

Biography of Ken Thompson - **Sean McDonagh 15319517**

Introduction

Ken Thompson was the inventor of the well known Unix operating system, which is the basis for so many other operating systems out there today, such as all Linux distributions. Hence making Prof. Thompson a massively influential Software Engineer.

He was born in 1943 in New Orleans. He received both a Bachelors and Masters degree in electrical engineering from the university of California Berkeley, graduating in 1965 and 1966 respectively.

He began research in the Bell labs in 1966 after completing his Masters degree. This led to an amazing adventure of

- 1) Contributing work to regular expressions.
- 2) Creating a new Programming language (B programming language, a precursor to the programming language C).
- 3) Creating the Unix OS(operating system).
- 4) Contributing to Go programming language

Work of Ken Thompson

Regular Expression

Whilst Ken was working in the Bell labs, in the 1960's, he began his work on regular expressions. He developed a new editor called CTSS, which was an extension of QED. The editor used a thing called regular expressions, to find certain sequences of text within a file/program. Thompsons later created a text editor called ed, which was used by his Unix OS that he created, which he designed to use regular expressions. As Unix was increasingly being used by more and more people, the popularity of regular expressions and searching for text within a string/program greatly increased as well.

To optimise the time taken to find a given sequence of text within a file, he created an algorithm called Thompson's construction algorithm which converted regular expressions into nondeterministic finite automation. Most Regular Expression compilers to date use this algorithm.

