facilitate coding efficiency and be portable. They all also recognize separate compilation.

In the 1960s, Bell Telephone Laboratories had been working on a project called Multics with MIT and General Electric, with a goal of creating an operating system for a large computer to be used by a thousand users. Bell Labs withdrew from the project, however, because they believed it would take too much time and too much expense. As an alternative, Ken Thompson and co-workers began developing their own computing system which used many of Multics’s innovations. One major difference, however, between Multics and this new system was that they chose to write in assembly language to conserve memory. At first, they used GEMAP assembler on a GE-635 machine and printed out paper tape, which was then read by the PDP-7. Eventually, the entire project was written for the PDP-7 assembler, which was notable in its extreme simplicity. This computing system was eventually named Unix, as a reference to Multics.

Thompson eventually decided to create a system programming language for his new operating system which would cater to the memory limitations. He therefore created B, a typeless language based on BCPL, but condensed and simplified to fit into 8000 bytes of memory. Most of the semantics remained from the parent language, though much of the syntax changed. Both languages had a single data type called the “word” or “cell,” which was a fixed-length big pattern. Memory was made up of a linear array of these cells, connected by operators. Pointers are integer indices in the memory array. Differences, however, included requiring that an entire program be fed into the compiler at once, in order to offset problems caused by BCPL’s “global vector” mechanism. Additionally, after Thompson finished B, he continually revised it; while additions to the language increased the memory used by the compiler, it increased the memory efficiency of the actual programming. One such addition includes his invention of the increment/decrement operators ++ and --.

In 1972, Dennis M. Ritchie developed the existing B language to his new language, C, which used much of B but introduced data typing and structures. Eventually, most of Unix was rewritten in C, because it was powerful and flexible. Therefore, as Unix spread, so did C, and vice versa, increasing the popularity and usage of both. Also, the compiler was retargeted to other machines. From 1973 to 1980, C further developed, including, for example, the unsigned, long, union, and enumeration types, and an emphasis on structures. Most further development was geared toward increased portability and type safety. After the language achieved portability, its usage dramatically increased, especially shown by the System III and System V versions created by AT&T’s Computer Systems division.

The classic handbook “The C Programming Language, 1st edition,” written by Ritchie and Kernighan and published in 1978, became the standard for the language for years. In 1983, however, an American National Standards Institute committee developed a new standardization, ANSI C, which mostly kept everything the same as the original. The International Standards Organization later adopted this standardization.

Works Cited

CPL. (n.d.). Retrieved December 16, 2013, from Mathematics and Informatics: Bulgarian Academy of Sciences website: http://www.math.bas.bg/~bantchev/place/cpl.html

The C programming language. (1999, December 7). Retrieved December 16, 2013, from University of Michigan-Dearborn website: http://groups.engin.umd.umich.edu/CIS/course.des/cis400/c/c.html

The history of the C language. (2010, July). Retrieved December 16, 2013, from Coding Unit: Programming Tutorials website: http://www.codingunit.com/the-history-of-the-c-language

Ritchie, D. (2013). C history. Retrieved December 16, 2013, from Bell Labs website: http://cm.bell-labs.com/who/dmr/chist.html