CS 33

Introduction to Computer Systems

What You'll Learn

- Programming in C
- Data representation
- Programming in x86 assembler language
- High-level computer architecture
- Optimizing programs
- Linking and libraries
- Basic OS functionality
- Memory management
- Network programming (Sockets)
- Multithreaded programming (POSIX threads)

Prerequisites: What You Need to Know

- Ability to program in some reasonable language (e.g., Java)
  - CS15 or CS18

What You'll Do

- Eleven 2-hour labs
- Eight one-to-two-week programming assignments
- One midterm exam
- No final exam!

Textbook

- ***Computer Systems: A Programmer's Perspective,*** **2nd Edition, Bryant and O'Hallaron, Prentice Hall 2011**
  - **very definitely required**

If Programming Languages Were Cars …

- Java would be an SUV
  - automatic transmission
  - cruise control
  - GPS navigation
  - traction control
  - gets you where you want to go
    - » safe
    - » **boring**

- C would be a Morgan
  - manual everything
  - dangerous
  - **–fun**

U-Turn Algorithm (Java Version)

1. Switch on turn signal
2. Slow down to less than 3 mph
3. Check for oncoming traffic
4. Press the accelerator lightly while turning the steering wheel pretty far in the direction you want to turn
5. Lift your foot off the accelerator and coast through the turn; press accelerator lightly as needed
6. Enter your new lane and begin driving

U-Turn Algorithm (C Version)

1. Enter turn at 30 mph in second gear
2. Position left hand on steering wheel so you can quickly turn it one full circle
3. Ease off accelerator; fully depress clutch
4. Quickly turn steering wheel either left or right as far as possible
5. A split second after starting turn, pull hard on handbrake, locking rear wheels
6. As car (rapidly) rotates, restore steering wheel to straight-ahead position
7. When car has completed 180° turn, release handbrake and clutch, fully depress accelerator

