Career Trajectories: A Study in Change
Important Software Tools In My Early Career
Important Software Tools In My Early Career

Image courtesy of Konstantin Lanzet, CC BY-SA 3.0.
Important Software Tools In My Early Career
Important Software Tools In My Early Career

Because you should never trust bits that are not visible to the naked eye!
Project Running Out of Core Memory (4096 Words!)
Our Editor? An ASR33 Teletype!!!
1977 PDP-12 System: Vital Statistics

- Memory: 4096 12-bit words (two characters per word)
- 256Kword “mass storage”: Two DECTape drives
  - Extremely robust: Could punch holes in tape and read without errors
  - Networking: 110 baud serial connection to teletype
- Two (count them, two!) registers: A and Q
  - Oh, and a one-bit “link” register for overflow/carry detection
- Add, AND, rotate, and multiply instructions
  - You want floating point? Then implement it in software!
- CPU Clock Frequency: 625 kilohertz
But We Achieved Full-Rate Video: >20 FPS
But We Achieved Full-Rate Video: >>20 FPS
But We Achieved Full-Rate Video: >>20 FPS
Highly Optimized Sine and Cosine Functions

/ RETURNS SIN(AC), AC IS IN 128-THS
/ OF A CIRCLE, SINE IS OF FORM
/ +-S.BBBBBBBBBBB

SINSRC, 0
    AND TRGMSK
    TAD SNTABL
    DCA TRGSTO
    TAD I TRGSTO
    JMP I SINSRC

SNTABL, SINTBL

/ RETURNS COS(AC)

COSRCH, 0
    AND TRGMSK
    TAD CSTABL
    DCA TRGSTO
    TAD I TRGSTO
    JMP I COSRCH

CSTABL, COSTBL
TRGMSK, 0177
1977 PDP-12 System: Vital Statistics

- Memory: 4096 12-bit words (two characters per word)
- CPU Clock Frequency: 625 kilohertz
- Two (count them, **two**) registers: A and Q
  - Oh, and a one-bit “link” register for overflow/carry detection
- Add, AND, rotate, and multiply instructions
  - You want floating point? Then implement it in software!
- 256Kword “mass storage”: Two DECTape drives
  - Extremely robust: Could punch holes in tape and read without errors
- Networking: 110 baud serial connection to teletype
- Low *low* price of $25K!!!
Much Better Than 1949 CSIRAC!

https://en.wikipedia.org/wiki/CSIRAC
Photo by John O’Neill under GNU FDL v1.2
1949: CSIRAC Vital Statistics

- **CSIRAC**: Oldest intact electronic stored-program computer
  - Operational in November 1949 at University of Melbourne

- **2,000 Vacuum tubes**: Each an incandescent lightbulb in size
  - And less capable than a transistor: Need more tubes than transistors

- **768 words of memory, 20 bits each**, in mercury delay lines
  - Hence “surviving” rather than operational
    - 2017 safety regs unforgiving of metallic mercury and exposed 600V wiring

- **CPU core clock frequency of... 1KHz**

- **Energy-efficient design sips only 30kW**

- **Price? $10,000,000 AU**
2017 Smartphone SoC Vital Statistics

- Samsung Galaxy S8
- 8-core 64-bit CPU
- Several gigabytes of DRAM (but only eight bits wide)
- CPU core clock frequency of... 1.7GHz - 2.3GHz
- Energy-efficient design sips way less than 30W
  – Let alone 30KW – call it 3W
- Price? $750 US
- Plus has cellular modem, WiFi, camera, display, touchscreen, audio, Bluetooth, DSP, GPU, GPS, NFC, USB, Flash, ...
### What a Difference 68 Years Makes!!!

<table>
<thead>
<tr>
<th></th>
<th>CSIRAC</th>
<th>Galaxy S8</th>
<th>OOM</th>
<th>Doublings</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUs</td>
<td>1</td>
<td>8</td>
<td>0.90</td>
<td>3.00</td>
</tr>
<tr>
<td>CPU Clock</td>
<td>1KHz</td>
<td>1.7GHz</td>
<td>6.23</td>
<td>20.70</td>
</tr>
<tr>
<td>RAM (bits)</td>
<td>15,360</td>
<td>17,179,869,184</td>
<td>6.05</td>
<td>20.09</td>
</tr>
<tr>
<td>Power</td>
<td>30KW</td>
<td>~3W</td>
<td>4.00</td>
<td>13.29</td>
</tr>
<tr>
<td>Price</td>
<td>$10,000,000</td>
<td>$750</td>
<td>4.12</td>
<td>13.70</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td>21.31</td>
<td></td>
<td>70.78</td>
</tr>
</tbody>
</table>

