CSU33012 Software Engineering

Biography of a Software Engineer

Aodhán Keane Student ID: 19334599

Introduction

This is a short biography about Linus Torvalds. Topics of discussion include his early programming ability, as well as his work, the impact he had on software engineering as a practice, and finally, my own reflection on his story.

Linus Torvalds - Software Engineer in the making

Linus Torvalds was born in 1969, in Finland. It was clear in his youth that he was a formidable programmer. By age 11, he was already learning computer programming on his grandfather's Commodore VIC-20 computer, in BASIC. Years later he bought his own computer, for which he wrote his own assembler, editor, and various games. He also tweaked its operating system. "All the software was off in Britain" he said years later, in an interview with Jeremy Allision. "You couldn't buy anything in Finland". No choice but to make it himself!

He started attending the University of Helsinki in 1988. During his studies, he encountered the MINIX operating system, which he preferred to the MS-DOS operating system that was on his computer. However, certain aspects of the operating system were not to his taste. This prompted his work on his own operating system, "Linux".

In 1996 he graduated with a degree in Computer Science. His master's thesis was titled "Linux: A Portable Operating System".

Torvalds' Work

Linux

Linus Torvalds' work on Linux began long ago.

He began developing Linux in 1991, while he was still attending university. He sought to augment the Unix-based MINIX operating system he had used in his studies. He later made his code public, seeking opinions on its strengths and flaws. Because it was open source, any interested programmer could have a look. There soon grew a team of eager developers all contributing to the Linux kernel.

Also in 1991, Torvalds attended a talk given by Richard Stallman, which was Torvalds' first exposure to the GNU Project, which strives for free software and the rights of users to study, alter or share it. He would go on to make other future projects, such as the Git source control utility, and his scuba diving software “Subsurface”, open source. He ended up using the GPLv2 license for Linux in 1992.

With Linux’s rapid refinement under Torvald’s supervision, and the voluntary efforts of its developers, it wasn’t long before it gained significant traction. Large corporations like Intel and Oracle began Linux support on systems, as a secondary choice to Microsoft’s Windows. Estimates have it that by 1999, Linux was running on seven million computers. And it was, and still is free!

After some years spent working at Transmeta, Torvalds’ moved over to the Open Source Development Labs, while continuing to administrate Linux kernel development. This organization eventually merged with the nonprofit Free Standards Group. The resulting organization was called the “Linux Foundation”, and to this day, Torvalds gets the final say on what code changes are implemented into the Linux Kernel.

Git

Until 2004, Linux kernel developers had been using a closed-source revision control system called “Bitkeeper”, run by Larry McVoy. However, after developer Andrew Tridgell attempted reverse-engineering it, so he could create a free software system for the Linux community to use instead, McVoy terminated the Bitkeeper service for development of the Linux kernel, because such action was forbidden.

Lacking any acceptable open-source alternatives, Torvalds put his kernel work on hold, and spent over a week designing “Git”, his own free, open-source version control system. Within weeks of its initial design, Git could maintain version control for itself, and after a few months, it matured to full functionality.

Git was a significant improvement over Bitkeeper and other similar systems. It was much faster, because speed was of paramount importance when Torvalds designed it. Six months’ work was done by Torvalds on Git, after which he handed the role of its maintenance to Junio C. Hamano. Torvalds continued from then onwards with kernel development.

His impact on software engineering

Linus Torvalds’ contributions to the discipline of software engineering are difficult to fully appreciate. His work has permanently influenced the way software is designed and maintained. Linux and Git, being so popular, are also fantastic examples of open-source software attaining success, and have likely been the reason so much other software is kept open source, too.

The top 500 supercomputers in the world all run Torvalds’ Linux operating system, as do the vast majority of servers. Many smartphones run some form of Linux, too. This is a testament to the flexibility the Linux operating system provides to developers engineering complex software systems.

Its customizability and versatility are unmatched, and it has enabled software engineers to design in a way that other privately owned, closed-source systems such as Windows, do not. By granting software engineers more control over the environment their software runs in, it has granted them an extra means of achieving their goals. His work on Linux has, without question, changed the way software is designed forever.

Similarly influential, Git has since revolutionized version control, an integral part of the software developing experience. It was the first tool of its kind to allow local functionality, instead of having to connect to a server. This eased and sped up the development process considerably, because actions like branching could happen extremely quickly on one's local machine, instead of having to do so online.

It also facilitates quick experiments in local branches, which can be added to a main branch later, instead of having to try something out in a branch in the repository, which developers may hesitate to do. Its ubiquity is unparalleled, and is proof of Linus Torvalds' extraordinary influence on how modern software is engineered and maintained.

Inspiration

What amazed me during my research for this biography was just how much of an impact one individual could have one the world with their trusty keyboard. Few other inventions, if any, have ever landed one man's ideas in the minds and devices of billions of people. Software is everywhere these days, and it is here to stay for as long as I'll be around. A valuable lesson I've learned is that, with so much potential influence, it is essential that I, too, responsibly engineer decent software.

Linus Torvalds' motivation for Linux and Git was his dissatisfaction with the software available at the time, and through careful collaboration, he spawned two of the most successful software systems ever made. In this regard, I sincerely admire Torvalds, and I hope that in the future, I will also be writing software of a similar caliber, for the better of all who use it.

However, I do not look to be as aggressive as Torvalds in my critique of those with a different vision. In my research, I found various articles reporting on Torvalds and his infamously offensive lambasting of other developers. I was shocked and appalled at the abuse and curses he hurled at those who submitted code not to his liking. He has since apologized for his remarks, but what I read has stuck with me, and cemented my understanding of Linus Torvalds as an unsavory character.

So far in my education at Trinity College, all the programming teams I have been a part of have, thankfully, been successful, all the while maintaining friendly, mannered collaboration, even through our disagreements. It is obvious to me that this is the best way to conduct software engineering teams, because a happy team makes for a more productive team. Being mean to people who are only trying to do their part in improving anything in life, such as software systems, only dampens the likelihood of them doing so again. This is an attitude I will do my best to avoid inheriting.

My study of Linus Torvalds and his software engineering career taught me a lot. I have learned what I should strive to be as a software engineer, as well as what I would do well to avoid. I will, without a doubt, remember these lessons with me in the future.

Sources

https://en.wikipedia.org/wiki/Linus_Torvalds

https://www.internethalloffame.org/inductees/linus-torvalds

http://www.computinghistory.org.uk/det/1790/Linus-Torvalds/

https://computerhistory.org/profile/linus-torvalds/

https://www.youtube.com/watch?v=05pgVwzAZ6k&t=46s

https://blog.storagecraft.com/linux-history-linus-torvalds/

https://pandorafms.com/blog/linus-torvalds/

https://www.tag1consulting.com/blog/interview-linus-torvalds-linux-and-git

https://www.techrepublic.com/article/linus-torvalds-git-proved-i-could-be-more-than-a-one-hit-wonder/

https://www.linuxfoundation.org/blog/10-years-of-git-an-interview-with-git-creator-linus-torvalds/

https://www.linuxjournal.com/content/git-origin-story

https://www.tag1consulting.com/blog/interview-linus-torvalds-linux-and-git

https://blog.codacy.com/the-impact-of-git-on-software-development/

https://www.wired.com/story/linuxs-creator-is-sorry-but-will-he-change/

https://www.newyorker.com/science/elements/after-years-of-abusive-e-mails-the-creator-of-linux-steps-aside