Looking Backwards
The Coming Decade of BSD

George Neville-Neil

Welcome to EuroBSD 2026!

- FreeBSD 15
  - Dropped support for sparc64 and PC98
- NetBSD 11.0
  - Dropped VAX, Amiga, and Atari ST Support
- OpenBSD 9.0
  - First implementation of SMP!

Some Notable BSD Achievements

- Scaling to 32K CPU cores
- Single System Serving 10 Terabits/sec
- Always on Petabyte File Server
- Security Isolation Technology in Every Mobile Device
- Most commonly deployed IoT OS
- The most used OS technology in the world

2017 BSD Declared Dead (again)

- 64 bit inode work complete
- First exabyte scale UFS3 deployment
- Network stack librarification continues
- Integration of Concurrency Kit primitives
- BSD API Standards Published
  - LLVM Compiler Extensions Begin

2018 Linux On the Desktop

- Three new schedulers added as libraries
  - Massive Multicore (MMC)
  - Little John (Big/Little written by John Baldwin)
  - Skimpy Sched (Power aware scheduler for embedded)
  - Enhanced NUMA Awareness started in MMC Scheduler
- 1 Terabit NICs support
- VFS system packaged as a library
- MSDOSFS first FS to be turned into a library
  - Adopted as standard by most embedded systems projects

2019 Hinkley Point B Meltdown tracked to use of Linux 2.6 kernel

- All network stack components are now libraries
  - Based on pioneering work with ifLib
  - Network device drivers shrink by 2/3
- Librarification of VM system starts
- First working version of LLVM assisted system configurator
- LLDB and LLVM now default for all BSD systems and CPU architectures
- All calls to printf() replaced by DTrace debugging
- NVDIMM Support Complete
  - Libraries may now use memory that never goes away

2021 Google Abandons Go in Favor of Rust

- VM system as a library
- All user level configuration programs now consume and emit machine readable output
- All BSDs now come in flavors which may or may not look like distributions
- pkg system achieves sentience and demands a vacation

2022 DragonFly Selected as Default OS on Open Compute

- GEOM and Storage Layers as a library
  - Storage drivers shrink by 2/3
- bhyve now default virtualization system on all BSDs
- Configurator can now build kernel images between 1M and 512G
  - Support for RPi10
  - Support for HAL 9000
    - Which is now 25 years late
      - Which we know is typical

2023 OpenBSD Adopted as the primary OS at NSA, GCHQ, FSB, etc.

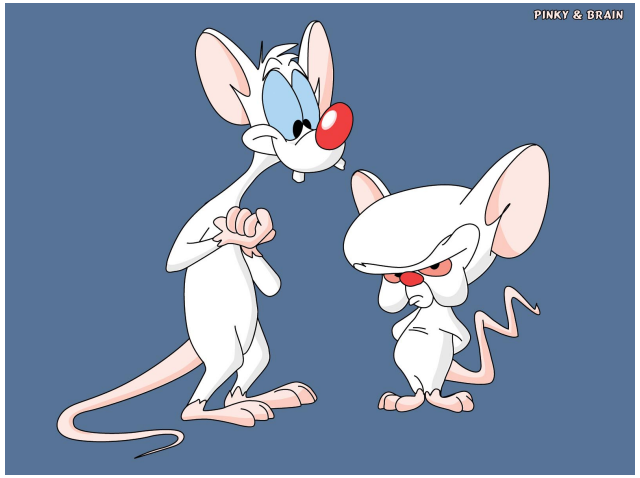
- PCI as a Fabric Support Added
- Capsicumization of kernel and user space components complete
  - OpenBSD adopts capsicum
- Configurator can remote or localize code
- Adoption of new X12 windowing system
- Java added to the base system of all BSDs

2025 Apple Donates to the FreeBSD, NetBSD and OpenBSD Foundations

- Universal Peace
- World Hunger Ends
- Realization of the Human Millennium
- Everyone gets a pony!

What do we want to achieve?

