CS 107
Lecture 26: A short history of UNIX and Linux

Monday, March 11, 2024

Computer Systems
Winter 2024
Stanford University
Computer Science Department

Reading: Course Reader: x86-64 Assembly Language, Textbook: Chapter 3.1-3.4

Lecturer: Chris Gregg
• Assignment 6, due Thursday
• Final exam prep materials have been released
  • Generics will be on the exam, so look at the midterm questions that have that material
• Final exam:
  • Monday, March 18th, 3:30pm-6:30pm, CEMEX auditorium
• Today's topics:
  • The history of UNIX and Linux
"Since heap allocator is our capstone project, I also believe that it should be our final. It is an extremely quick turnaround to submit heap allocator on Friday night and then start studying for a final on Monday afternoon. Additionally, many students (including myself) have multiple finals early during finals week, so we have to dedicate time for studying for those classes too. I believe that for a 5 unit class, it is sufficient for heap allocator to count as our final project. On a separate note, binary bomb took me over 60 hours to complete. I know that I am not alone in this, as I have talked to my classmates and it also took them a long time to complete. While the assignment accomplishes its goal of teaching us assembly (I truly did learn a lot) maybe consider only making it three levels. I think we would learn just as much by removing the second or third level and keeping the fourth as a challenge."

1. We're having a final because I want to ensure that you've learned the material. The assignments are great, but there are a lot of students who make it through the assignments without really learning the material.
2. Yes, I know you have multiple exams. Unfortunately, that's one of the challenges of college.
3. Binary Bomb was, until a few years ago, five levels.

Final thoughts: this is a hard class! You've done great — hopefully the frustration is/was worth it!
A history of UNIX and Linux
For most of the 20th century, the research lab to be associated with was Bell Labs. Researchers at Bell Labs had the following discoveries and inventions:

- Discovered cosmic background radiation
- Invented the transistor
- Invented the laser
- Invented the photovoltaic cell
- Invented the charge-coupled device (CCD) (the image sensor in your phone's digital camera)
- Made significant contributions to information theory

**Invented the Unix operating system**

- Invented the following programming languages:
  - B, C, C++, S, SNOBOL, AWK, AMPL (among others)

- There have been 10 Nobel prizes awarded for work done at Bell Labs
The PDP11

• Needless to say, Bell has a storied history, and there are lots of videos on YouTube about it (examples: 1, 2, 3)

• Two employees of Bell Labs, Dennis Ritchie and Ken Thompson, decided in 1969 to create an operating system that could be used on a minicomputer.
  • They initially planned the operating system for Bell Labs use
  • The "minicomputers" they ran it on were 16-bit machines designed in the 1960s. One of the most prominent minicomputers was the Digital Equipment Corporation's PDP-11 minicomputer
    • They were not cheap: the first one cost $20,000 ($160K today)
    • They were not powerful: 16-bits, and the early ones had 4KB of memory (!). They eventually had 1-2MB of memory, and between 10MB and 500MB hard drives (your computer may have a 1TB solid state drive, with 2000 times more space).
    • There were not that "mini" — they weighed more than 500lbs (oh, and that hard drive weighed another 300lbs)
  • The computer was usually (through the mid-1970s) interacted with through a "teletype machine," not a keyboard and monitor.
The Unix Philosophy

- Ken Thompson (left) and Dennis Ritchie (right) decided that Unix would have a set of basic rules, known as "the Unix Philosophy."
- "The Unix philosophy emphasizes building simple, compact, clear, modular, and extensible code that can be easily maintained and repurposed by developers other than its creators." (Wikipedia, The Unix Philosophy)
- The Unix philosophy is documented by Doug McIlroy in the Bell System Technical Journal from 1978:
  - Make each program do one thing well. To do a new job, build afresh rather than complicate old programs by adding new "features".
  - Expect the output of every program to become the input to another, as yet unknown, program. Don't clutter output with extraneous information. Avoid stringently columnar or binary input formats. Don't insist on interactive input.
  - Design and build software, even operating systems, to be tried early, ideally within weeks. Don't hesitate to throw away the clumsy parts and rebuild them.
  - Use tools in preference to unskilled help to lighten a programming task, even if you have to detour to build the tools and expect to throw some of them out after you've finished using them.
Many of the basic ideas that Thompson and Ritchie came up with for Unix are still used in today's Linux operating systems.

- The following commands are still used today (and `cd` was originally `chdir`):
  - `ls`, `mkdir`, `rmdir`, `sort`, `who`, `echo`, `find`, `cat`, `chmod`, `exit`, `rm`, `date`, `dc`, `ed`, `mail`, `wc`

- The idea of "pipelining" came along in an early version of Unix
  - Users can create modular chains of producer-consumer programs to do various tasks (you've seen this in some of the tests for our programs, e.g., `sort samples/names | uniq -c | sort -n | ./mytail`
  
- When you have basic utilities, you can do some powerful things with them, and we can still do those today — the video above has an example with numbers, and I can do that on my Mac today:
  ```
  echo "2^100" | bc | number | say
  ```
Unix Expands Outside Bell Labs

• The original Bell Labs Unix was completely open source, though Bell (by then, AT&T) sold licenses for others to use it.
• In 1974, UC Berkeley got a copy of the source, and installed it onto a PDP11 that was bought to run the system (though they had to share the computer with the math and statistics departments, who ran a different OS, so Unix only ran for 8 hours per day or night).
• In 1975, Ken Thompson took a sabbatical from Bell Labs to go to Berkeley to continue work on the OS.

