OS Foundations

Peter J. Denning
October 4, 2015
Our Story

OS Principles began emerging 1960,
Grew across many generations of technologies,
And left a rich heritage in the minds, hearts, and souls of all who use computers.
Our Story

Timelines

Personal example

Research Lessons

OS a force on all of computing
We will be surfing ...
<table>
<thead>
<tr>
<th>Decade</th>
<th>Number of new OS’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950s</td>
<td>9</td>
</tr>
<tr>
<td>1960s</td>
<td>40</td>
</tr>
<tr>
<td>1970s</td>
<td>50</td>
</tr>
<tr>
<td>1980s</td>
<td>96</td>
</tr>
<tr>
<td>1990s</td>
<td>73</td>
</tr>
<tr>
<td>2000s</td>
<td>245</td>
</tr>
<tr>
<td>2010s</td>
<td>320</td>
</tr>
</tbody>
</table>

Number of new OS’s per decade (Wikipedia)
Eras of Operating Systems

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>Interactive</td>
<td>(L)</td>
<td>Distributed</td>
<td>(I)</td>
<td>Cloud-Mobile</td>
<td></td>
</tr>
</tbody>
</table>
Eras of Operating Systems

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>Interactive</td>
<td>(L)</td>
<td>Distributed</td>
<td>(I)</td>
<td>Cloud-Mobile</td>
<td></td>
</tr>
</tbody>
</table>
Eras of Operating Systems

<table>
<thead>
<tr>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td>Batch</td>
</tr>
<tr>
<td>1960</td>
<td>Interactive</td>
</tr>
<tr>
<td>1970</td>
<td>(L) Distributed</td>
</tr>
<tr>
<td>1980</td>
<td>(I)</td>
</tr>
<tr>
<td>1990</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>Cloud-Mobile</td>
</tr>
<tr>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

“computer utility”
Eras of Operating Systems

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>Interactive</td>
<td>(L)</td>
<td>Distributed</td>
<td>(I)</td>
<td>Cloud-Mobile</td>
<td></td>
</tr>
</tbody>
</table>
Eras of Operating Systems

<table>
<thead>
<tr>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td>Batch</td>
</tr>
<tr>
<td>1960</td>
<td>Interactive</td>
</tr>
<tr>
<td>1970</td>
<td>(L)</td>
</tr>
<tr>
<td>1980</td>
<td>Distributed</td>
</tr>
<tr>
<td>1990</td>
<td>(I)</td>
</tr>
<tr>
<td>2000</td>
<td>Cloud-Mobile</td>
</tr>
<tr>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

Personal Computers
# Eras of Operating Systems

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Batch</strong></td>
<td><strong>Interactive</strong></td>
<td>(L)</td>
<td><strong>Distributed</strong></td>
<td>(I)</td>
<td><strong>Cloud-Mobile</strong></td>
<td></td>
</tr>
<tr>
<td>One job at a time</td>
<td>Many jobs sharing</td>
<td>Personalized immersive world managing work (desktop)</td>
<td></td>
<td></td>
<td>Personalized immersive world managing life and social relations</td>
<td></td>
</tr>
</tbody>
</table>
Eras of Operating Systems


Batch Interactive (L) Distributed (I) Cloud-Mobile

Internet

“OS interfaces with”: TELNET FTP SMTP RLOGIN RCOPY

“OS integrates with”: Protocol software IPC, RPC Daemon processes Client-server, X-windows Hyperlink, URL Browser Search
Eras of Operating Systems

<table>
<thead>
<tr>
<th>Year</th>
<th>Era</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td>Batch</td>
</tr>
<tr>
<td>1960</td>
<td>Interactive</td>
</tr>
<tr>
<td>1970</td>
<td>(L)</td>
</tr>
<tr>
<td>1980</td>
<td>Distributed</td>
</tr>
<tr>
<td>1990</td>
<td>(I)</td>
</tr>
<tr>
<td>2000</td>
<td>Cloud-Mobile</td>
</tr>
<tr>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

Protection-security, languages, abstraction, memory management, files, fault tolerance, virtualization, parallel computing, network, cloud
Eras of Operating Systems

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>Interactive</td>
<td>(L)</td>
<td>Distributed</td>
<td>(I)</td>
<td>Cloud-Mobile</td>
<td></td>
</tr>
</tbody>
</table>

- Protection-security, languages, abstraction, memory management, files, fault tolerance, virtualization, parallel computing, network, cloud
- OS principles in education
Eras of Operating Systems

<table>
<thead>
<tr>
<th>Year</th>
<th>Era</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td>Batch</td>
</tr>
<tr>
<td>1960</td>
<td>Interactive</td>
</tr>
<tr>
<td>1970</td>
<td>(L) Distributed</td>
</tr>
<tr>
<td>1980</td>
<td>(I) Distributed</td>
</tr>
<tr>
<td>1990</td>
<td>Cloud-Mobile</td>
</tr>
<tr>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

Protection-security, languages, abstraction, memory management, files, fault tolerance, virtualization, parallel computing, network, cloud

OS principles in education

Capabilities

15
Development of Principles

What is a computing principle?
Development of Principles

What is a computing principle?

