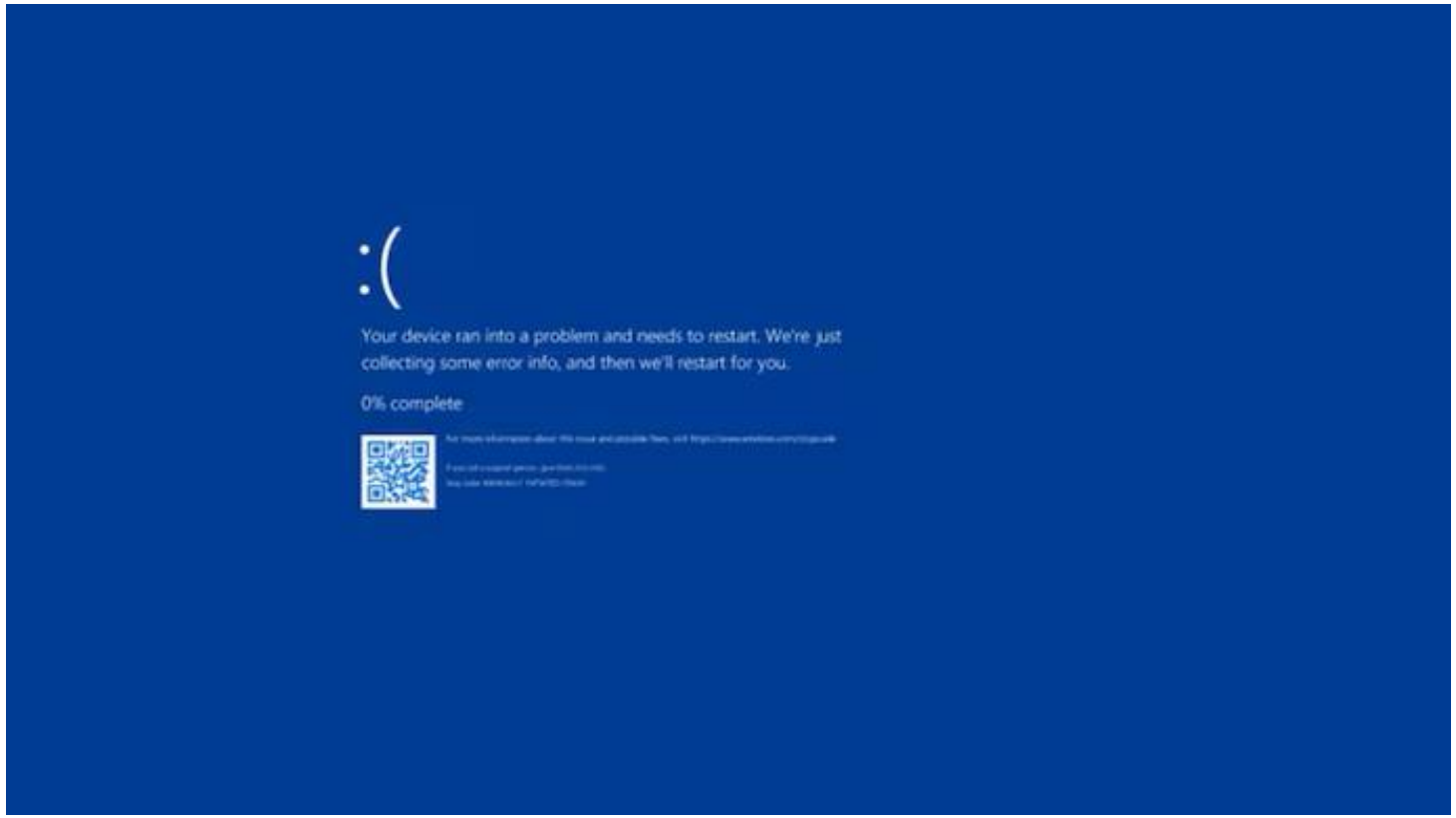
The Next Revolutions in Open Source

George V. Neville-Neil (gnn@freebsd.org, gnn@cs.yale.edu, etc.)