By 1977, Berkeley had their own version of Unix, called "Berkeley Software Distribution" Unix, which Bill Joy led (Joy later founded Sun Microsystems, and created vi, the precursor to vim. Joy was actually the 16th employee at Sun, though he was given founder status, as Sun was an offshoot of a Stanford student project).

• Other universities wanted copies, so Berkeley sent them out, and kept improving on their version.
• BSD kept improving, with new releases over the years, and eventually had versions superior in many ways than AT&T's versions. The licenses also caused some legal trouble in the 1980s and 1990s.
• BSD Unix had many descendants, including Sun's OSs, NeXT's OS, FreeBSD, and Darwin, which is the core of the MacOS system. Yes, the Mac operating system is a true Unix OS.
Unix Standardization

- In the late 1980s, the IEEE decided to create standards for operating systems to maintain compatibility. This was called the *Portable Operating System Interface*, or POSIX. Unix was selected as the basis for the system, which includes the Standard C library (all of the functions we have come to love this quarter!)

- Unix versions can be granted "POSIX certification," and MacOS is one of the few big names that has this designation. Linux and Android are listed as "mostly POSIX-compliant."

- Unix was originally written in assembly language, but in 1973, it was re-written (by Dennis Ritchie) in the C language, making it much more portable (though there are still many low-level parts of all OSs that must still be written in assembly)
The Birth of Linux

- In 1984, Unix was still a proprietary product.
- A programmer, hacker, and developer of the most popular version of Emacs (and parrot *aficionado*, but *don't buy one for him!* at the MIT AI Laboratory, Richard Stallman decided that he wanted to make a completely open source and free Unix-compatible system.
- In 1985, Stallman published his GNU Manifesto encouraging the support for his free "GNU is Not Unix" operating system (and yes, GNU is recursively defined).
- Stallman is one of the world's most strident supporters of free (as in beer) software. He has promoted the idea of *copyleft*, which uses ideas from copyright law to preserve the use of free software. He also wrote the GNU General Public License (GPL), which is used for many pieces of free software.
- Stallman is famously eccentric — his website is worth a look.
- Stallman and others built the GNU operating system in the late 1980s, and by the early 1990s, it was mostly complete, except for the "hard" parts, which mostly meant the kernel, or the main, underlying code.
Linus Torvalds, and Linux

- Another Unix clone, called MINIX was developed by Andrew S. Tanenbaum as an academic version of Unix, though it was minimal in its implementation. It was also open source, but the licensing did not make it free software.

- In 1990, Linus Torvalds, a student studying at the University of Helsinki enrolled in a Unix course. The course used a textbook written by Tanenbaum, and it included a copy of MINIX. But, the licensing frustrated him, and he decided to start working on his own OS kernel, which eventually became the Linux kernel.

- Torvalds's paired his kernel with GNU applications, and switched the license he created for it to the GNU General Purpose License. With outreach on Internet message boards (this was still before the World Wide Web!), more developers came onboard, and the OS started taking off. After an original name of "Freax," Torvalds was convinced by others to change the name to Linux, even though he thought it was egotistical.

- In the mid-1990s, Linux use started to take off, primarily on
In the mid-1990s, Linux use started to take off, primarily on supercomputers. NASA started using Linux on clusters of computers, replacing their giant mainframes. Other companies started to do the same. These days, virtually all supercomputers use Linux operating systems.

Microsoft had the market for desktop computing basically cornered, but with the lure of a free OS, companies such as Hewlett-Packard and Dell started supporting Linux on their machines.

Linux is also used on many other devices, including all Android phones (Android is a Linux OS), and in embedded systems, and tablets. There is also SteamOS, which is a game-oriented Linux distribution.

Linux has been ported to almost all computers (including on 1980s 8-bit computers like the Commodore 64). It is famous for having versions that run on older hardware (particularly x86 hardware), keeping old computers in service much longer than anticipated.
Linux Distributions

- There are *hundreds* of distributions of Linux in the world today, and some are very popular.
- The myth cluster uses Ubuntu, as do many of the other servers as Stanford.
- It is easy (and free) to install a virtual Linux machine on many operating systems (e.g., Windows and MacOS), and it is easy to start playing around with it.
- You can find a nice video about the making of Linux here: [https://www.youtube.com/watch?v=s7u7jBwlocU](https://www.youtube.com/watch?v=s7u7jBwlocU)