Biography of a Software Engineer – Dennis Ritchie

Dennis Ritchie was born in 1941, Bronxville, New York. He went on to study in Harvard University graduating with degrees in physics and applied mathematics and PhD in mathematics also. He had major contributions in the field of computing throughout his live but is most well known for the creation of the C programming language and codeveloping the UNIX operating system. These are some of the most influential inventions of the last century and his involvement in both is a tribute to his lasting impact on the field of computer science.

C is the original general-purpose programming language. Its longevity is a testament to its design. It is such a fundamental language that its impact is often overlooked as it is as close to assembly language that some like to get now, but it is a high-level language that was the building blocks upon most of what we now use. The creation of C got off to somewhat of a rocky start. It began life a wild idea funded out of Bell Labs, New Jersey in the 1960s, that tried to tackle issues surrounding terminal sharing this project was known as Multics. Over 5 years AT&T invested 5 million into the project, but it fell behind schedule and the plug was pulled on the project in 1969.

This set-back didn’t discourage all the members involved as 4 men were hell-bent on seeing it through, these were the aforementioned Dennis Ritchie, Ken Thompson, Doug Milroy and J.F. Ossanna. They continued to work on this project handwriting most of their ideas. Ritchie and Thompson developed a game called Space Travel and while in the process on developing this the started implementing some of their ideas about hierarchical filesystems, concepts of computer processes and device files, command-line interpreter, pipes for easy inter-process communication and small utility programmers. Some of which were salvaged from the wreckage of the Multics project. The systems were extremely underpowered compared to the ones they’d previously developed on with the funding of AT&T forcing much of the original specifications were written by hand.

These ideas they were implementing may not seem groundbreaking but were essential in the development of an operating system. The Unix File system is a hierarchy. That is, it can be viewed as a tree structure. Sub directories appear as branches emanating from their parent directories. The tree allows for only one parent for each sub directory, but a parent directory may contain many sub directories. Hierarchical folders have been widely used for managing digital files. A well-constructed hierarchical structure can keep files organized. In computing, a process is the instance of a computer program that is being executed by one or many threads.

A device file is an interface to a device driver that appears in a file system as if it were an ordinary file. These special files allow an application program to interact with a device by using its device driver via standard input/output system calls. Using standard system calls simplifies many programming tasks and leads to consistent user-space I/O mechanisms regardless of device features and functions. A command-line interpreter is applied to computer programs designed to interpret a sequence of lines of text which may be entered by a user, read from a file or another kind of data stream. Extremely widespread in its use today.

A pipeline is a mechanism for inter-process communication using message passing. A pipeline is a set of processes chained together by their standard streams, so that the output text of each process (stdout) is passed directly as input (stdin) to the next one. The second process is started as the first process is still executing, and they are executed concurrently. A key feature of these pipelines is their "hiding of internals" which was later developed by Ritchie & Thompson in 1974.

They worked on a PDP-7 machine which was a minicomputer by the day’s standards, it was first shipped in 1965 and cost $72,000 dollars (equivalent to $591,282 dollars as of 2020), which was considered cheap at the time. This machine weighed in at 500kg and only had a standard memory capable of storing 4,000 words which is the same as 9 kilobytes, this could be upgraded to maximum of 144KB. If we compare this to the iPhone 13 which has 4GB of RAM as standard, the iPhone has 444,444 times the memory of this device in a package that weighs 2874 times less. I think this is important to understand as you start to understand the restrictions theses software engineers were put under and some of the frustration and hardware limitations they had to endure.

Ritchie’s partner Thompson quickly began to work on a Unix language shortly after its development, this language was called B, this was considered too similar to other programming languages at the time most notably its predecessor BCPL. Some of the main issues with this language was that it was typeless and that it was limited to 4 Char strings. This was a good steppingstone to something greater but in its current state B’s assembly language peers were yielding faster program times than possible in B. Ritchie become more involved modifying B adding character types and building his own complier, and within two years a new language was born “C”.

How influential is C? The has been and continues to be powered by C. Projects that exist today written in C have been started decades ago. Oracle database one of the most popular databases in the world started development in 1977 and was rewritten from assembly to c in 1983. Another example of a huge project undertaken with C is Windows. Windows 1.0 was released in 1985 and even though we cannot know for certain what its source code is as it is not been made public It is widely acknowledged that it is mostly written in C with some assembly. This is the world’s most widely used operating system with over 90% market share – developed in C. But its impact is not limited to endeavors undertaken years ago, it is an active language with projects still being started to this day in C. Linux is also written mainly in C; this is commonly used in today’s supercomputers as well as personal computers. The Mac Operating system OS X is also mainly written in C. This means that the 3 most popular OS systems are running on C-powered kernel.

Why is C still used to this day? This can be answered in many ways and will change vastly based upon the functionality needed by the software engineer but here are some of the ways it differentiates itself from other high-level languages.

Efficiency: C is as close to machine code it’s possible to get while maintaining universal availability for existing processor architectures. There is a C compiler for every existing architecture, and it is so highly optimized that it is not easy to improve on efficiency with handwritten assembly code. compilers, libraries, and interpreters of other programming languages are often implemented in C. languages like python use C in their primary implementations. This means that, instead of generating machine code for every architecture to be supported, compilers for those languages just generate intermediate C code, and the C compiler handles the machine code generation.