

Biography of an Influential Software Engineer

Denis Ritchie

Adam Warde
Student Number: 13328034

November 9, 2017

Introduction

Denis Ritchie was one of the most influential software engineers of all time and also one of the earliest. You'll have a hard time finding a computer scientist who has contributed more to the field than Denis Ritchie.

Ritchie was an American computer Scientist born in New York September 9th 1941. Ritchie grew up in Summit, New Jersey where he attended high school. Ritchie more than likely inherited his interest in maths and science from his father Alistair Ritchie who worked as a scientist for Bell Labs writing the "*The design of Switching Circuits*" on switching circuit theory.

After High School Ritchie went on to attend Harvard where he a degree in physics as well as applied maths. Although Ritchie wrote a PhD Thesis he never recieved a PhD. After finishing in Harvard he following in his fathers foot steps and began working at the Bell Labs Computing Sciences Research Labs.

Contributions to Software Development

While working at Bell Labs Ritchie met Ken Thompson another famous software engineer. Together they worked on an operating system called Multics. Although Multics was eventually pulled from production it was the ground work for Ritchie and Thompsons first big breakthrough, Unix. But by no means were Multics and Unix the same. Multics was far more clunky and put a lot of the strain on the CPU's of the day. This was rectified in Unix which would be more light weight.

Unix was a success with most modern operating systems being somewhat based on it including OS X and Linux distributions. Unix was portable, multiuser and multitasking in a time sharing configuration. It had a hierarchical file system, inter-process communication and a large number of software tools. In Unix the kernel is a piece of software that starts and stops programs, schedules programs on the CPU and manages the file system and memory. The kernel

is used in almost every OS now whether it is monolithic like in Unix or a micro kernel with more distributed control. Unix was seen as suitable for almost any computer and was pivotal in the development of the internet due to its client-server model.

Ritchie also designed a new higher level language to supplement the Operating Systems assembly language. This was called C. C is an imperative programming language. It was designed to be low level with the ability to manipulate individual bits and access memory directly. It was also designed to be simple to compile and hence easily portable. Infact today many languages are compiled into C before assembly. C and its later incarnations (C++,C#) have been extremely influential in the development of software and are still widely used today by many. Ritchie co-wrote *"The C Programming Language"*, a book on C. C was the successor to B a language written by Thompson. However it had many shortcomings which led Ritchie creating C.

Ritchie later went on to work on problems in cryptography including work on a cyphertext-only attack on the M-209 US cipher machine that the NSA asked him not publish for political reasons.

Conclusion

Ritchie's work has not gone unrecognised with him receiving the Turing Award, the Hamming medal from the IEEE and National Medal of Technology from Bill Clinton along with a host of other awards.

His influence on computer science and software today cannot be understated with both C and Unix still influencing design today. Many believe that without Ritchie Apple would never have become as successful as it is today due to OS X being so heavily influenced by Unix.

The open source community is also grateful to Ritchie with most open source products being written and worked on in a linux distribution. Linux is probably the most famous and direct example of an OS to come out of Unix and its creator Linus Torvald also owes his success partly to Ritchie.

At the heart of Ritchie's design philosophy was team work. He was a great believer in team programming with him and Ken Thompson working very closely together almost on a subroutine by subroutine basis. Part of the design philosophy of Unix was to encourage communication between programmers which can be seen in the time sharing features of the OS.

Ritchie retired in 2007 as head of Lucent Technologies System Software's research department. He died in New Jersey in 2011.

To end here is a quote about Ritchie after his death that illustrates his influence in the history of computer science.

"Ritchie, on the other hand, invented and co-invented two key software technologies which make up the DNA of effectively every single computer software product we use directly or even indirectly in the modern age. It sounds like a wild claim, but it really is true."