CMPE 220

Class 29 – Interesting Stuff, Past and Future

The Babbage Difference Engine

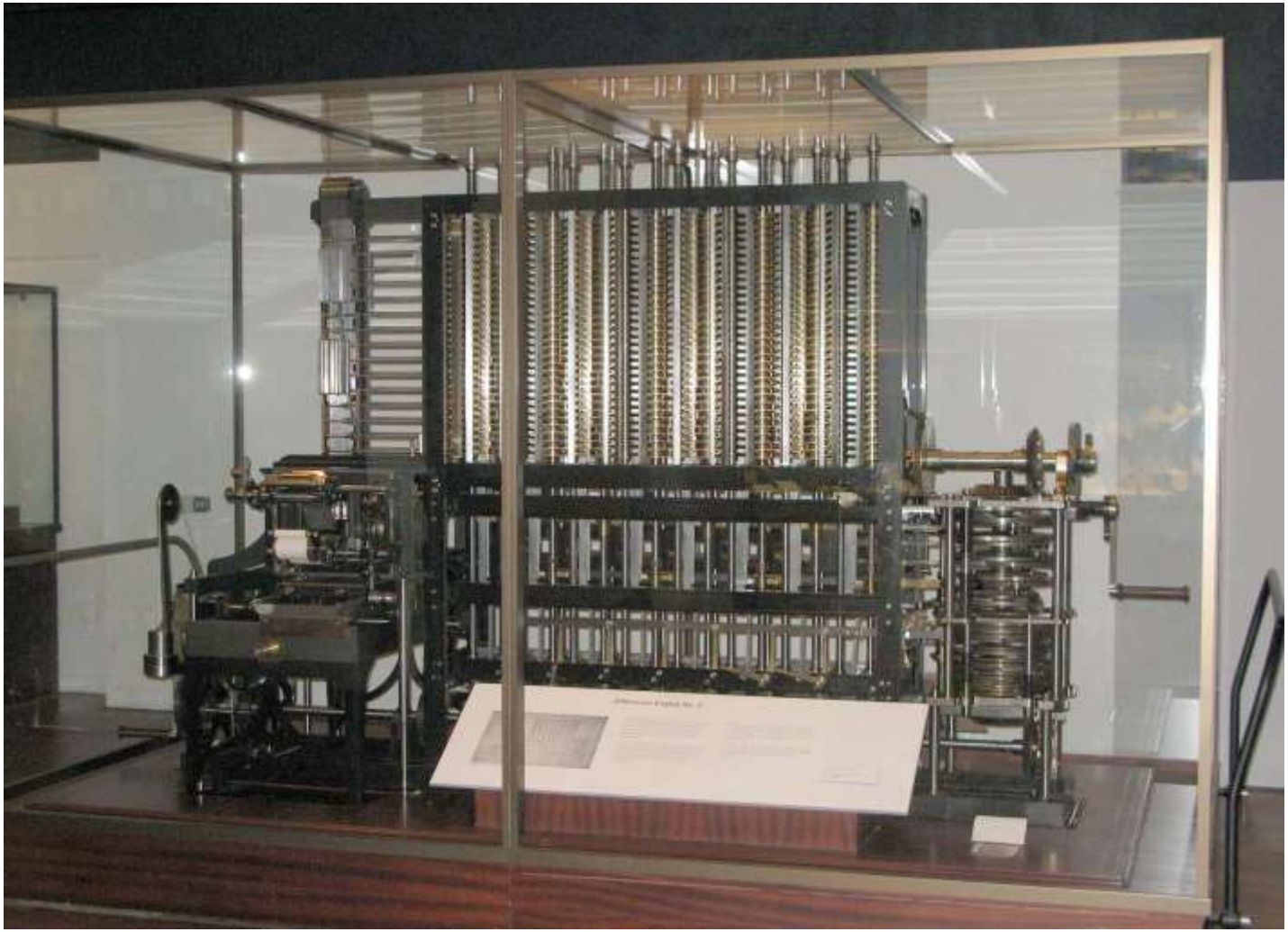
- Designed and partially built by Charles Babbage c. 1820
- A programmable mechanical computer used wheels with ten digits
- Designed to repetitively calculate mathematical functions
- Funded by the British Royal Navy to compute tables used for navigation with a sextant and timepiece
  - At the time, navigation tables were calculated by hand
  - The *difference engine* would have greater precision, and more accurate navigation

