Dennis Ritchie –

Unix was developed by programmers at Bell labs, who included Dennis Ritchie and Ken Thompson, as an operating system for mini computers – the precursors to personal computers. The key component of Unix was the kernel, or master control program. **Although this operating system may not be recognizable to most people, Unix or systems that are derived from it or are compatible with it, like Linux, are ubiquitous in modern technology such as smart phones and personal computers.** As Dennis Ritchie is one of the creators of such a pervasive system as Unix, it is interesting to hear his take on the future of computers, namely that they will continue to shrink in size and will eventually be found embedded in clothing and other unexpected places.

Bjarne Stroustrup –

Bjarne Stroustrup briefly described how programmers went from writing code directly to hardware using bits and bytes to writing code that was closer to human language. The early coding languages, like Fortran and Cobalt, were written to meet specialized needs but weren’t quite able to meet more universal needs. So, in the ‘60s a group of Norwegians created Simula, a language that was intended to be able to meet all needs. This program first introduced the concept of class in programming languages. Classes were used to represent concepts within applications. These classes could be organized hierarchically into what is known as object-oriented programming, or data abstraction. Bjarne Stroustrup decided to take these concepts and combine them with C, which was an easy-to-understand (for humans) programming language created by Dennis Ritchie at Bell Labs. Thus was created C++.

Bjarne Stroustrup argues that the strength of C++ lies in its ability to efficiently use abstractions and provide a direct interface with hardware, thereby creating application infrastructures that are stable and reliable for long periods of time. Being able to maintain one’s programs for decades without fail and without having to struggle to find someone who knows the language is a savings of money and effort for any company.

The Unix Operating System –

A large programming project can be daunting, but a good operating environment can make it easier. Because of constant demands for improvements and upgrades to software, Bell labs focused on making their software change tolerant. To do this they first ensured that programs were written to be clear and easy to read by using proper programming structure. They also created each piece of software by combining many smaller modules, which allows necessary changes to be made to only to the relevant modules rather than to the entire piece of software. When Dennis Ritchie and Ken Thompson got together to create the Unix operating system, they were trying to create a simple system that worked with only a few “primitives”. There are three main parts that make up the Unix system. The kernel controls the hardware resources. The shell is the interface between the users and the kernel. The utilities are useful programs that the users use. The thing that made Unix special is that programs could be used together in very flexible ways by means of pipelines. Doing this made it possible to do complicated tasks and make improvements on those tasks without having to write new code. The programming community also would dispense these improvements to the entire community, allowing for improvements to be built upon each other. They also created pattern matching algorithms that helped improve programs. The very simple and hierarchical file system also made the Unix system superior since you could save information and retrieve it later. The things that could be filed included commands, which meant that users could execute a series of commands that would run designated programs, all with one written word. This made using a computer much easier and much more efficient, while also saving time. While making Unix, the developers created the high-level C programming language to make the system easier to use, although the system is so friendly that other languages work just fine on it. Unix was particularly helpful for creating complicated circuit designs.

Linus Torvalds –