Software Systems Research is Irrelevant
The Rob Pike Polemic

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu
August 22, 2001

http://www.cs.unc.edu/~jeffay/courses/comp290

Rob Pike (Lucent Bell Labs) isn’t happy…

◆ Rob is pessimistic about systems research these days
  » “The situation is genuinely bad and requires action”

◆ Systems software research has become a sideline to
  the excitement in the computing industry
  » When did you last see an exciting non-commercial demo?

◆ Ironically, at a time when computing is almost the
  definition of innovation, research in both software
  and hardware at universities and much of industry is
  becoming insular, ossified, and irrelevant

◆ There are many reasons, some avoidable, some
  endemic

A field in decline

- Who needs new operating systems, anyway?
  » Maybe no one, but then that supports the thesis.

- “But now there are lots of papers in file systems,
  performance, security, web caching, etc.”
  » Yes, but is anyone outside the research field paying
    attention?

Where is the innovation?

- Microsoft, mostly
  » Exercise: Compare 1990 Microsoft software with 2000

- If you claim that’s not innovation, but copying, I
  reply that Java is to C++ as Windows is to the
  Macintosh: an industrial response to an interesting
  but technically flawed piece of systems software

- If systems research was relevant, we’d see new
  operating systems and new languages making inroads
  into the industry, the way we did in the ’70s and ’80s

- Instead, we see a thriving software industry that
  largely ignores research, and a research community
  that writes papers rather than software

A high-end workstation:

<table>
<thead>
<tr>
<th></th>
<th>1990</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unix</td>
<td>Unix</td>
<td>Unix</td>
</tr>
<tr>
<td>X Windows</td>
<td>X Windows</td>
<td>X Windows</td>
</tr>
<tr>
<td>Emacs</td>
<td>Emacs</td>
<td>Netscape</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>TCP/IP</td>
<td>Netscape</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>C++</td>
</tr>
<tr>
<td>C++</td>
<td>C++</td>
<td>Java</td>
</tr>
<tr>
<td>Perl (a little)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hardware has changed dramatically; software is stagnant

Innovation? New? No, it’s just another copy of the
same old stuff

OLD stuff. Compare program development on Linux
with Microsoft Visual Studio or one of the IBM
Java/web toolkits

Linux’s success may indeed be the single strongest
argument for my thesis: The excitement generated by
a clone of a decades-old operating system demonstra-
tes the void that the systems software research
community has failed to fill

Besides, Linux’s cleverness is not in the software, but
in the development model, hardly a triumph of
academic CS (especially software engineering) by any
measure
What is systems research these days?

- In a misguided attempt to seem scientific, there’s too much measurement: performance minutiae & bad charts
- By contrast, a new language or OS can make the machine feel different, give excitement, novelty. But today that’s done by a cool web site or a higher CPU clock rate or some cute little device that should be a computer but isn’t
- The art is gone
- But art is not science, and that’s part of the point. Systems research cannot be just science; there must be engineering, design, and art

What happened?

- A lot of things:
  - PC
  - Microsoft
  - Web
  - Standards
  - Orthodoxy
  - Change of scale
  - Unix
  - Linux
  - Startups
  - Grandma

What happened? — The PC

- Hardware became cheap & cheap hardware became good
- Eventually, if it didn’t run on a PC, it didn’t matter because the average, mean, median, and mode computer was a PC
- Even into the 1980s, much systems work revolved around new architectures (RISC, iAPX/432, Lisp Machines). No more. A major source of interesting problems and, perhaps, interesting solutions is gone
- Much systems work also revolved around making stuff work across architectures: portability. But when hardware’s all the same, it’s a non-issue
- And that’s just the PC as hardware; as software, it’s the same sort of story

What happened? — Microsoft

- Enough has been said about this topic (Although people will continue to say lots more)
- Microsoft is an easy target, but it’s a scapegoat, not the real source of difficulty

What happened? — The Web

- The web happened in the early 1990s and it surprised the computer science community as much as the commercial one
- It then came to dominate much of the discussion, but not to much effect. Business controls it
  - The web came from physicists and prospered in industry
- Bruce Lindsay of IBM: HDLC C HTTP/HTML; 3270s have been replaced by web browsers
  - Compare with Visicalc and PC
- Research has contributed little, despite a huge flow of papers on caches, proxies, server architectures, etc.

What happened? — Standards

- To be a viable computer system, one must honor a huge list of large, and often changing, standards
  - TCP/IP, HTTP, HTML, XML, CORBA, Unicode, POSIX, NFS, SMB, MIME, POP, IMAP, X, ...
- A huge amount of work, but if you don’t honor the standards you’re marginalized
  - Estimate that 90-95% of the work in Plan 9 was directly or indirectly to honor externally imposed standards
- At another level, instruction architectures, buses, etc. have the same influence
- With so much externally imposed structure, there’s little slop left for novelty
- Plus, commercial companies that ‘own’ standards, e.g., Microsoft, Cisco, deliberately make standards hard to comply with, to frustrate competition
  - Academia is a casualty
Systems Software Research is Irrelevant
What happened? — Orthodoxy