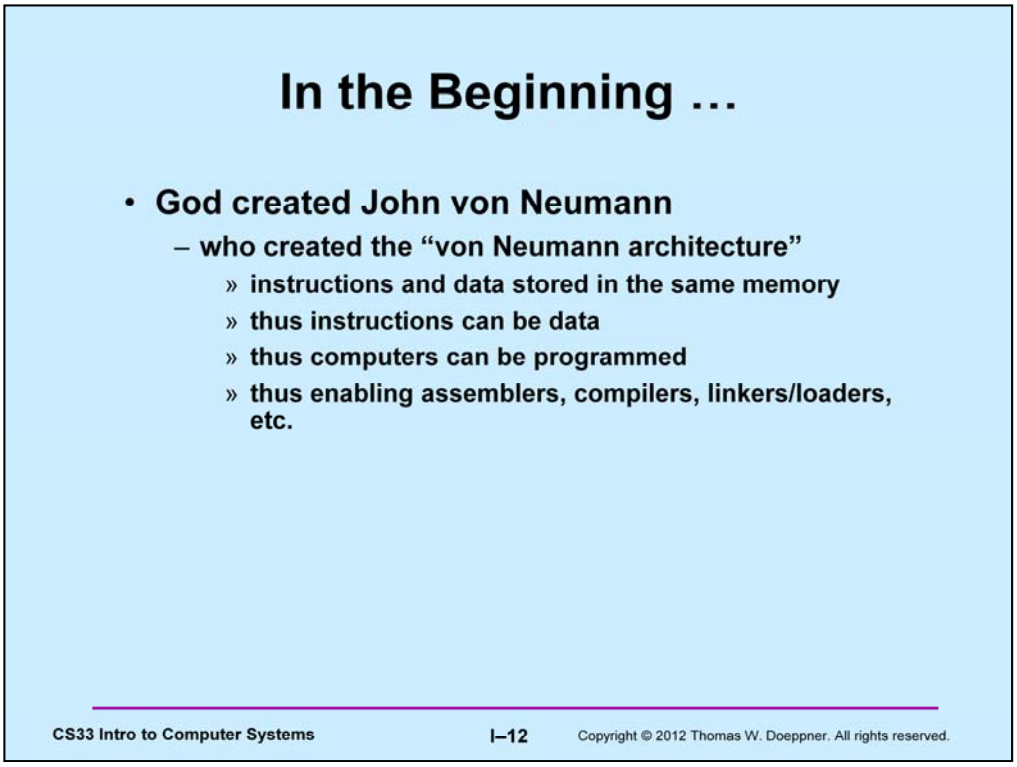
History of C

- Early 1960s: CPL (Combined Programming Language)
  - developed at Cambridge University and University of London
- 1966: BCPL (Basic CPL): simplified CPL
  - intended for systems programming
- 1969: B: simplified BCPL (stripped down so its compiler would run on minicomputer)
  - used to implement earliest Unix
- Early 1970s: C: expanded from B
  - motivation: they wanted to play "Space Travel" on minicomputer
  - used to implement all subsequent Unix OSes

See http://en.wikipedia.org/wiki/C_programming_language.

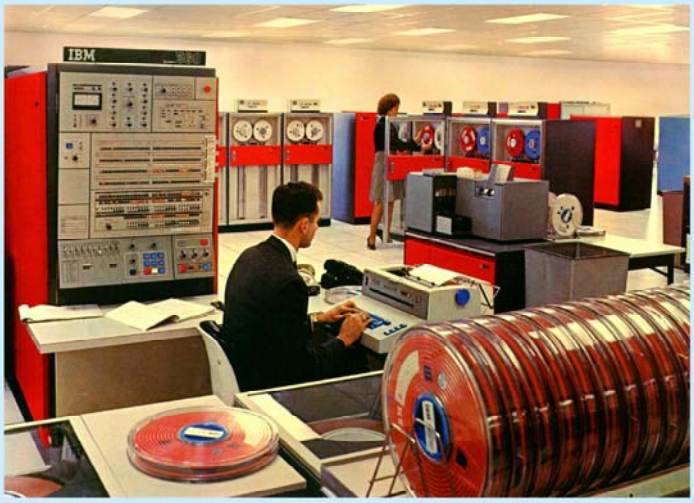
More History of C

- 1978: Textbook by Brian Kernighan and Dennis Ritchie (K&R), 1st edition, published
  - de facto standard for the language
- 1989: ANSI C specification (ANSI C)
  - 1988: K&R, 2nd edition, published, based on draft of ANSI C
- 1990: ISO C specification (C90)
  - essentially ANSI C
- 1999: Revised ISO C specification (C99)
- 2011: Further revised ISO C specification (C11)
  - too new to affect us



For a lot more information about the von Neumann architecture, see http://en.wikipedia.org/wiki/Von_Neumann_architecture.

Big Computers



This is a typical IBM 360 installation, a “mainframe” computer, in the late 1960s. The processor occupies all of the unit in the left-hand portion of the photo (the unit with “IBM” at the top).