- Not allowing for cellular modem, WiFi, camera, display, touchscreen, audio, Bluetooth, DSP, GPU, GPS, NFC, USB, Flash, ...
- Not allowing for currency conversion or (much more important) inflation, which is another factor of 10
IBM used up to six qubits of its superconducting quantum processor to address electronic structure problems for the molecules H$_2$, LiH and BeH$_2$

50-qubit system performance/scalability PoC planned

But the Progress is Not Just in Computing!!!
But the Progress is Not Just in Computing!!!
50 Years of Progress in Mathematics

- 1970: Proof that Hilbert’s 10th problem is unsolvable
- 1976: Proof of the four-color problem (stood for centuries)
- 1984: Polynomial-time algorithm for solving linear programming problems
- 1994: Proof of Fermat’s Last Theorem (stood for centuries)
- 1998: Proof of Kepler’s conjecture (sphere packing, stood for centuries)
- 2002: Proof of Catalan’s conjecture ($2^3$ and $3^2$, stood for centuries)
- 2002: Polynomial-time integer primality test
- 2003: Proof of the Poincaré conjecture (topology)
- 2004: Proof of the classification of finite simple groups
- 2013: Proof that there is no bound on the values of pairs of primes differing by a finite number (first real progress in more than two millennia)
But the Progress is Not Just in Computing!!!
50 Years of Progress in Mathematics

- 1970: Proof that Hilbert's 10th problem is unsolvable
- 1976: Proof of the four-color problem (stood for centuries)
- 1984: Polynomial-time algorithm for solving linear programming problems
- 1994: Proof of Fermat's Last Theorem (stood for centuries)
- 1998: Proof of Kepler's conjecture (sphere packing, stood for centuries)
- 2002: Proof of Catalan's conjecture (2^3 and 3^2, stood for centuries)
- 2002: Polynomial-time integer primality test
- 2003: Proof of the Poincaré conjecture (topology)
- 2004: Proof of the classification of finite simple groups
- 2013: Proof that there is no bound on the values of pairs of primes differing by a finite number (first real progress in more than two millennia)

You are living in a golden age surpassing all golden ages!!!
Where Might Your Career Go?

Computing Mostly Here

People

Ask

Tell

Researcher

Superprogrammer

Things
Where Might Your Career Go?

- Your career might go a lot of different places
- No education system on this earth can prepare you for all the different things you might need to do
  - In 1976-1981, would OSU have prepared me for the Web? Mobile? Open-source licensing? Other legal issues?
- Your education is a foundation to support life-long learning
Expanding Our Educational Foundations
And foundations need maintenance, repair, and sometimes expansion!
Expanding My Educational Foundation

- BS Mechanical Engineering, Oregon State University, 1981
- BS Computer Science, Oregon State University, 1981
Expanding My Educational Foundation

- BS Mechanical Engineering, Oregon State University, 1981
- BS Computer Science, Oregon State University, 1981
- MS Computer Science, Oregon State University, 1988
Expanding My Educational Foundation

- BS Mechanical Engineering, Oregon State University, 1981
- BS Computer Science, Oregon State University, 1981
- MS Computer Science, Oregon State University, 1988
- Ph.D. Computer Science and Engineering, OGI School of Science and Engineering at Oregon Health & Sciences University, 2004
Expanding My Educational Foundation

- BS Mechanical Engineering, Oregon State University, 1981
- BS Computer Science, Oregon State University, 1981
- MS Computer Science, Oregon State University, 1988
- Ph.D. Computer Science and Engineering, OGI School of Science and Engineering at Oregon Health & Sciences University, 2004
- But I have never taken a course on artificial intelligence, C language, hardware architecture, version control, software validation, IP law, or thread-based parallel programming
Expanding My Educational Foundation

- BS Mechanical Engineering, Oregon State University, 1981
- BS Computer Science, Oregon State University, 1981
- MS Computer Science, Oregon State University, 1988
- Ph.D. Computer Science and Engineering, OGI School of Science and Engineering at Oregon Health & Sciences University, 2004
- But I have never taken a course on artificial intelligence, C language, hardware architecture, version control, software validation, IP law, or thread-based parallel programming – I instead learned these critically important topics on the job
How My Career Evolved

- **1977-1980**
  - Business applications in FORTRAN and COBOL

- **1981-1985**
  - Soft real-time systems in PASCAL, assembly, and C

- **1986-1990**
  - UNIX systems administration, networking research in C on SunOS

- **1990-1999**
  - Parallel UNIX kernel hacking in C on DYNIX/ptx

- **1999-2000**
  - Parallel UNIX kernel hacking in C on AIX

- **2001-present**
  - Parallel, real-time, and energy-efficient kernel hacking in C on Linux
How Might Things Change in the Future?