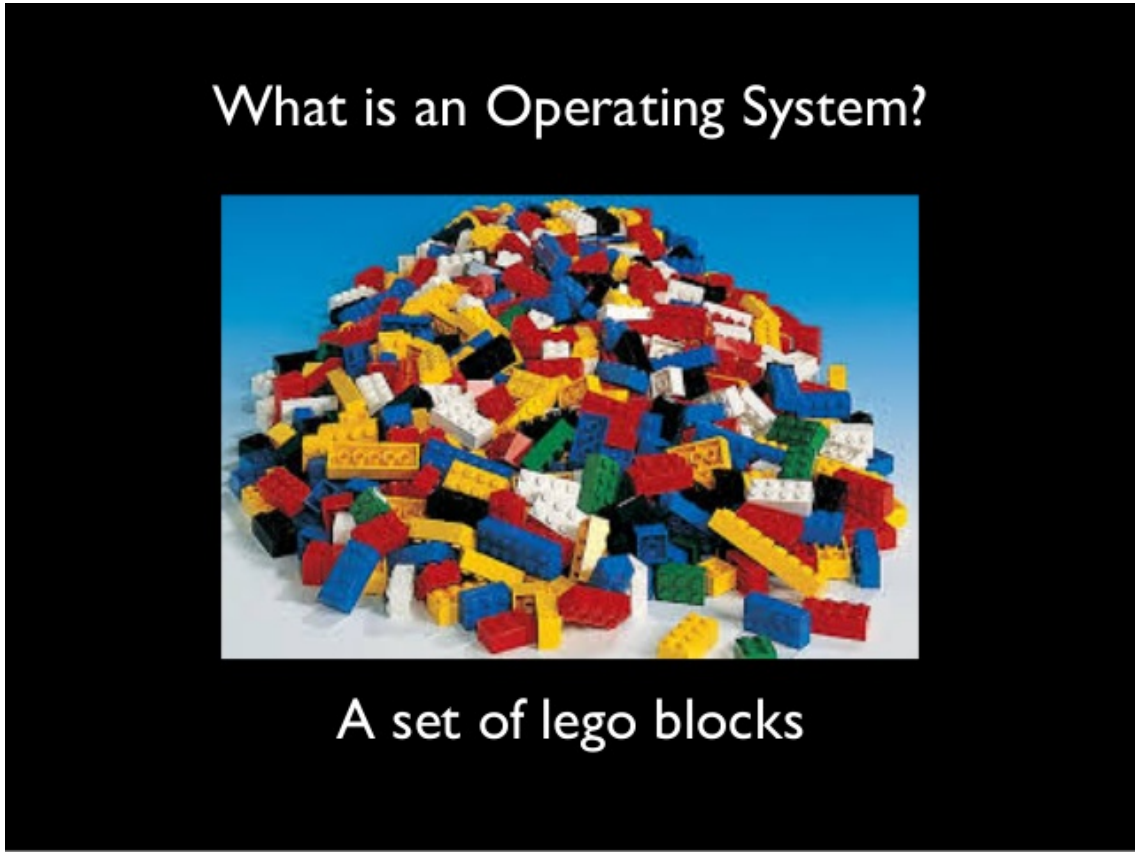
- The most used OS technology in the world
- Scaling to many more CPU cores
- Single System Serving many Terabits/sec
- Always on Yottabyte File Server
- Security Isolation Technology in Every Mobile Device
- Most commonly deployed IoT OS
  - Or would you prefer Linux or Windows to run your next automobile?