Law
Development of Principles

What is a computing principle?

Law

\[ c(t) = \min( a(t), s(t)+1 ) \]

\[ M = (\text{spacetime}) \times (\text{throughput}) \]

Mean Value Equations

Locality
Development of Principles

What is a computing principle?

Law

Statement of Design Wisdom
Development of Principles

What is a computing principle?

Law

Statement of Design Wisdom

Information hiding

Levels of abstraction
Development of Principles

What is a computing principle?

Law

Statement of Design Wisdom

Cosmic = timeless + spaceless (Jim Gray)
Eight programming support objectives added by 1965 seeded the research

- Hierarchical file systems
- Controlled information sharing
- Interactive programming
- Modular programming
- Multiprogramming
- Fault tolerant structures
- Interrupt systems
- Automated overlays (virtual memory)

+HLLs for programming the OS
Principles govern only the recurrences part of our story.

Accidents and unforeseen contingencies constantly appear.

We respond to them:

With bug fixes, patches, new designs, new apps.

With research seeking greater understanding and occasionally opening new insights and new principles.
OS Principles into CS

- Interactivity
- Concurrent Processes
- Locality
- Naming - Mapping
- Protection - Sharing
- Virtualization
- System Languages
OS Principles into CS

In all,

13 of 41 CS Principles (30%)

Source: greatprinciples.org
OS Principles into CS

• First non-math core course 1971
• Expanded definition of core to include systems
• Unchallenged for 44 years
Two Cosmic Principles Revealed in Memory Management

Locality

Location independent addressing

Emerged from virtual memory research

Originally seeking to automate overlays
Locality Principle

Big Adventure

I became involved at Project MAC 1965

Many people involved (thousands!)
First motivations ca. 1965 performance related:

- Performance of virtual memory
- Multiprogramming
- Thrashing
Thrashing
Saltzer’s Challenge

Tune one parameter to lock in optimal performance
Reference Map – Initial Intuitions

Source: Adrian McMenamin
Key insights:

• Temporal and spatial clustering (Belady, Denning 1966)
• Working sets (Denning 1966)
• Reference maps (IBM ca 1969)
• Optimality principle (Belady 1966, Prieve and Fabry 1976, Gray 1995)
Working set

Pages used in previous virtual time window of size T

Vertical slice in page reference map
What We Have Learned

WS = locality set most of the time

WS policy near optimal (VMIN)

Economical implementations (WSCLOCK)

Prevents thrashing

Answers Saltzer’s challenge
Locality Principle

Initial intuitions confirmed

All computations display locality (empirical)

All computations must display locality (theory)

Harnessing locality always pays off

- caching
- parallelizing
- performance
- no-thrashing
Location Independent Addressing

Key insights:

• Paging (U Manchester 1949)
• Virtual v. real address (ca 1959 Kilburn and Fotheringham)
• Segmentation in universal hierarchical address space (Dennis 1965)

From which flowed:

• Dynamic mapping virtual to real via page table
• Demand paging
• Replacement algorithms
• MMU and TLB mapping architecture
• Hierarchical naming systems
Huge benefits:

- Location independence
- Logical partitioning (address space isolation)
- Artificial contiguity
- Relocation
- Distributed naming authorities

Evolved into global, all-time unique addresses for digital objects anywhere in a system.

Hierarchical Internet URLs and domain names.

Now Internet is a huge virtual address space of capabilities, URLs, and DOIs with mapping via DNS and handle-servers.
Patterns I learned from OS Research

1. There is never certainty
2. Occasionally an insight charts a new direction
3. Technology inflection points may trigger avalanches
4. Searching for what works: building, experimenting, tinkering
5. Always in a social network
6. Theory follows practice
Summary

• Rich heritage from OS evolution
Summary

• Rich heritage from OS evolution

Batch ... Interactive ... Immersive
Summary

• Rich heritage from OS evolution

  Batch ... Interactive ... Immersive

• Fundamental CS principles emerged from our research
Summary

- Rich heritage from OS evolution
  
  Batch … Interactive … Immersive

- Fundamental CS principles emerged from our research

- OS drove major change to CS curriculum
Summary

• Rich heritage from OS evolution

  Batch ... Interactive ... Immersive

• Fundamental CS principles emerged from our research

• OS drove major change to CS curriculum

• OS research a unique blend of engineering, experimenting, and modeling
Challenges Ahead

Automation of knowledge work – end of “knowledge age”

Network: space of social power, action, identities

Security and privacy in the Internet of Things

Size, complexity, scale of systems

Integrating with bio and nano tech
End of this story