Big Computers

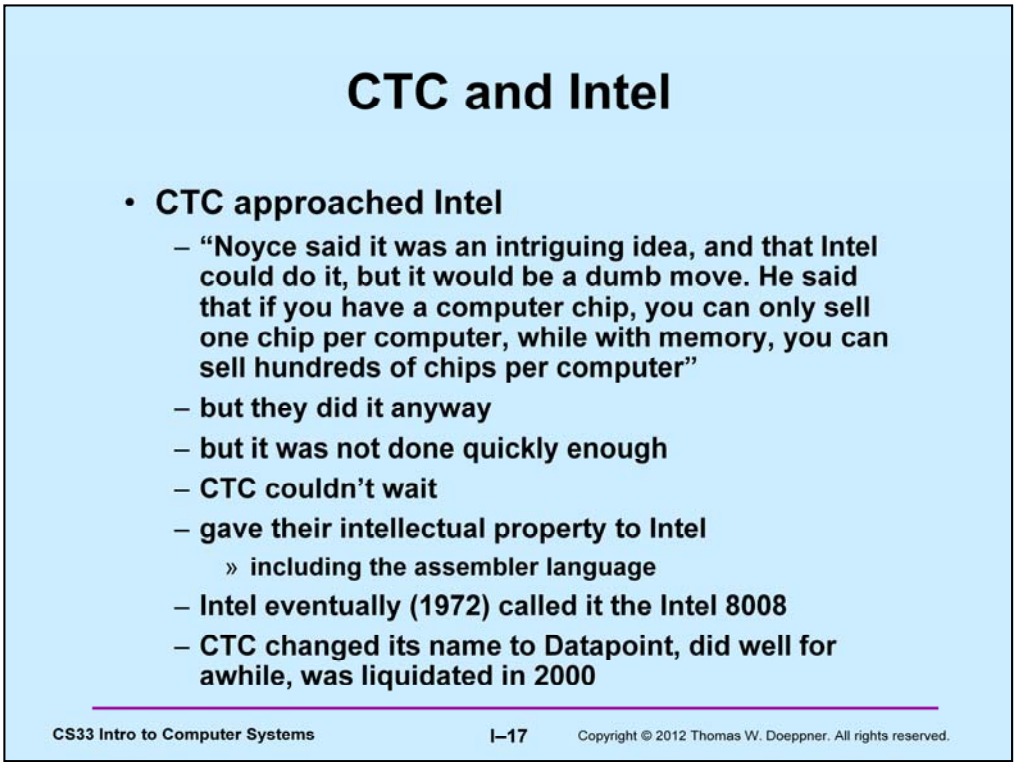
- Early computers were massive and expensive
- Processors were built from discrete logic
  - first tubes
  - then transistors
  - then LSI
- By the end of the 60s …
  - most of the architectural concepts required for this course were implemented
    - » not necessarily well …

Datapoint 2200

This is arguably the first personal computer, announced in June 1970, shipped starting in 1971. Its display is shaped like an IBM card (punched card) so as to look familiar to the typical computer user of the time. Its “footprint” was similar to an IBM Selectric typewriter, so as not to intimidate people.

CTC

- 1968: CTC wanted to build a "personal computer"
  - to get funding, they called it a "terminal"
- Designed computer using discrete components
- Developed their own assembler language
- Used a lot of power and was very hot
  - why not put processor on single chip?



The quote is from http://www.computerworld.com/s/article/print/9111341/Forgotten_PC_history_The_true_origins_of_the_personal_computer. Robert Noyce was the head of Intel at the time. Note the Intel 4004 came before the Intel 8008 and is often called the first microprocessor, but it was an entirely different project from that which produced the 8008 (and its successors).

Intel

- 8008 begat 8080
- 8080 begat 8086
- 8086 begat 8088
- 8086 begat 286
- 286 begat 386
- 386 begat 486
- 486 begat Pentium
- Pentium begat Pentium Pro
- Pentium Pro begat Pentium II
- ad infinitum

More to the Story …

- Why did the x86 dominate?
  - many competitors in the 1970s
    - » Zilog Z80, Z8000
      - used by TRS-80
    - » MOS 6502
      - used by Apple II, Commodore PET
    - » Motorola 6800
      - used by later TRS-80s

IBM PC

Enter IBM

- Dominant in the "mainframe" computer market
  - the Microsoft/Apple/Google/Facebook of the day
- Wanted to enter/dominate the "personal computer" market
  - didn't want to waste time going about it
    - » used off-the-shelf components
    - » chose Intel 8088 (cheaper version of 8086)
  - needed an OS
    - » tried to get rights to CP/M
      - product of Digital Research
      - dominant hobbyist OS
      - couldn't come to terms

Enter Microsoft

- Early products were compilers/interpreters
  - IBM contracted them for Basic interpreter
- When IBM couldn't get CP/M, Microsoft offered their services for OS
- They had an OS …
  - Xenix — a version of Unix originally for Zilog Z8000
- Didn't want to use it for IBM PC
  - had to pay royalties to AT&T
- Purchased QDOS (quick and dirty OS)
  - looked surprisingly like CP/M
  - eventually settled in court