- The metalworking techniques of the time did not allow the full machine to be built, and the project was abandoned in 1842

The Babbage Difference Engine

- Babbage designed an improved machine, Difference Engine #2, in 1847-1849
- Over 8,000 parts, including a mechanical printer
- Over 4 tons, and 8 feet high
- Capable of calculating 16 digits
- In 1991, the Science Museum in London built Difference Engine #2 from Babbage's plans, and demonstrated than the machine actually works!
  - The printer was built and added in 2002
- A second machine was funded by Microsoft CTO Nathan Myhrvold and was exhibited at the **Computer History Museum** in Mountain View from 2008 to 2016

The Babbage Difference Engine #2



San José State University

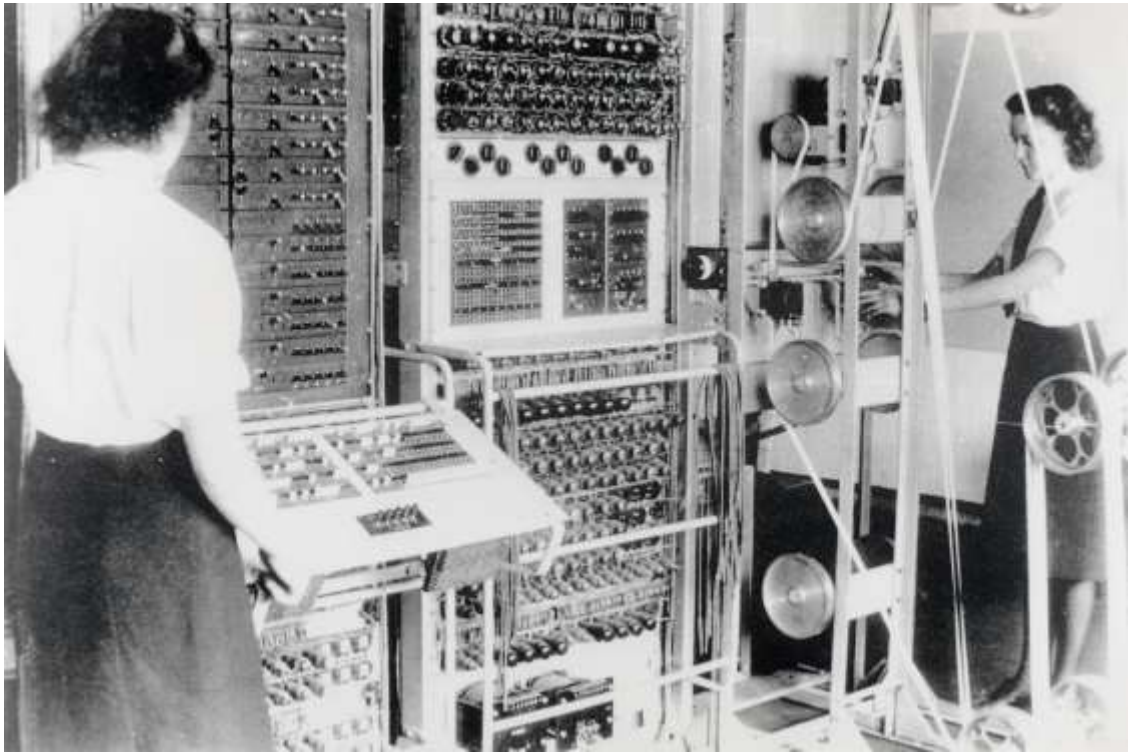
Computer History Museum – Mountain View

https://computerhistory.org/

San José State University

Electronic Computers - Collosus

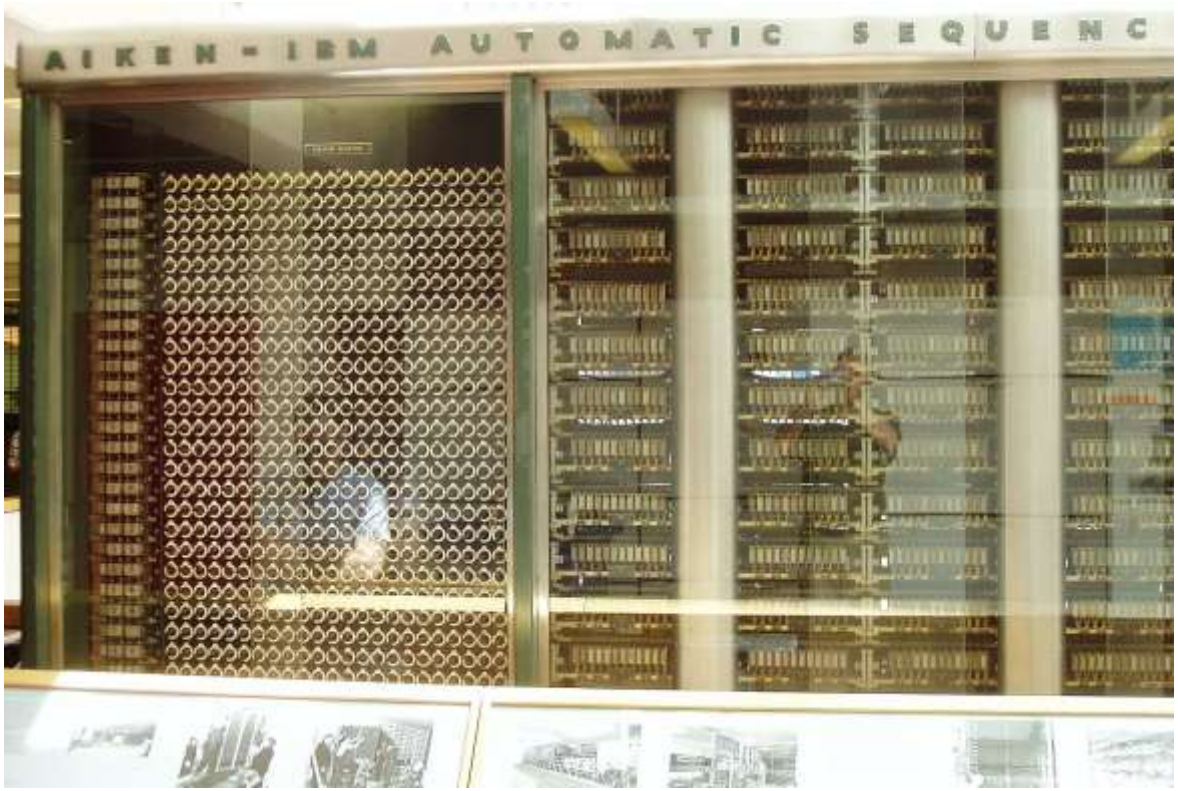
- **Colossus** – a series of computers built by the British government in 1943-1945, used to break German codes during WW II
- Alan Turning contributed to the design
- Kept Secret until 1970!