COSCUP 2024

Taipei, Taiwan

I guess we're feeling pretty smug…



We Shouldn’t Be

- Most modern Operating Systems built on old models of computation
  - 1970s
- Built on unsafe hardware
- With unsafe languages
  - C/C++
    - Aka Assembly Language with for() loops

A brief history of operating systems

- 1950s
  - SABRE and others
- 1960s
  - ATLAS: ASM
  - MULTICS: ESPOL
  - OS/360: ASM
- 1970s
  - UNIX: C
  - DOS: ASM
  - Home Computers: ASM
- 1980s
  - UNIX: C
  - Windows: C
  - Mach: C
  - …
- 2024
  - No real changes
  - Why?

ATLAS (Special Mention)

- System Calls
- Protection
- Nearly every feature you see in UNIX started here!
- If you want to know about the history of operating systems then read
  - **The Atlas Supervisor**
    T Kilburn, R B Payne, D J Howarth 1962

A Computer (Circa 1972)

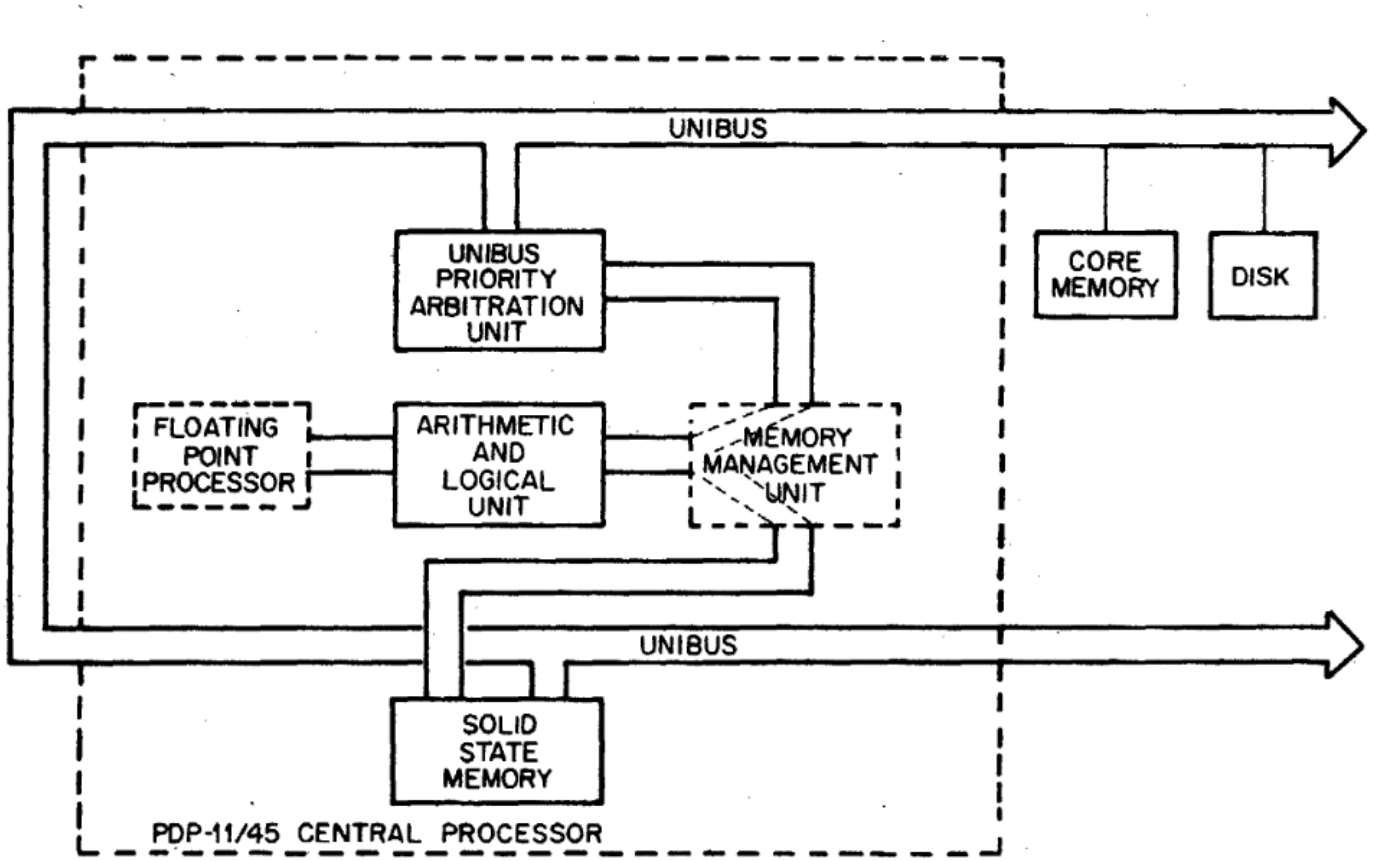
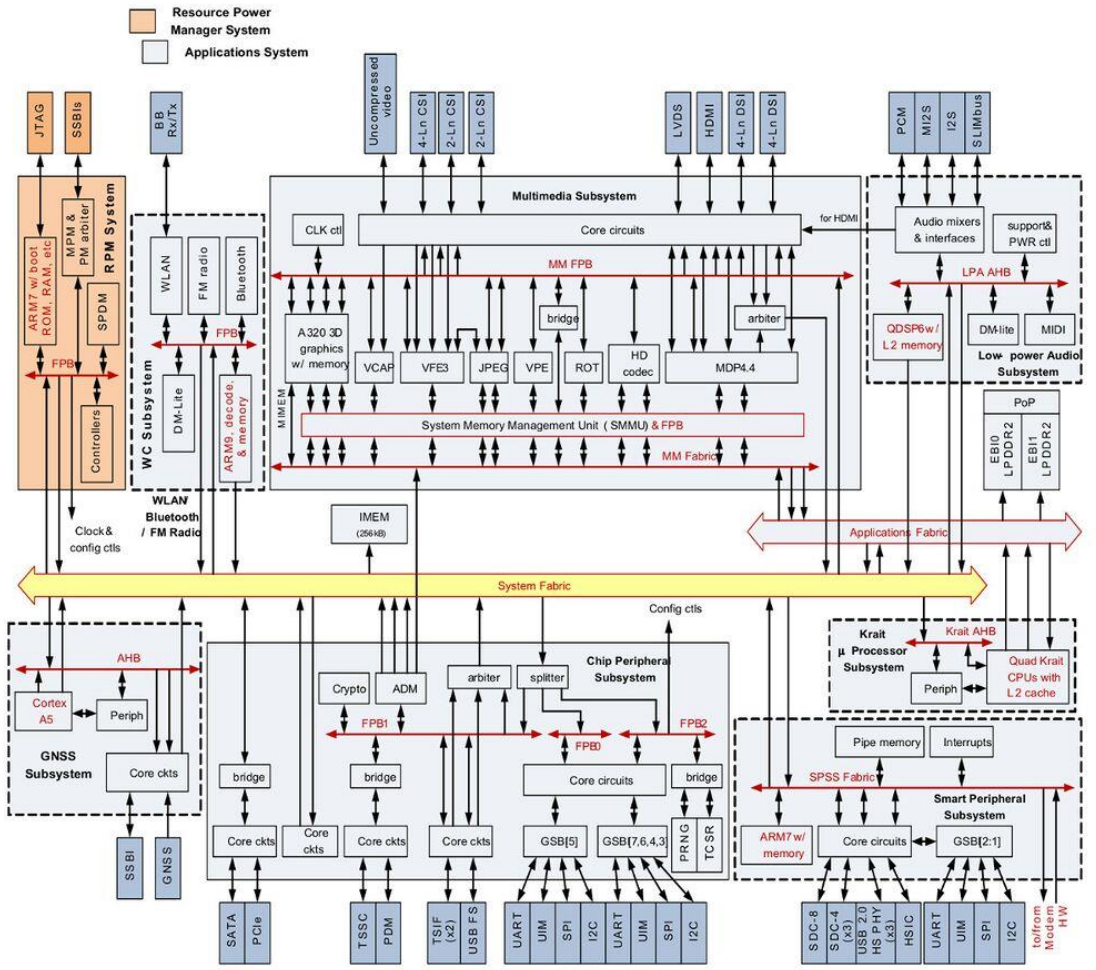