- If you are 18 years old and plan to retire at age 70: 2069!!!
How Might Things Change in the Future?

- If you are 18 years old and plan to retire at age 70: 2069!!!
- Not much is known about the year 2069
  - Anyone who will be over 50 has already been born
How Might Things Change in the Future?

- If you are 18 years old and plan to retire at age 70: 2069!!
- Not much is known about the year 2069
  - Anyone who will be over 50 has already been born
  - Unless someone invents a time machine
Education: A Foundation For Your Career

How Might Things Change in the Future?

- If you are 18 years old and plan to retire at age 70: 2069!!!
- Not much is known about the year 2069
  - Anyone who will be over 50 has already been born
  - Unless someone invents a time machine
- But we can look at changes since 1965:
How Might Things Change in the Future?

- If you are 18 years old and plan to retire at age 70: 2069!!!
- Not much is known about the year 2069
  - Anyone who will be over 50 has already been born
  - Unless someone invents a time machine
- But we can look at changes since 1965:
  - Windows and icons and mice created and took over the world
  - Ethernet created, took over the world, supplanted by WiFi
  - Intel x86 was created and took over much of the world
  - LEDs red/infrared lab curiosities to pretty much any color at low price
  - DOS created, took over much of the world, obsoleted by Windows
  - Cellphones were created and took over much of the world
  - UNIX and then Linux were created and took over much of the world
  - World wide web was created and took over much of the world
  - Smartphones were created and took over much of the world
  - Self-driving cars were created, and might be taking over the world
But Some Things Never Change
But Some Things Never Change

- Quality is always critically important
  - But what constitutes “quality” varies over both time and space

- Communication, both verbal and written is critically important
  - Beyond a certain point, an idea you cannot communicate is useless
  - And there is a big advantage of written communication as you age...

- Your job will probably become more about people over time
  - Hence the “communication” point above and the “scope” point below

- Your scope must increase over time
  - What was good enough then is probably not good enough now...
How Can You Increase Your Scope?
How Can You Increase Your Scope?

- Over time, anyone in a technical field must increase their scope just to maintain a constant level of seniority:
  - Move to management or program management
  - Move to “hot” field du jour (for example, evangelist or trainer)
  - Training (both giving and taking)
  - Move to sales, technical sales support, technical marketing
  - Increase technical skill/efficiency
  - Practice, practice, practice: strive for fluency
  - Create artifacts that are widely used
One Way To Increase Your Scope

People

Analysis
Inbound Marketing

Manager
Professor

Ask

Computing Mostly Here

Tell

Researcher

Superprogrammer

Things
Another Way to Increase Your Scope
“Scope” Means Different Things In Different Areas

Productivity
- Web/Mobile
- Applications

Frameworks/Middleware (e.g., NoSQL)
- System Libraries
- Operating-System Kernel
- Firmware
- Hardware
How Can You Increase Your Scope?

- Over time, anyone in a technical field must increase their scope just to maintain a constant level of seniority:
  - Move to management or program management
  - Move to “hot” field du jour
  - Training (both giving and taking)
  - Move to sales, technical sales support, technical marketing
  - Increase technical skill/efficiency
  - Practice, practice, practice: strive for fluency
  - Create artifacts that are widely used
  - *Avoid carrying the burdens of the past*
Software Quality and Burdens of the Past

- Software's master or software's prisoner?
  - Hard to increase your scope while a prisoner of your SW's bugs!!!
Software Quality and Burdens of the Past

- Software's master or software's prisoner?
  - Hard to increase your scope while a prisoner of your SW's bugs!!!
  - But there is a time and place for “fast and dirty” software...

![Image of a prisoner in a cage]

![Image of a man with a whip]
Summary

- **Education is a foundation for your career**
  - You will need to maintain it, add to it, and build on top of it
  - The world will change, and your skill set must change with it

- **But some things never change**
  - Importance of communication and quality, the need to increase your scope over time, and the need to increase your people skills over time

- **You will need to let go of some of the past to grasp the future**
  - Which of today's technologies are tomorrow's core memory?
Summary

- Education is a foundation for your career
  - You will need to maintain it, add to it, and build on top of it
  - The world will change, and your skill set must change with it

- But some things never change
  - Importance of communication and quality, the need to increase your scope over time, and the need to increase your people skills over time

- You will need to let go of some of the past to grasp the future
  - Which of today's technologies are tomorrow's core memory?
  - Tell which is which and you'll be rich!!!
Questions?