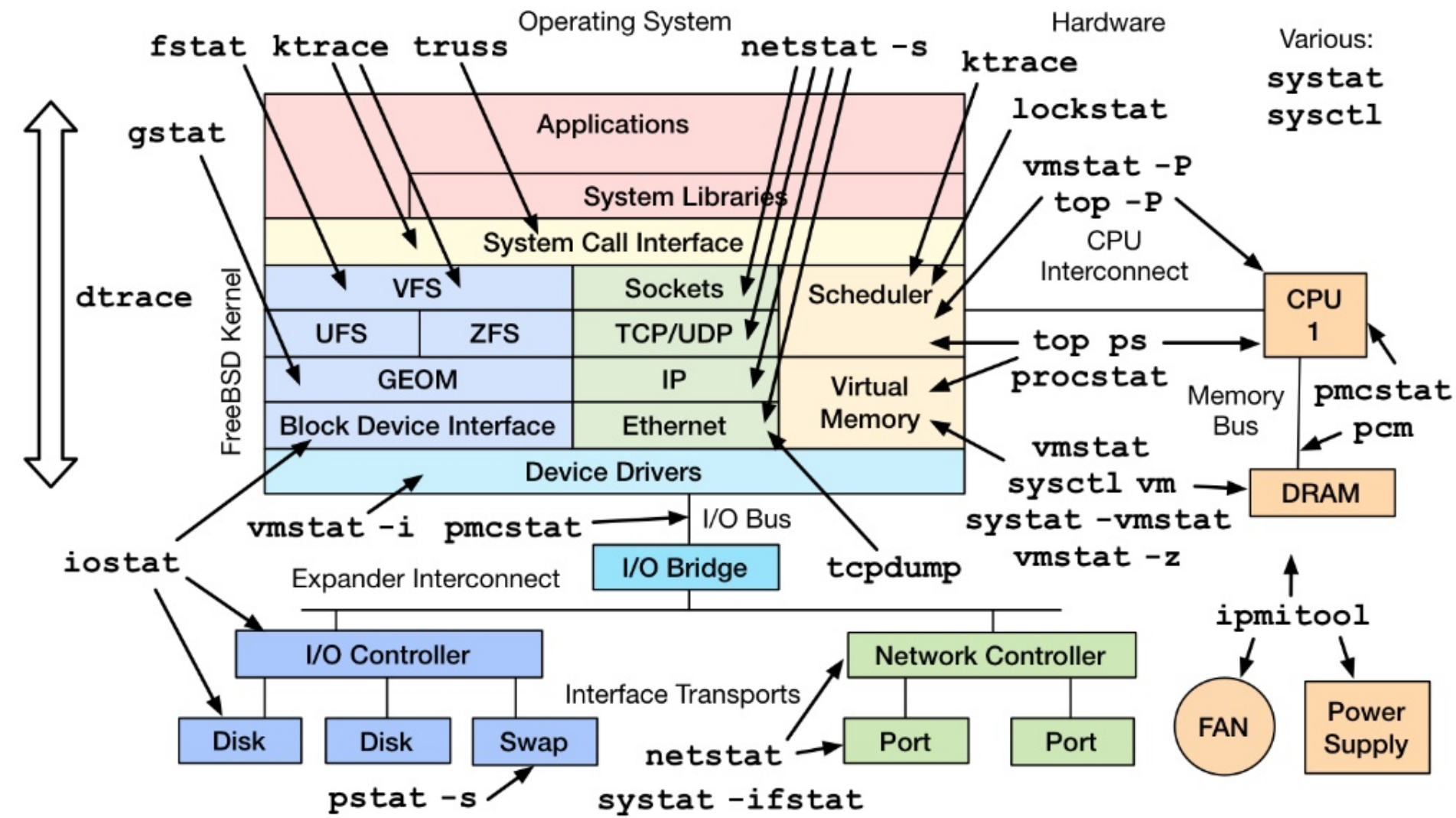
How do we get there?

- APIs
  - Design Guidelines
  - Ease of remoting
- Libraries
  - Shatter the kernel, and glue it back together
- Tooling
  - We now have the most flexible, open source, compiler on the planet
  - But we barely use its advanced features
  - Or create our own extensions
  - That, must, change…

Jordan Hubbard is Correct…



BSD Observability



Hardware/Software Co-Evolution

- CPU Extensions effect on UNIX
- NVME – Faster than SSD
- NVDIMM – Memory that never goes away
- More cores (18/36 available in 2014)
- More caches (128 MB of L4 will available on SkyLake)
- Faster NICs
  - Terabit is not as far away as you think

What was UNIX written for?