Figure 2-1 PDP-11/45 System Block Diagram

Hardware Evolution

Name	PDP 11/45	Sun 1	Pentium III	Xeon 5680	Snapdragon
Year	1972	1982	2000	2010	2023
Transistors		68,000	9.5M	1.17B	16B
Clock Speed	1-3KHz	1MHz	400-800 MHz	3GHz	3G.2Hz
Cores	1	1	1	6	8/8
Cache	N	N	256K	12MB	
RAM	256K	2MB	512M	288G	
Storage	10M	300M	10G	3T	
Feature					
Graphics?	N	Y	Y	Y	Y
GPU?	N	N	N	Y	Y
TPU?	N	N	N	N	Y
NUMA?	N	N	N	Y	Y
Network	N	10 Mbps	100Mbps	1Gbps	Wifi, Cell, NFC
Cost (2023)	$140,000	$25,000	$9,000	$5,000	$1,200

A recent pocket computer



Drivers of Change (Moore's Law and Economics)

- One computer to support many people (Cost dominates)
  - Mainframe (Company)
  - Mini-Computer (Work Group)
- One computer (CPU) per person
  - Home Computing Revolution
  - Early Workstations
  - PCs, Laptops, etc.
- Many computers per person (with many cores)
  - Cell Phone
  - Watch
  - Car
  - …

What Has Really Changed in Hardware?

- Too Many Transistors (Moore's Law)
- Same Frequency (end of Dennard Scaling)
- More cores
- More memory
- More offloaded processing
- More "features"

How Do We Take Advantage of this Embarrassment of Riches?

- Exploit the Hardware
- Try new programming models
- Retry old ones
  - Learn from the Past to Build the Future
- Languages
- Tools
- Operating Systems
- Applications

Tooling Matters

- Compilers
  - LLVM
- Debuggers
  - …
- Formal Verification
  - CoQ et al
- Tracing
  - Dtrace
  - eBPF
- IDEs
  - Eclipse
  - VSCode

The Language Explosion

- Rust
- Go
- ???
- Enabled by LLVM
  - Tooling Matters!

New Life for Old Models (3 Examples)

- Capabilities
- Micro-Kernels
- Message Passing

Capabilities

- Pointers are dangers (see first slide!)
- Capabilities confer access based on cryptographic operations
- Only the valid owner has the right to make changes.
- Increased Pointer Size (Memory, TLB, etc.)
  - Use those transistors!
- Enforced by Hardware (see CHERI)