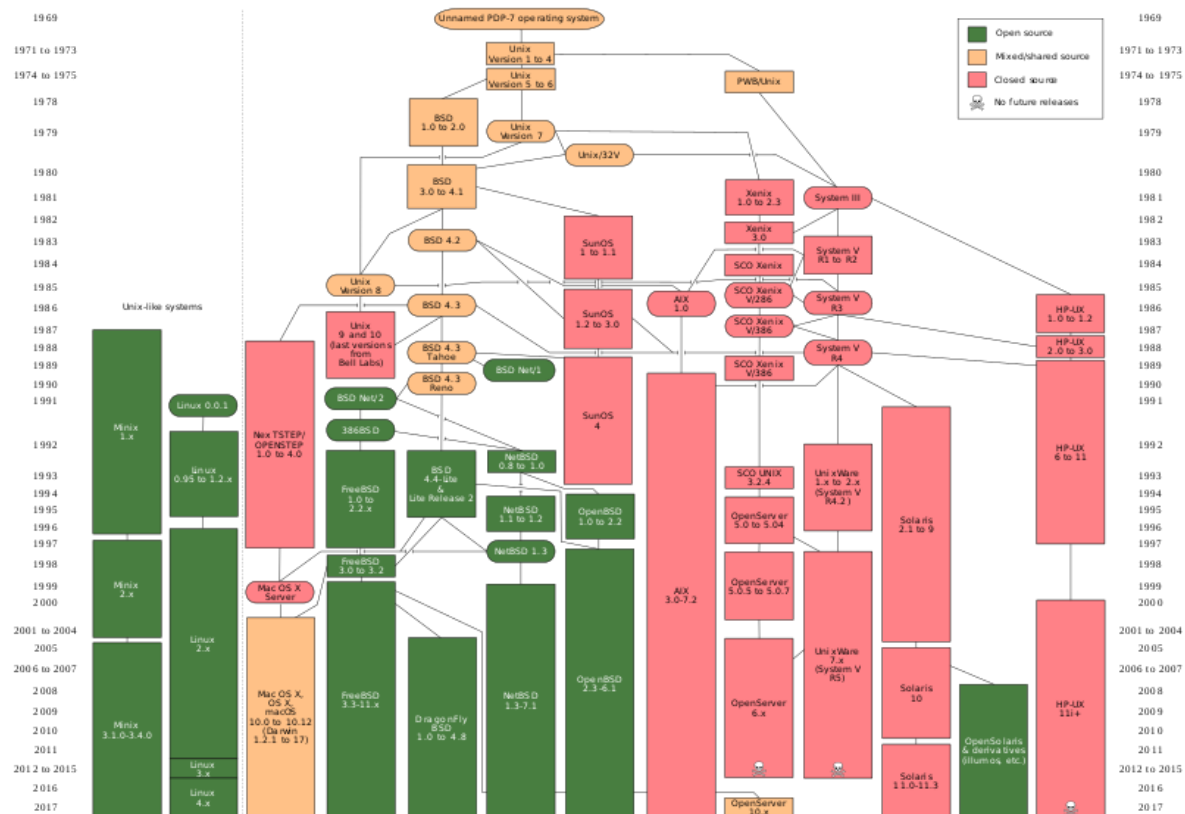
B Programming Language

Ken created B (the first working version created in 1969), alongside Dennis Ritchie (who later wrote the language C), which was initially an effort to make improvements to the language BCPL - (Basic Combined Programming Language). The language C, derives a lot of syntax (for example - `variable++`; `i += 2`;) and other features (memory management) from B. Therefore, it is no doubt that B was a precursor to C.

Undoubtedly, the impact of creating B is phenomenal. It led to the development of C, and hence the development of Unix(whose kernel was written in C), and hence the development of Linux which is used by millions of people worldwide for development purposes.

Unix

After Bell Labs retracted itself from contributing to the MULTICS OS project in 1969, Ken began to write his own OS which is known as the Unnamed PDP-7 computer operating system. The purpose of creating the OS was to create a decent system in order to play his Space Travel game on. The picture below illustrates the timeline from the PDP-7 unnamed OS to the first version of Unix.



The PDP-7 had only an 18 bit word length, and 4kb of memory. This led to the very unique creation of his unnamed OS. The reasons as to why it is unique from other

OS's of it's time is due to its efficient use of memory and it's simplistic design which allowed it to be a great OS parent, hence all of the various OS descendents. All of it's OS descendents drew great advantage from its simplistic design and small usage of memory. A popular OS which does not directly descend from Unix, but is unix like in it's culture and style is Linux (created by Linus Torvalds). Linux replicates Unix' superior use of memory and its simplicity of design.

Then in 1972, Ken rewrote the Unix kernel in C (which had just been created by Dennis Ritchie at the time). This is what guaranteed the success of Unix. Writing it in C allowed the OS to be portable, i.e it allowed UNIX to be useable on various computer systems, with differing sizes of memory and different processor architectures. This was a break in the trend. Usually OS's would have been invented with a particular Computer/processor architecture in mind and hence be bounded by this intended Computer/processor architecture, and would only be able to run on such an architecture. The Unix OS could be ran on any processor architecture, and hence established a new era of computing.

Go

Jumping now into the year 2000, Ken leaves the Bell Labs. Come 2006, Ken begins working in Google as a Distinguished Engineer. During his time at Google (He retired in 2014), his work included the co-design of the [Go](#) programming language. His precise contribution, is the creation of the compiler for Go. Go's compiler is known for its incredible speed of compilation of complex programs, and its multi-level assembly code optimisation.

In an interview, which is referenced by Dr.Dobbs website, he States that "Yes. When the three of us [Thompson, Rob Pike, and Robert Griesemer] got started, it was pure research. The three of us got together and decided that we hated C++. [laughter]".

The reason why He hates C++ is because of its unpredictability. Hence his desire in creating Go was to create an efficient language in terms of time of compilation but also a language that would maintain stability and predictability when code bases become very large and complex (which he argues does not happen with C++), which is what happens in companies like Google.

Conclusion/ Impact of work

In conclusion, the work of Thompson has greatly impacted the world, and more specifically the world of software development.

Regular expressions have had a massive impact on the world of software development, by decreasing the amount of time taken to search for a given text within a file. Also, he created a standard syntax for regular expression (Thompson's syntax), which gives a generic way for carrying out pattern matching.

His work on creating Unix is by far his most impactful piece of work to date. Without the creation of the Unix OS, then the Linux we use today would (probably) not exist, and thus the most preferred OS to develop software in would (probably) no longer exist.

Finally, Go is another incredible contribution by Ken (and Glen Pike and other people working at Google). The language is having a massive impact on the speed of development, scalability of software (which is one thing that Google always concerns itself with, considering the amount of end users it has at any one moment in time), by having clean syntax, great garbage collection, good memory management and quick code compilation time.

References

- 1) <http://www.computerhistory.org/fellowawards/hall/ken-thompson/>
- 2) <http://www.linfo.org/thompson.html>
- 3) https://en.wikipedia.org/wiki/Ken_Thompson
- 4) <https://en.wikipedia.org/wiki/Unix>