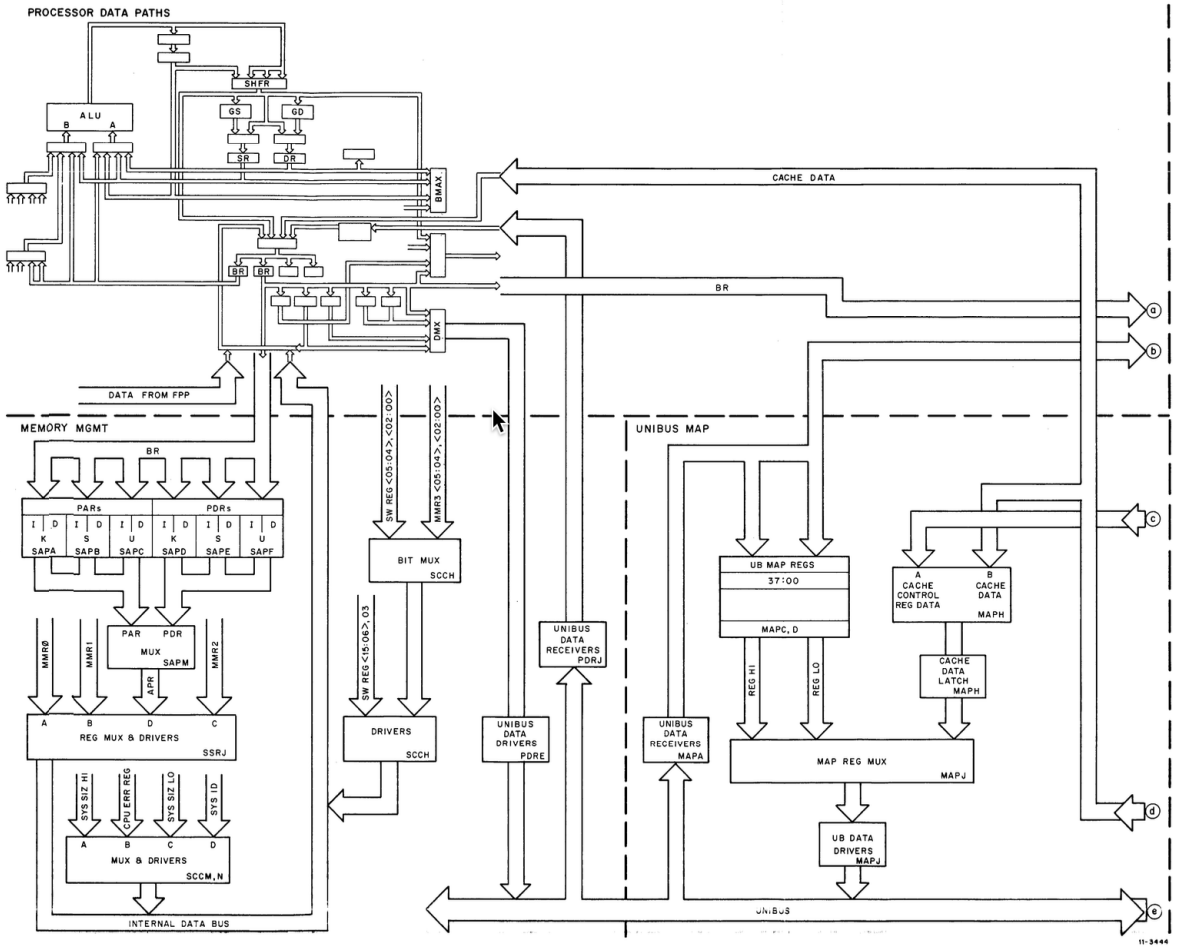


Figure A-2 PDP-11/70 Data Paths
Block Diagram (Sheet 1 of 2)

Hot, bed time, reading

EK-KB11C-TM-001

KB11-C PROCESSOR
MANUAL (PDP-11/70)

digital equipment corporation · maynard, massachusetts

A company that cared

KB11-C PROCESSOR (PDP-11/70)
EK-KB11C-TM-001

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

What features are most useful? _____

What faults do you find with the manual? _____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

Would you please indicate any factual errors you have found. _____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