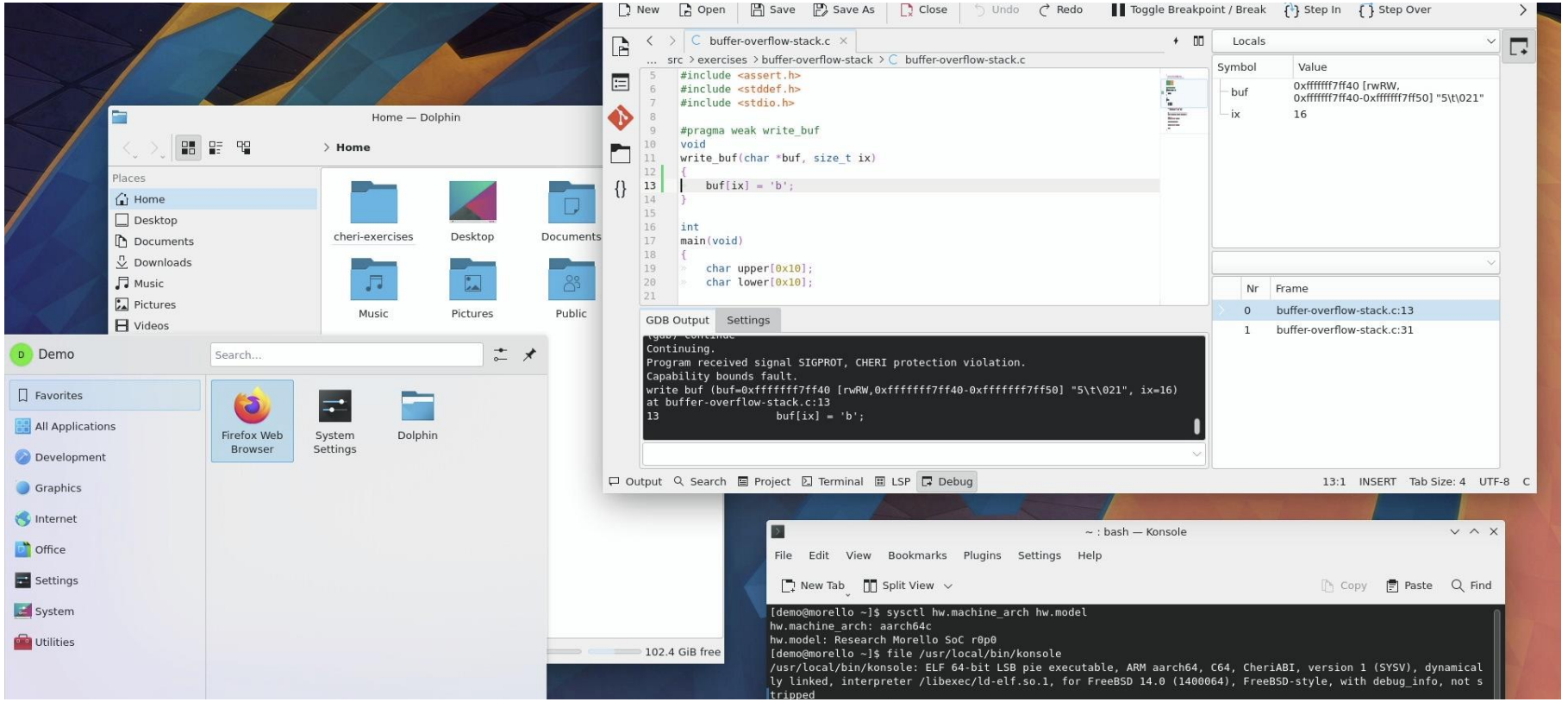
1976

2023

Then and Now

- **CAP Computer**
- CISC
- TTL Logic
- Completely Custom
- 32 bit processor
- 256 Kilobytes of RAM
- Operating System
- File Systems
- Coded in Assembler

- **Arm Morello**
- RISC
- VLSI
- Modified ARM design
- 64 bit ARMv8 Processor
- 64G and more of RAM
- PCI-E and other standard busses



Capability Enhanced, Open Source Desktop Operating System

www.cheribsd.org

Message Passing

- Mach, QNX and others in the 1990s
- Micro-Kernel Designs
- Even in macOS (BSD and Mach based) these features were removed
- Why?
  - Performance
- But on uniprocessor machines!

Message passing on a uniprocessor (You're Playing against Yourself!)

On a multiprocessor… (Everyone can do more work!)

New Areas to Explore

- Capability Systems for Embedded
- Data Centric Programming Models
- Kernel as Database
- Isolation First

CherIOT
Capability based IOT

- Capabilities for Small Systems
- Hardware and Software
- C++/C Code
- MIT License
- Initially Developed at Microsoft
- https://cheriot.org
- https://github.com/microsoft/cheriot-rtos

Twizzler
A Data Centric Operating System

- Data Dominates not Code
- Written 100% in Rust
- Actively developed
- BSD 2 Clause License
- https://twizzler.io
- https://github.com/twizzler-operating-system/twizzler

Kernel Data as Database

- Tracing (DTrace, eBPF)
  - Shows who called whom
- Data
  - The state of the system at any point in time
- Kernel Data is Relatively Simple
  - Lists of Structures
  - Some trees, but not many
- OSDB
  - SQLite + FreeBSD
  - First paper submitted this week!

