CSNB113: System Administration
-
3rd Topic: The need for an operating system // history of operating systems

Remember: Booting - III

As soon as a MBR is found on one of devices scanned by the BIOS, the control is passed to this device, and the location of the operating system is 'navigated' to. (A MBR essentially is a pointer to a specific location/address on the specific disk drive.)

- At that location on the drive, a **basic program loader** needs to be found.
  If this is not the case, the boot process fails
- This **program loader** then loads the **kernel** of the operating system.
- The kernel in turn loads the **operating system** and its applications.

Simple loader

A simple **program loader** could also be used, and was used in the early days of computers, just to load a program and run it.
DOS (Disk Operating System) is actually a program loader.

This method works, but not too well:

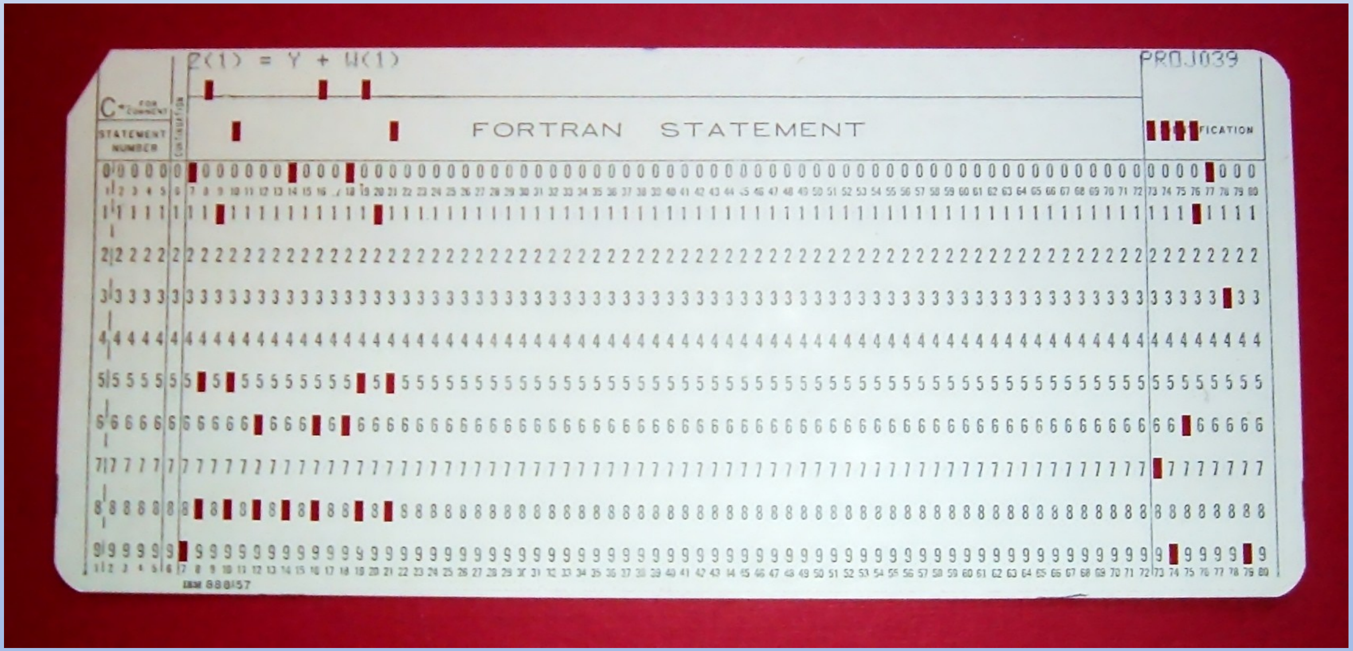
- Then **only one single program** can run at a time. One cannot open more than one 'window',
- One cannot use more than one application
- Only **one user** can use the machine, and if (s)he does not use it, it will have to remain idle
- When this program or application gets 'stuck', the machine is 'dead' and needs a 'hard reset'

That is why the concept of '**operating system**' was invented:

- A **kernel** to allocate and control resources
- A system to control the **running of programs**, starting and stopping them
- A number of **system programs** to calculate time, manage users, manage printers, manage the display, manage the network connections, notify users, etc.