CUT OUT ON DOT' LINE

— — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — —

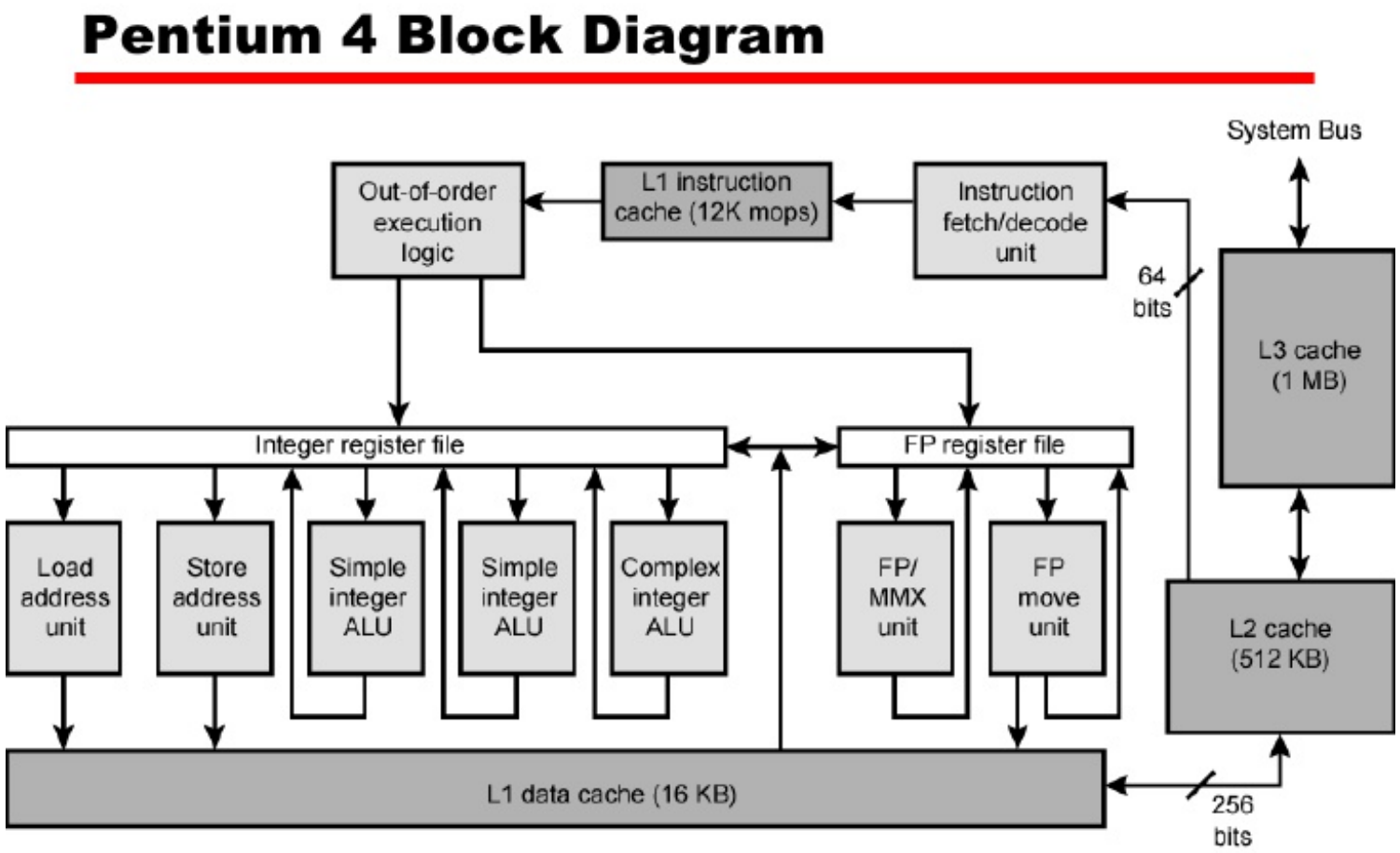
FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