◆ Today’s graduating PhDs use Unix, X, Emacs, and Tex
  » That’s their world. It’s often the only computing world
  they’ve ever used for technical work
◆ Twenty years ago, a student would have been exposed
to a wide variety of operating systems, all with good
and bad points
◆ New employees in our lab now bring their world with
them, or expect it to be there when they arrive. That’s
reasonable, but there was a time when joining a new lab
was a chance to explore new ways of working
◆ Narrowness of experience leads to narrowness of
imagination

Systems Software Research is Irrelevant
Change of scale

◆ With so many external constraints, and so many
things already done, much of the interesting work
requires effort on a large scale. Many person-years
are required to write a modern, realistic system
  » This is beyond the scope of most university departments
◆ Also, the time scale is long: from design to final
version can be five years
  » Again, that’s beyond the scope of most grad students
◆ This means that industry tends to do the big, defining
projects (operating systems, infrastructure, etc.) and
small research groups must find smaller things to
work on

Systems Software Research is Irrelevant
UNIX

◆ New operating systems today tend to be just ways of
reimplementing Unix
  » If they have a novel architecture, and some do, the first thing to
  build is the Unix emulation layer
◆ How can OS research be relevant when the resulting
operating systems are all indistinguishable?
◆ There was a claim in the late ‘70s & early ‘80s that Unix
had killed operating systems research because no one
would try anything else. I didn’t believe it but now
grudgingly accept that the claim may be true (Microsoft
notwithstanding)
◆ A victim of its own success: portability led to ubiquity
  » That meant architecture didn’t matter, so now there’s only one
◆ Linux is the hot new thing... but it’s just another Unix

Systems Software Research is Irrelevant
Linux the academic Microsoft Windows

◆ The holy trinity: Linux, gcc, and Netscape
  » Of course, it’s just another orthodoxy
◆ These have become icons not because of what they
are, but because of what they are not: Microsoft
◆ But technically, they’re not that hot. And Microsoft
has been working hard, and I claim that on many (not
all) dimensions, their corresponding products are
superior technically (and they continue to improve)
◆ Linux may fall into the Macintosh trap: smug
isolation leading to (near) obsolescence
◆ Besides, systems research is doing little to advance
the trinity
Startups

- Startups are the dominant competition for academia for ideas, funds, personnel, and students
  - Others are Microsoft, big corporations, legions of free hackers, and the IETF
- In response, government-funded & especially corporate research is directed at very fast 'return on investment'
- This distorts the priorities: Research is bent towards what can make big money (IPO) in a year.
  - Horizon is too short for long-term work. (There go infrastructure and the problems of scale.)
- Funding sources (government, industry) perceive the same pressures, so there is a vicious circle
  - The metric of merit is wrong
- Stanford now encourages students to go to startups because successful CEOs give money to the campus

Things to do

- Startups are too focused on short time scale & practical results to try new things, big corporations are too focused on existing priorities to try new things, startups suck energy from research
  - But gold rushes leave ghost towns; be prepared to move in
- Go back to thinking about & building systems; narrowness is irrelevant; breadth is relevant: it’s the essence of system.
  - Work on how systems behave/work, not just how they compare
  - Concentrate on interfaces & architecture, not just engineering.
  - Be courageous. Try different things; experiment. Try to give a cool demo
- Measure success by ideas, not just papers and money
  - Make the industry want your work

Things to build

- There has been much talk about component architectures but only one true success: Unix pipes
  - It should be possible to build interactive and distributed applications from piece parts
- The future is distributed computation, but the language community has done very little to address that possibility
- The Web has dominated how systems present and use information: the model is forced interaction; the user must go get it
  - Let’s go back to having the data come to the user instead

Grandma

- Grandma’s on line. This means that the industry is designing systems and services for ordinary people
- The focus is on applications and devices, not on infrastructure and architecture, the domain of systems research
  - The cause is largely marketing, the result a proliferation of incompatible devices. You can’t make money on software, only hardware, so design a niche gimmick, not a Big New Idea
  - Programmability once the Big Idea in computing has fallen by the wayside
- Again, systems research loses out.

Things to build

- There are lots of valid, useful, interesting things to do
  - If the field is moribund, it’s not from a lack of possibilities
- Only one GUI has ever been seriously tried, and its best ideas date from the 1970s
  - In some ways, it’s been getting worse; today the screen is covered with confusing little pictures
- Surely there are other possibilities
  - Linux’s interface isn’t even as good as Windows!
- System administration remains a deeply difficult problem
- Unglamorous, sure, but there’s plenty of room to make a huge, even commercial, contribution
**Systems Software Research is Irrelevant**

**Conclusions**

- The world has decided how it wants computers to be
- The systems software research community influenced that decision somewhat, but very little, and now it is shut out of the discussion
- It has reached the point where I doubt that a brilliant systems project would even be funded, and if funded, wouldn’t find the bodies to do the work
  - The odds of success were always low; now they’re essentially zero
- The community universities, students, industry, funding bodies must change its priorities
  - The community must accept and explore unorthodox ideas
  - The community must separate research from market capitalization