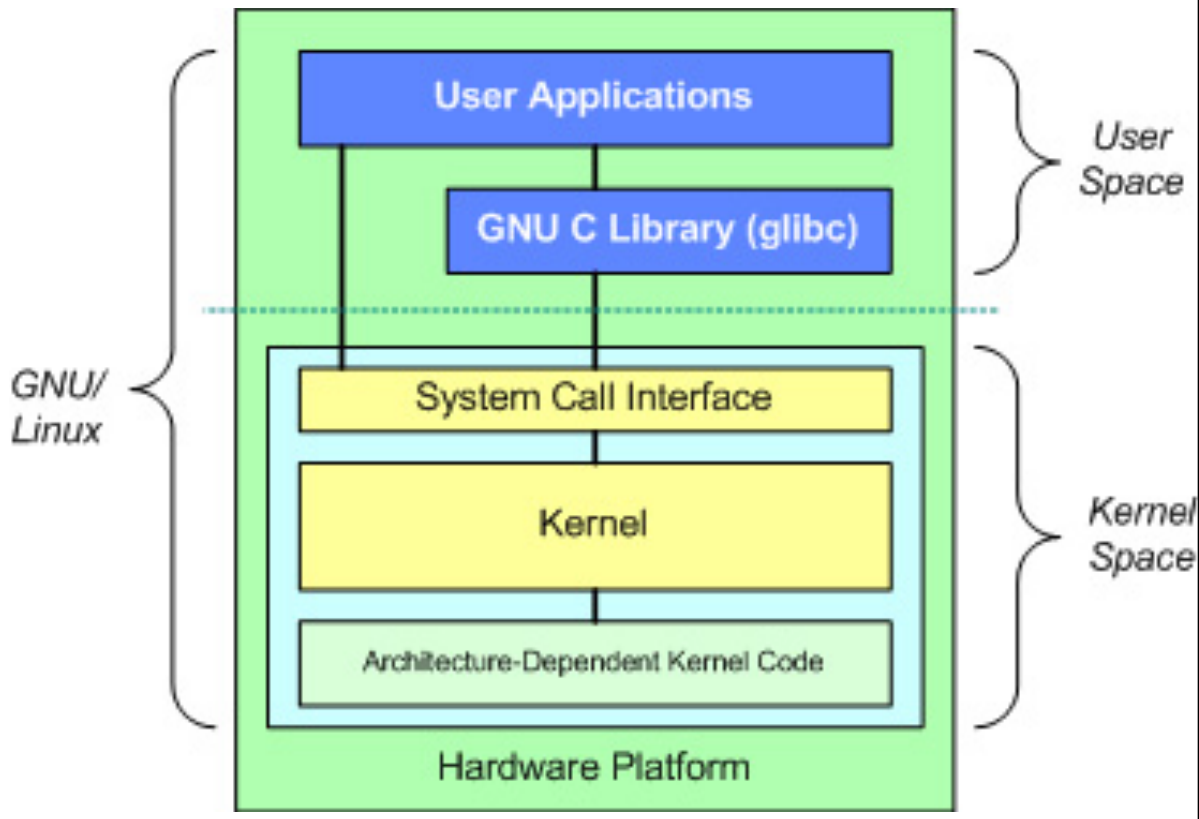
Intro to Operating Systems

Paul Gazzillo

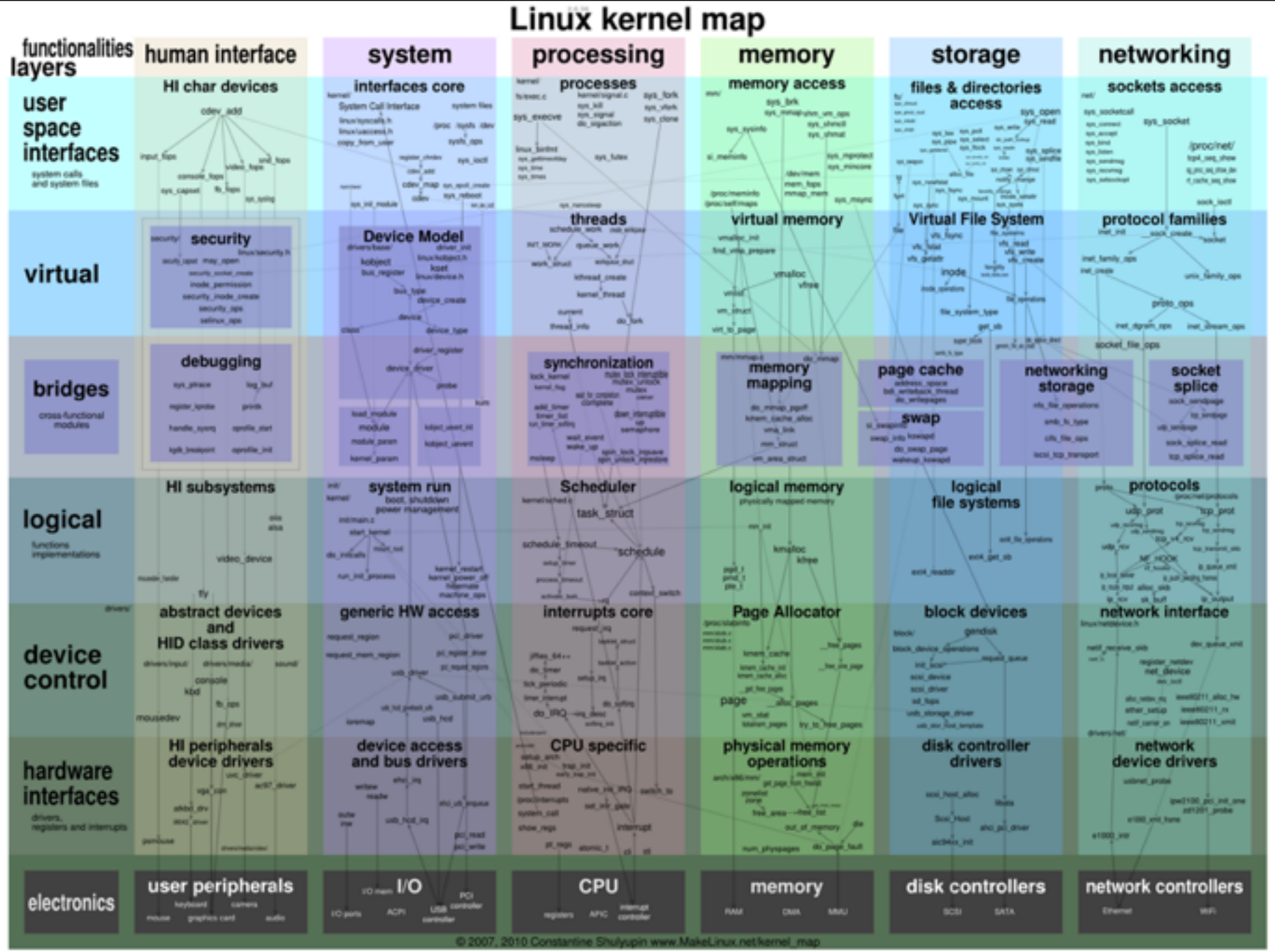
Computing Infrastructure: Layered Design