San José State University

Electronic Computers – Harvard Mark 1

- **Harvard Mark 1 (aka IBM Automatic Sequence Controller)** – completed in 1944 and used by the US government during WW II
- John von Neumann used the Mark 1 for calculations for the Manhattan project
- Grace Hopper worked on the Mark 1



San José State University

Electronic Computers - Eniac

- **Electronic Numerical Integrator and Computer (ENIAC)** – completed in 1945 and used for artillery calculations

- Considered the first general purpose programmable computer
- Grace Hopper was senior mathematician

San José State University

Electronic Computers - Univac

- **Univac**– the first *commercial* computer, delivered to the US census bureau in 1951
- Designed by Presper Eckert and John Mauchly, the creators of the ENIAC

San José State University

Electronic Computers – IBM 704

- A successor to the IBM 701, it featured floating-point arithmetic and magnetic core memory
- John Backus and his team developed the first FORTRAN compiler for the 704 in 1956

San José State University

Operating Systems: Job Control Programs

- **Batch Operating Systems** (mid 1950s) - Loaded and ran programs sequentially
- A small "job control program" to remain resident in memory.
- This was a tradeoff. It used (precious) memory, but it made more efficient use of the computer
  - Batch processing
  - Moved on quickly when programs didn't work
  - Did not require the programmer to be present
- This began a slippery slope
  - Saving work from programmers by adding functions to the resident job control program. Programming became simpler, but the control programs got bigger.

The First Operating Systems

- The first operating system used for real work was GM-NAA I/O, produced in 1956 by General Motors' Research division for its IBM 704
- Most other early operating systems for IBM mainframes were also produced by customers
- The state of affairs continued until the 1960s when IBM, already a leading hardware vendor, develop a single operating system for their new hardware, the OS/360 – released in 1964

Unix: the “Grandfather” of Operating Systems

- Developed by Dennis Ritchie and Ken Thompson, working at Bell Labs, in 1970
- Ritchie and Thompson at a PDP-11 computer, from DEC (Digital Equipment Corporation)

San José State University

The Evolution of Operating Systems

Operating System	Year	Lines of Code
OS/360	1964	1 million
Windows 3.1 *	1993	5 million
MacOS X.4 (Tiger)	2004	86 million
Debian 5.0 Linux	2009	340 million

Google's entire codebase – for all products and services – is over 2 billion lines of code

Malware

- **The Creeper** – the first virus. Developed by Bob Thomas of BBN (Boole & Babbage Networks) in 1971 as a non-malicious proof-of-concept for a self-replicating program. It printed a message: "I'M THE CREEPER. CATCH ME IF YOU CAN!"
- **Rabbit** – the first malicious virus. Developed in 1974, it was a malicious virus that consumed system resources until the system crashed.
- **Animal** – the first Trojan. Developed by John Walker in 1975; it was a game program that relied on users to install it, and then copied itself to other locations on the user's computer.
- The term "computer virus" was coined in 1983 by Fred Cohen in one of the first ever published academic papers on computer malware.

Malware

- **Brain**– the first PC (DOS) virus. Release in 1986 on floppy disk. Marked sectors of the disk as bad and wrote a message to those sectors, slowing down the disk.

  It was originally intended to protect a commercial software product by installing itself if the original software was copied.
- **nVIR** – the first Macintosh virus. Appeared in 1987.
- **WinVer 1.4** – the first Windows virus. Appeared in 1992.

Anti-Malware

- The earliest anti-virus programs appeared in the mid-1980s.
- The first *may have been* developed for the Atari ST personal computer in 1985; another was released in 1987.
- Symantac Antivirus for Macintosh (SAM) was released in 1990, and was the first antimalware program that could be updated with signatures of new viruses as they appeared.

Some Random Computer Trivia

- Bill Gates originally wanted to call Windows “Interface Manager,” but was eventually persuaded that the program needed a more intuitive name.
- Oracle Corporation was founded in 1977 by Larry Ellison and Bob Miner. Miner originally wanted to develop a word processor; Ellison wanted to develop a database. The decision was made by the flip of a coin.
- Microsoft Office (Word, Excel, and PowerPoint) was released first on Macintosh in 1989; it wasn’t released for Windows until a year later, in 1990.