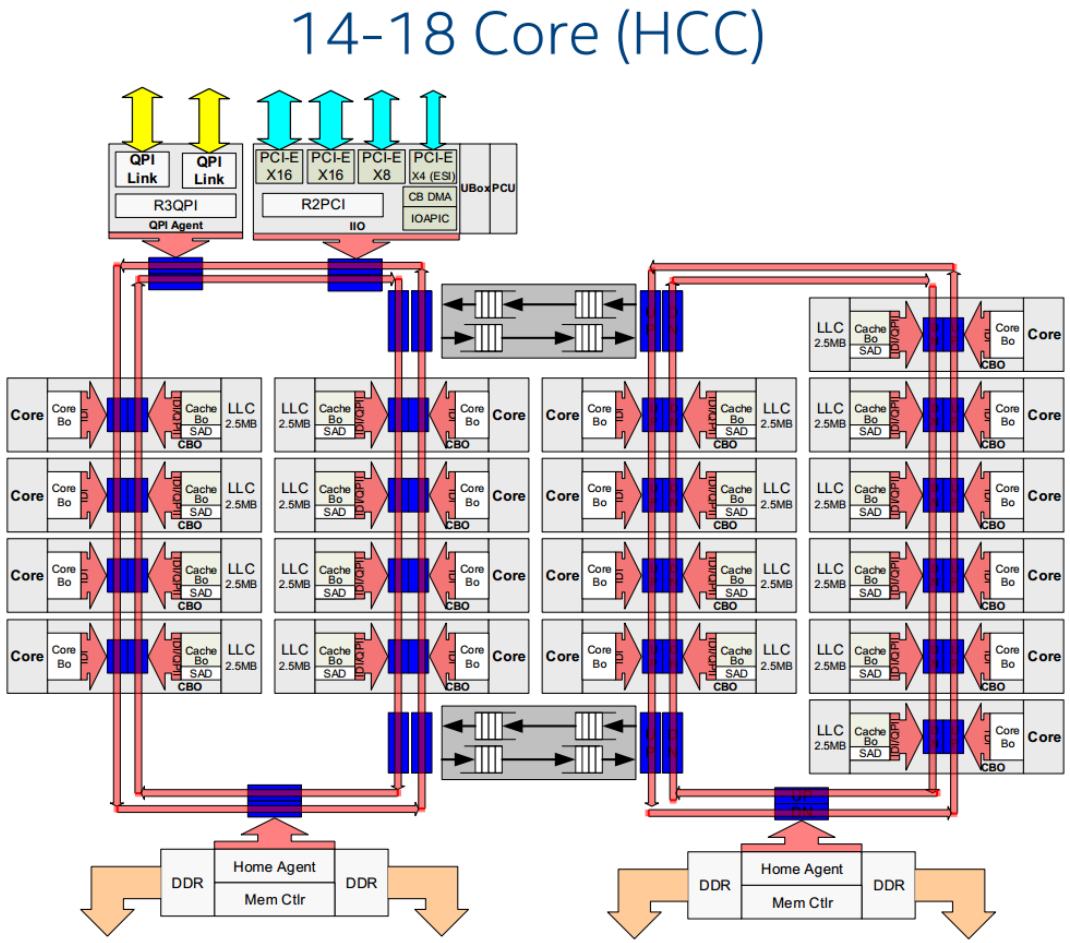
Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
146 Main Street
Maynard, Massachusetts 01754

Behold! The Pentium 4!



Current CPU Technology



Scheduler Upgrades

- Is already pluggable!
- Many more cores
- NUMA
- I/O Scheduling
- Cache Awareness
- Power
- Avoid the pitfalls

The Linux Scheduler: a Decade of Wasted Cores

NUMA Awareness

- We know the memory topology
- Memory must be allocated near the process
- And processes ought to be started where there is memory
- I/O Complicates the problem
- Extend the scheduler to know about the I/O layout

Scheduling for Cache

- Instructions are cheap, cache misses are expensive
- Now the overwhelming source of most bottlenecks
- Teach the scheduler about cache layout and constraints
- Optimize for cache coherency
- Feed hwpmc samples into the scheduling decisions

Power

- Big/Little Will Become More Common
- Need to understand the compute power of each core
- Do we schedule for…
  - Quickest to complete
  - Earliest deadline
  - Lowest power consumption

From monolith to building blocks Librarification

- NetBSD’s RUMP kernels
- libuinet
- ifLib
- Must have good API standards
- Documentation standard for APIs

Need to keep going

I want one of these!