Operating Systems: Batch processing

In the very beginning, the task to run more than one program was done by running strictly one after the other, so-called **Batch processing**. The programs were read from **punchcards**:



Operating Systems: Multics and Unix

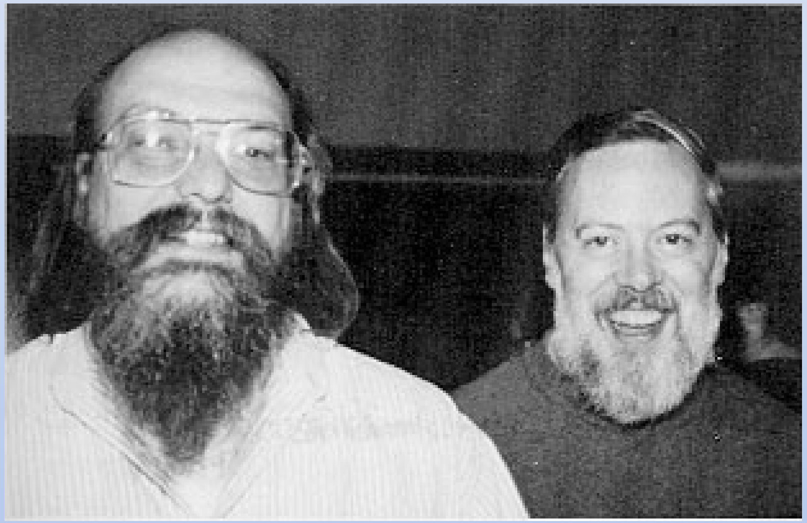
Later, at Bell Labs (AT&T), it was planned to have a system for **multiple applications** for **multiple users**.

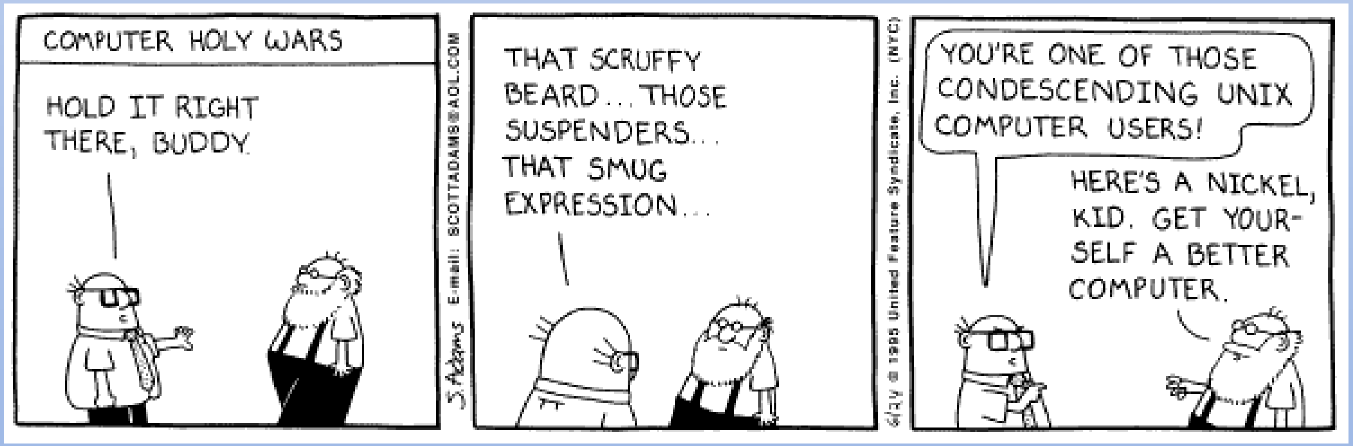
(Read "What's In A Name" on p.3 of the textbook for the details.)