The Future

Read the Research

- Incremental Refinement
  - Better algorithms for scheduling, virtual memory, file management, etc.
- Increased Virtualization
- Security: "certified" operating systems
- Specialized systems (IoT, Real Time, Cloud)
- Ritchie and Thompson would clearly recognize today's operating systems – 50 years later!

- Dominated by Windows, Macintosh, Linux, iOS, Android
  - Little room for new entries in existing markets
  - New systems will arise out of a paradigm shift

Future Directions

- Quantum Operating Systems
- Blockchain Operating Systems
- Artificial Intelligence Operating Systems

Quantum Computing

- Quantum Computers today are more like specialized processors…. like a GPU
  - They perform one task very well
  - Minimal (or non-existent) OS
- Each quantum computer has a unique language & OS
- The UK is funding development of a ”standard” quantum computer operating system
  - The goal is *portability* across quantum computers

- What does a quantum operating system do?
  - Job controller (similar to 1950s OS)

Blockchain Operating Systems

- Blockchain services are integrated into the OS, just like networking is integrated into a network OS
- Currently implemented as a virtual system.
- Secure, private, anonymous communications
- Ledgering of all interactions

- What is a Blockchain Operating System? (2018)
  - https://www.forbes.com/sites/adrianbridgwater/2018/07/03/what-is-a-blockchain-operating-system/

San José State University

AI-Based Operating Systems

History: a Dead-End

- AI research and development began in the 1980s with "expert systems"
- Expert systems attempted to replicate human reasoning
- Moderately successful for some classes of problems

- Expensive to develop
- Easy to understand and debug

AI Today

- Neural networks, machine learning, generative systems
- Based on mathematical models
- Arrives at conclusions using methods that are completely unrelated to human reasoning

- Relatively inexpensive to develop
- Extremely difficult to understand and debug

Rethinking Operating Systems Using AI (AIOS)

Two Directions

1. An operating system built to support AI – essentially an AI application tightly coupled with the underlying OS (which is more or less convention)
2. An operating system built using AI to provide OS functions

• Google, Microsoft, and others are working on both

1) Supporting AI

- **Deep Learning OS (DLOS):** This is an open-source operating system developed by NVIDIA that is optimized for deep learning applications
- **AIOS:** This is an AI operating system developed by IBM that is designed to help developers build and deploy AI applications
- **Neuromorphic Computing Operating System (NCS):** This is an operating system developed by Intel that is designed for neuromorphic computing, which is a type of computing that is modeled after the human brain
- **Brainwave:** This is an AI operating system developed by Microsoft that is optimized for running deep neural networks
- **Nengo:** This is an open-source software package for building and simulating large-scale neural models, which can be used as an operating system for AI applications

2) AI-Based Operating System

- AI used to improve algorithms for scheduling, VM, etc
- AI used to rethink fundamental OS architecture
  - E.g., do we need a scheduler?

- "The complexity of AI could make it hard for people to understand and fix any issues, or to configure and tune the system."

2) AI-Based Operating System

Research

- AI and the future of operating systems (2016)
  - https://content.iospress.com/download/information-services-and-use/isu794?id=information-services-and-use%2Fisu794
- AI Spells the End of OS as We Know It
  - https://www.bloomberg.com/opinion/articles/2023-04-19/ai-is-challenging-operating-system-supremacy#xj4y7vzkg
- AI Based OS – Future of Operating Systems (2023)
  - http://www.ijtrd.com/papers/IJTRD22752.pdf
- Artificial Intelligence in Operating Systems (2019)
  - https://www.academia.edu/42919002/Artificial_Intelligence_in_Operating_System
- Artificial Intelligence in Operating System (2019)
  - https://dl.acm.org/doi/abs/10.1145/3374587.3374635
- Artificial Intelligence in Distributed Operating System
  - https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dc78dd40d28f616386e14f72564d8a005af79faf