API Design

- Regularity
- Tractability
- Composability
- Assisted by the compiler toolchain
- Withered Drivers
- Easily forwardable APIs
- Better building blocks!

API Regularity

- The position of arguments matter
- What is the verb?
- What are the nouns?
- Are we writing English or Hebrew, or Japanese or?

```
void *memcpy(void *dst, const void *src, size_t len);
void bcopy(const void *src, void *dst, size_t len);
```

API Tractability: Goldilocks and the three APIs

- Too Big
  - Most Windows APIs
  - X11 is classically terrible
- Too Small
  - ioctl() considered harmful
    - What does it mean? I can't easily tell.
    - Use as a last resort
- Just Right
  - Between 5 and 7 arguments

API Forwarding

- In 2026 all systems are distributed systems
  - It was true in 2016 but we ignored that truth
- Deep structures are hard to pack
- What if this API was an RPC?
  - Pointers become more fun to deal with
- Go shallow
- Split structures into local and remote components

API to Resource Relationship

- Passing Pointers
  - Who allocates?
  - Who frees?
- Sharing Locks
  - Who locks?
  - Who unlocks?
- More about Goldilocks
  - Too big?
  - Too Small
  - Just right?

A worked example

arpLib

NAME — **arpLib** – Address Resolution Protocol (ARP) table manipulation library

ROUTINES —
***arpAdd*( )** – add an entry to the system ARP table
***arpDelete*( )** – delete an entry from the system ARP table
***arpFlush*( )** – flush all entries in the system ARP table

DESCRIPTION — This library provides functionality for manipulating the system Address Resolution Protocol (ARP) table (cache). ARP is used by the networking modules to map dynamically between Internet Protocol (IP) addresses and physical hardware (Ethernet) addresses. Once these addresses get resolved, they are stored in the system ARP table.

Two routines allow the caller to modify this ARP table manually: ***arpAdd*( )** and ***arpDelete*( )**. Use ***arpAdd*( )** to add new or modify existing entries in the ARP table. Use ***arpDelete*( )** to delete entries from the ARP table. Use ***arpShow*( )** to show current entries in the ARP table.

INCLUDE FILES — **arpLib.h**

Literally Littered with Libraries

Optimist or Pessimist?

It has been estimated by experts that the necessary software program would involve ten million (1x10^7) or more lines of code.

Opponents point out that no such program has ever been constructed and that experience would indicate that even if it could be built, it would be rife with untestable and undetectable errors.

Proponents say the software could be assembled in smaller pieces, which could probably be tested adequately or otherwise made “fault-tolerant.”

Pervasive Tracing and Debug

- Death to printf() !!!
- Tracing Features Must Be Pervasive
- Easy to use
- Produce Machine Readable Output

How big is an OS kernel?