The result was '**Unix**'. It ran for the first time in 1969, developed by **Ken Thompson** and **Dennis Ritchie**. Both also developed the **programming language C** around this time.

Unix was a huge success, many companies paid for the licenses, produced their own versions, and own **utilities**.

Ken and Dennis (and Dilbert)




COMPUTER HOLY WARS
HOLD IT RIGHT THERE, BUDDY.
S. Adams E-mail: SCOTTADAMS@AOL.COM
THAT SCRUFFY BEARD... THOSE SUSPENDERS... THAT SMUG EXPRESSION...
6/24 © 1995 United Feature Syndicate, Inc. (NYC)
YOU'RE ONE OF THOSE CONDESCENDING UNIX COMPUTER USERS!
HERE'S A NICKEL, KID. GET YOURSELF A BETTER COMPUTER.

Commercial Success

Unix was a big success, AT&T cashed a lot of licensing fees.

For business this was okay.

For academia it was not:

- Only rich universities and schools could afford the **licensing fees**
- Often, a program needs some modification in a research environment. But the **program sources** were usually not included, so modifications could not be done by the users, the academicians, the researchers

Free Software Foundation

Richard Stallman, a researcher in the Artificial Intelligence Lab of Massachusetts Institute of Technology (MIT), was convinced that **sharing** software was helpful, and **changing** it was necessary for continuous progress.

He suggested to re-write all **proprietary** Unix utilities under a **license** that allows to exchange, modify and re-distribute the source code freely.

He founded the **Free Software Foundation** (FSF) for this purpose in 1985.

Free Software

The **Free Software Foundation** developed **four essential freedoms**:

- The freedom to **run** the program, for any purpose (freedom 0).
- The freedom to **study** how the program works, and **change** it to make it do what you wish (freedom 1).
- The freedom to **redistribute** copies so you can help your neighbor (freedom 2).
- The freedom to **distribute copies of your modified versions** to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes.

Access to the source code is a precondition for freedom 1 and freedom 3.

Successes and Failures

The **Free Software Foundation** achieved its purposes, most of them. A **compiler** became available, an **editor**, a **shell**, and all other tools and utilities.

Everything was licensed under a license that gives the four freedoms to all users. But nobody is allowed to remove this license from the software that they distribute or sell. So these 4 freedoms remain with the software forever.

The license is called **General Public License** (GPL)

Failure?

The 'failure' was the lack of an operating system **kernel**. The **tools** all available under GPL, it still needed a **proprietary** license for a kernel.

Linus and Linux

The problem 'kernel' was solved in an unexpected manner. A student at University of Helsinki in Finland wanted to use the power of the **Unix-shell** on his Personal Computer at home. (In the 1980-s affordable computers had been introduced by IBM and trademarked as **PC**s. They were almost all running **DOS** - Disk Operating System, in principle a program loader with a basic **command prompt** tool.)

Linus Torvalds, that student, could use all the utilities offered under **GPL**, but the kernel was missing. So he took the initiative to write one himself, and had it published under the same license.

Finally, a **complete operating system** with kernel and utilities was available under a Free Software license. Since most Unix versions had been named with some -ix or similar at the end, like AIX, HP-UX, IRIX, Minix, Solaris, it was no surprise that this new one was called **Linux**.

Linux and GPL-Distributions

Linus (Linux) actually being the kernel 'only', the complete and correct picture of a complete operating system should include the **GNU** (GNU is Not Unix – read "What's In A Name" on p.16 of the textbook for the details.)

Since both kernel and GNU utilities are available as Free Software, many organisations, companies and individuals started to distribute their preferred sets of programs as **Linux Distributions**. The most famous are Red Hat, Suse, Ubuntu. Others are Debian, Gentoo, Slackware.

Linus and GNU/Linux

References

- http://en.wikipedia.org/wiki/Computer_programming_in_the_punched_card_era
- Textbook, p.3
- Textbook, p.16
- http://www.fsf.org/
- http://en.wikipedia.org/wiki/Linus_Torvalds