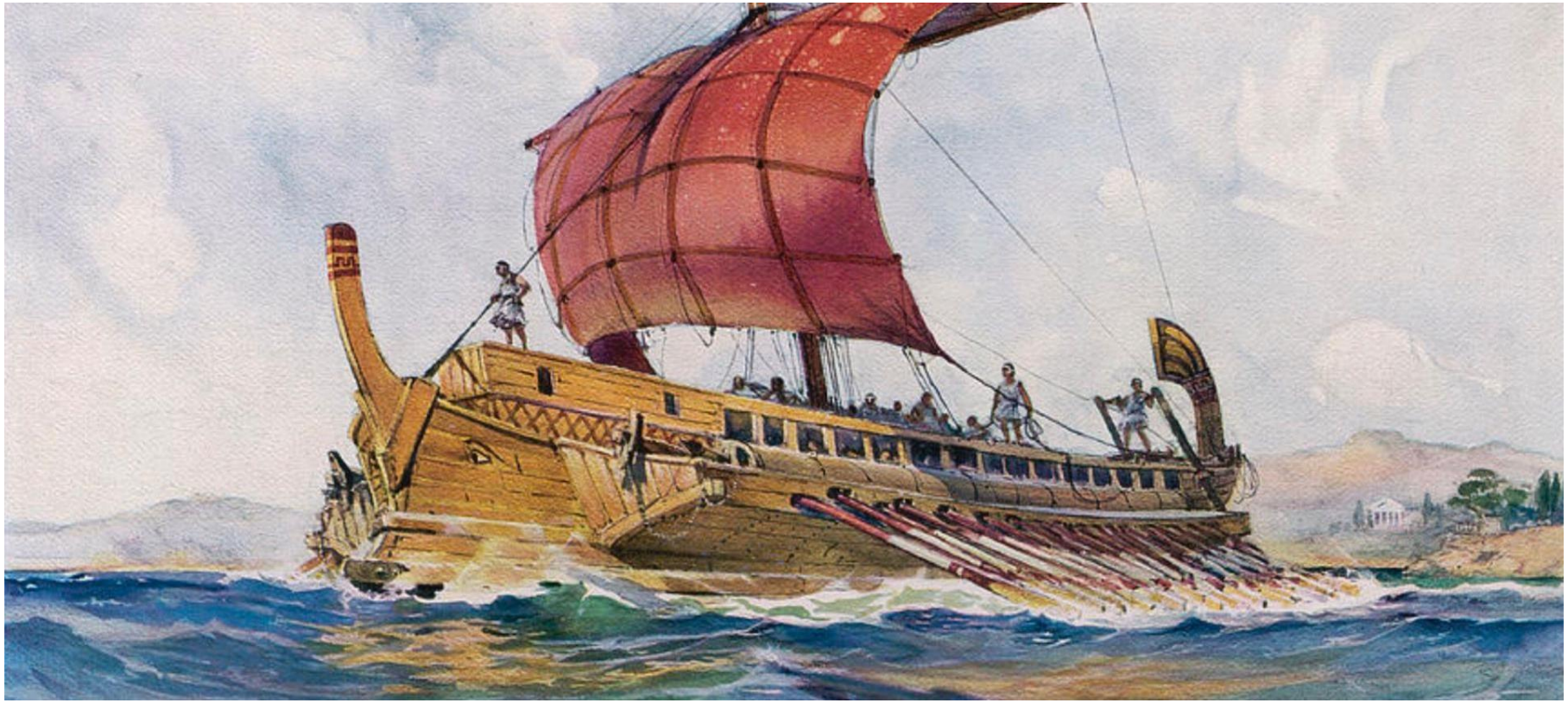
Zero
Isolation First

- No Sharing without Prior Agreement
- Femto Kernel
- Choose Your Bindings
- Open Research
- Green and Brown Field
  - Occupy BSD! (and Linux too)
- Implemented in Rust