Commercialization of the Workstation

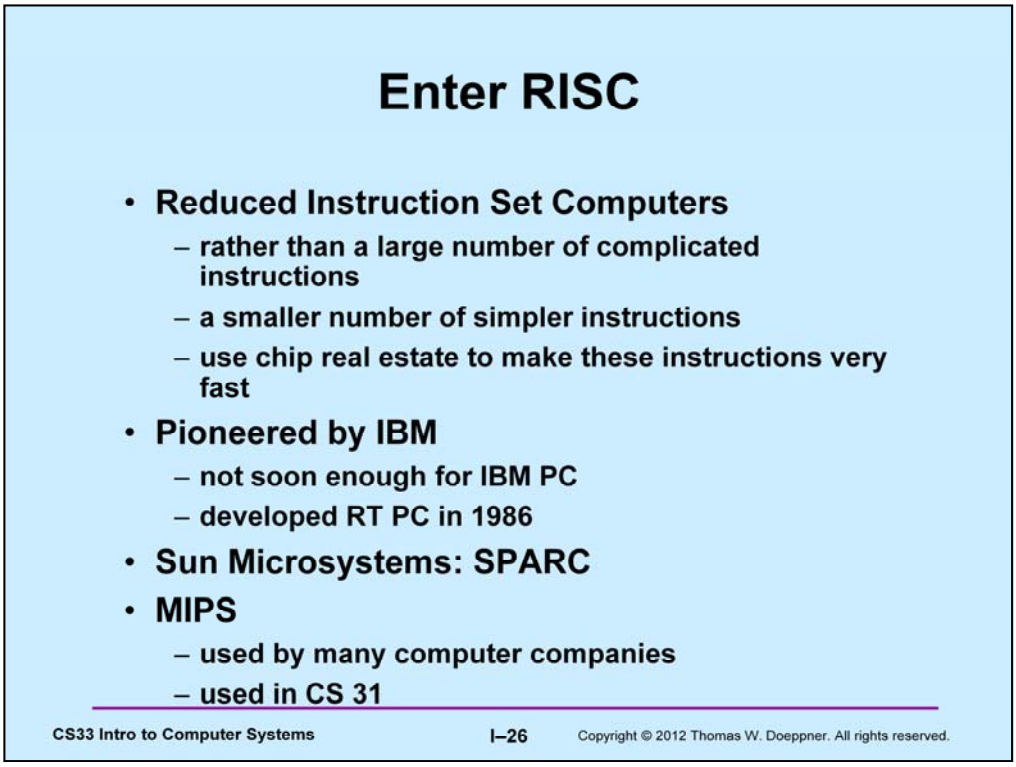
Ten years after the Xerox Alto was conceived, Apollo delivered its first product, the first commercial system offering the capabilities of the Xerox Alto (and more). Brown CS was one of the first customers. The machine in the photo resides in the CS computer museum. The "C" in the photo is the machine's serial number (in hexadecimal). These machines populated the predecessor of the Sun Lab (Foxboro Auditorium). The Apollo workstation used the Motorola 68000 processor.

Windows for the Masses

The early Macintoshes also used the Motorola 68000.

Processors in the early 80s

- Intel processors were used on the IBM PC and its clones throughout the 1980s
- Weren't sufficient for workstations
  - Microsoft used Xenix internally on a succession of non-Intel processors
  - Motorola 680x0 processors were dominant early on in the workstation market



IBM's John Cocke received ACM's Turing award (the CS equivalent of the Nobel Prize) in 1987 for work that included the development of the RISC architecture. With the advent of RISC, other designs (such as the x86) were called Complex Instruction Set Computers (CISC). SPARC stands for scalable processor architecture.

The Sun Lab

The Brown CS department chose SPARC-based Sun workstations to go into its brand-new computer lab in 1988. Hence it was called the “Sun Lab.” The photo is of an Ultra 10, a SPARC-based Sun workstation that populated the Sun Lab in the mid 1990s. It was the last computer from Sun to be in the Sun Lab.

Early 90s

- Microsoft uses Xenix until 1992
  - on RISC
- Most workstation manufacturers used RISC
- 1993: Microsoft releases new OS: Windows NT
  - ran (slowly) on x86
  - ran (faster) on PowerPC
    - » joint IBM/Motorola RISC architecture
- Apple switches to PowerPC

Intel Begins to Dominate

- x86 family
  - backwards compatible to 8086
  - 32-bit architecture now called IA32
  - CISC architecture
    - » difficult to make as fast as RISC
    - » Intel did it …
      - still can't match RISC power consumption
        - thus your iPhone isn't Intel