	Files	Lines
C	5,685	5,140,567
C Header Files	5,356	2,271,425

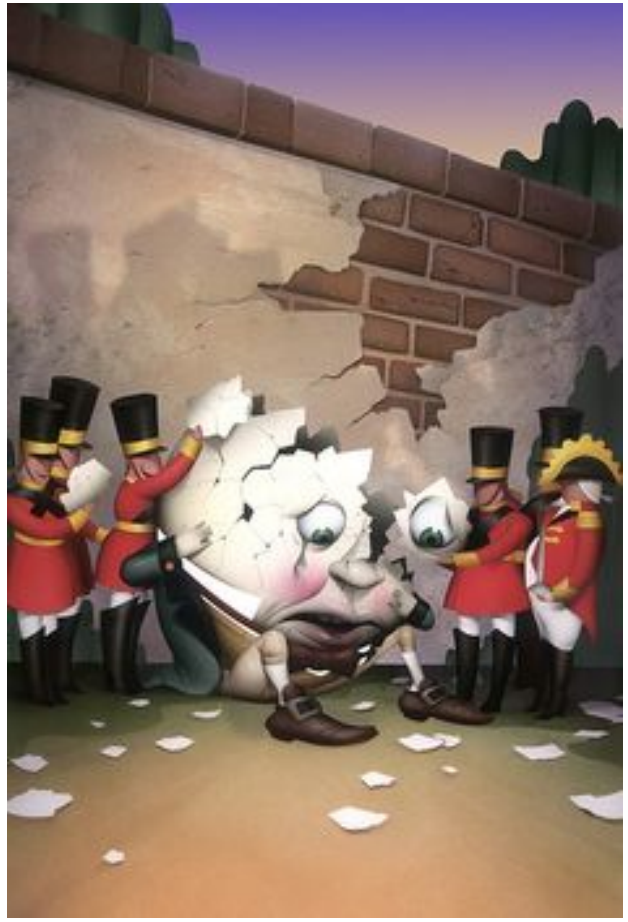
Unify the Control Plane

- Machine readable output
- Machine controllable input
- Address both humans and programmers
- Increase and improve automation
- If you're doing it by hand, you're doing it wrong!

"The study of computer science is the study of what can be automated." D. Knuth

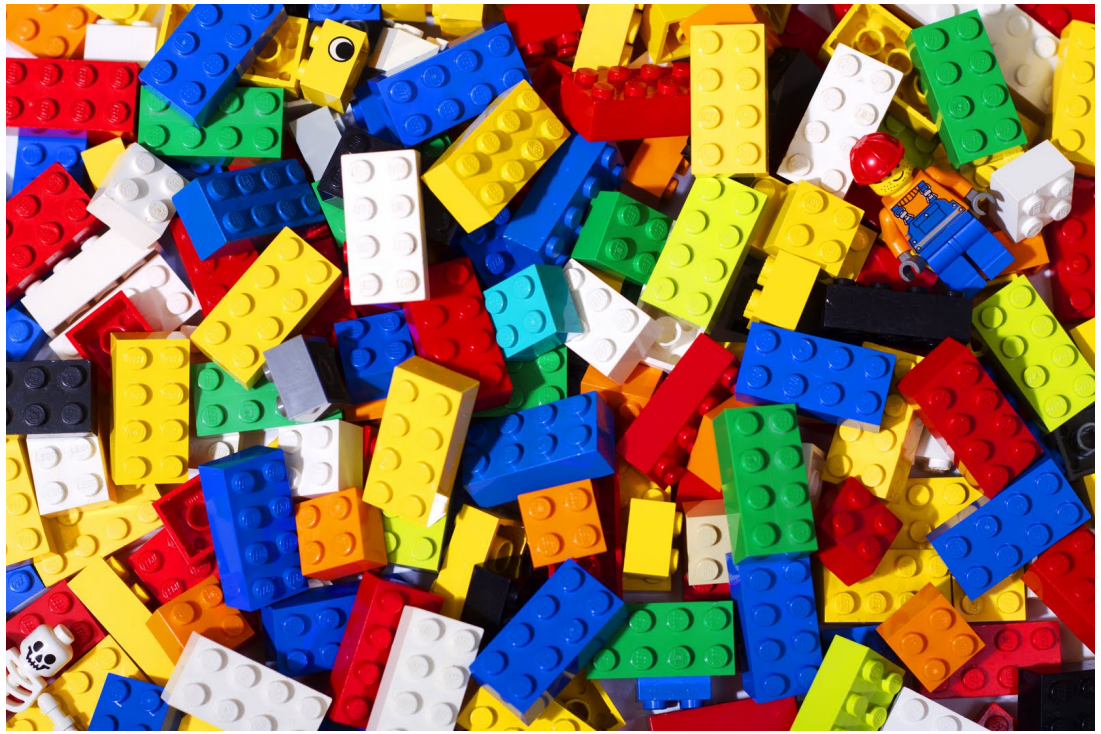
Putting the Pieces Together

- Humpty Dumpty Kernel
- Not a micro-kernel
  - Though it could be
- Need the Configurator
- Tooling, tooling, tooling
- “In the 80s people got paid to add features to the kernel, and in the 90s they got paid to take the same features out of it.” – H. Massalin



Operating Systems Are Like Legos

- The BSDs have always built solid architectures
- Small and Flexible Components
- Well defined APIs
- Built into libraries
- Come in many colors!



Comments? Questions?

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
146 Main Street
Maynard, Massachusetts 01754