Green Field vs. Brown Field

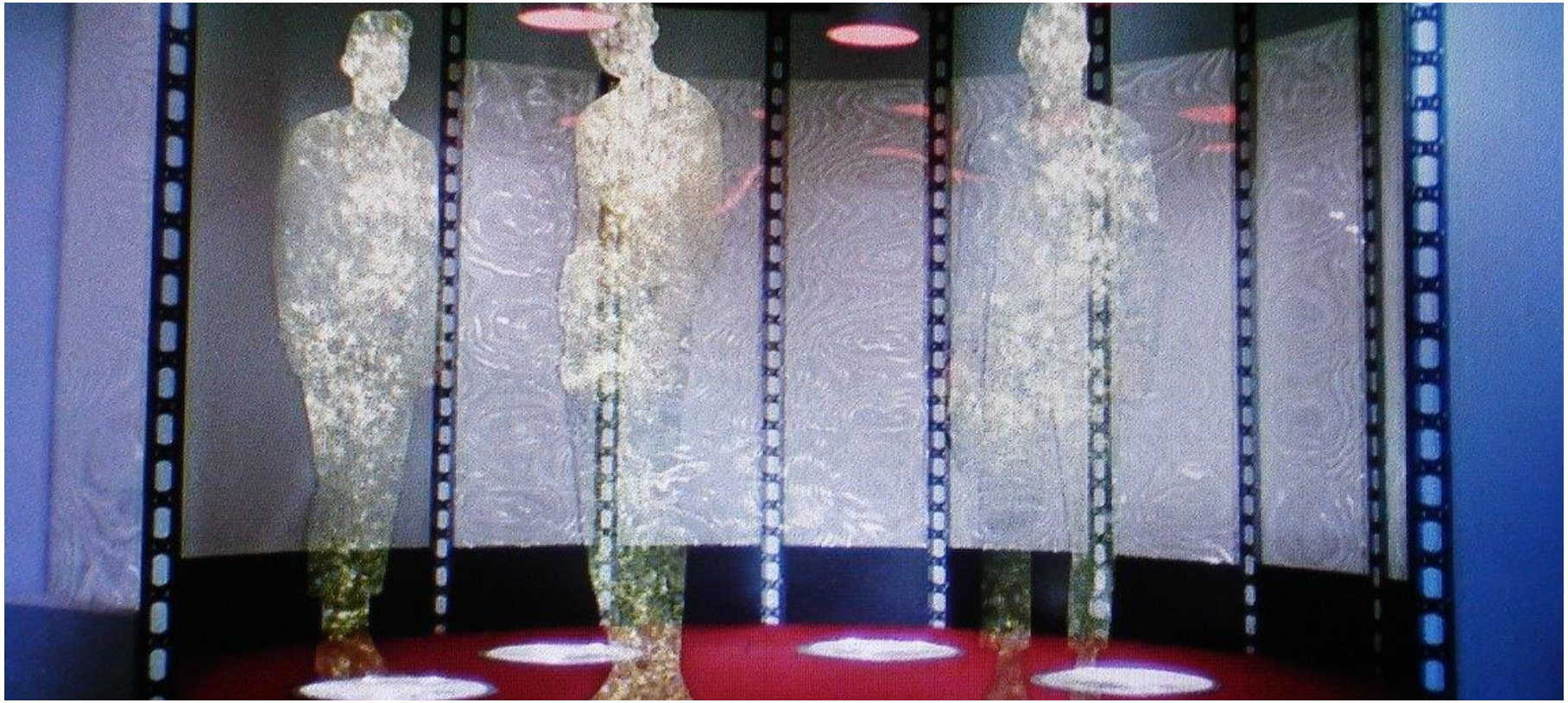
- **Green**
- Amoeba
- Sprite
- Sel4
- V

- **Brown**
- Mach
- Linux
- BSDs
- Windows Whatever…



The Ship of Theseus

If we replace every component, piece by piece, is the new system the same as the old one?



If you’re not a Greek Mythology nerd consider this…

Conclusions
Where to from here?

- Exploit the Hardware
- Try new programming models
- Retry old ones
  - Learn from the Past to Build the Future
- New Languages
- New Tools
- New Operating Systems
- New Applications