- Hardware (ask an ECE)
- Firmware (BIOS, UEFI, etc)
  - Programs on ROM, can boot OS
- Kernel
  - Abstracts away hardware differences
  - Manages access to RAM and processor
- Systems software
  - Libraries, e.g., glibc (printf, etc)
  - Tools (shell, compiler, linker, loader, etc)
- Application software
  - Word processors, games, browsers (?), etc



Kernel

- Abstracts away hardware differences
  - Analogy: electric vs gas-powered car
    - Underlying technology is different
    - Interface is the same to user: steering wheel and pedals
  - Permanent storage
    - USB flash, SATA solid state drive, EIDE spinning hard drive
    - Same interface to programmer: open, read, write, etc
- Manages access to resources
  - Virtual memory: applications request access to use more memory
  - Processor time: kernel schedules programs to share processor



http://www.makelinux.net/kernel_map/

What Is an Operating System?

- Vote
  - Kernel alone?
  - Kernel, systems software?
  - Kernel, systems software, GUI?
  - Kernel, systems software, GUI, fortnite?
  - Something else?
- Where is line between systems software and applications?

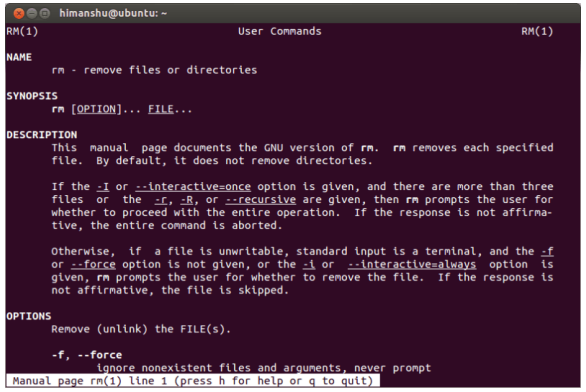
Browser Wars: Netscape vs Microsoft 1995-2001

- Netscape had large market share
- Microsoft bundles Internet Explorer (IE)
- Claims:
  - Microsoft using monopoly power to prevent competition
  - IE is as integral as memory management to OS
- https://m.youtube.com/watch?v=8Lbfcyh8dCM&t=6m30s

By Source, Fair use, https://en.wikipedia.org/w/index.php?curid=8688

Kernel vs OS

- The command-line interface is not Linux per se

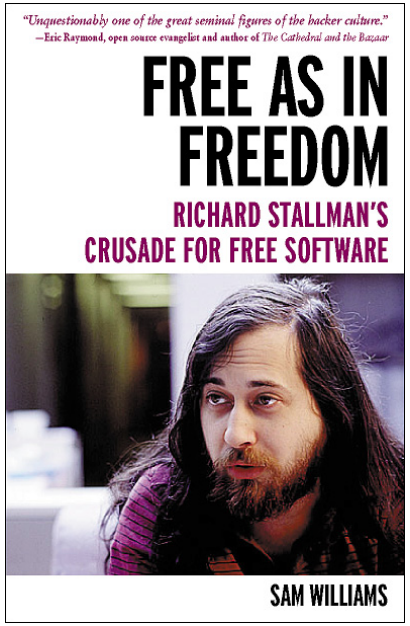


- The Linux codebase contains source code for the *kernel*
  - No command-line: bash, ls, cd, etc
  - No windowing
  - No compiler, linker/loader, libraries, etc

https://stallman.org/

GNU's Not Unix (GNU)

- 1983: Stallman announces GNU
  - Free, complete, Unix-like operating system
  - Unix owned by ATT, commercial product
- 1984: Development begins
  - GCC, glibc, GNOME, bash, binutils, coretools, etc
- 1985: The GNU Manifesto, Free Software Foundation
  - FLOSS – free/libre open-source software
- 1989: GNU Public License (GPL)
  - Free speech (libre), not beer (gratis)
- 1990: GNU Hurd kernel
  - Most of OS in good shape, except the kernel, drivers, etc



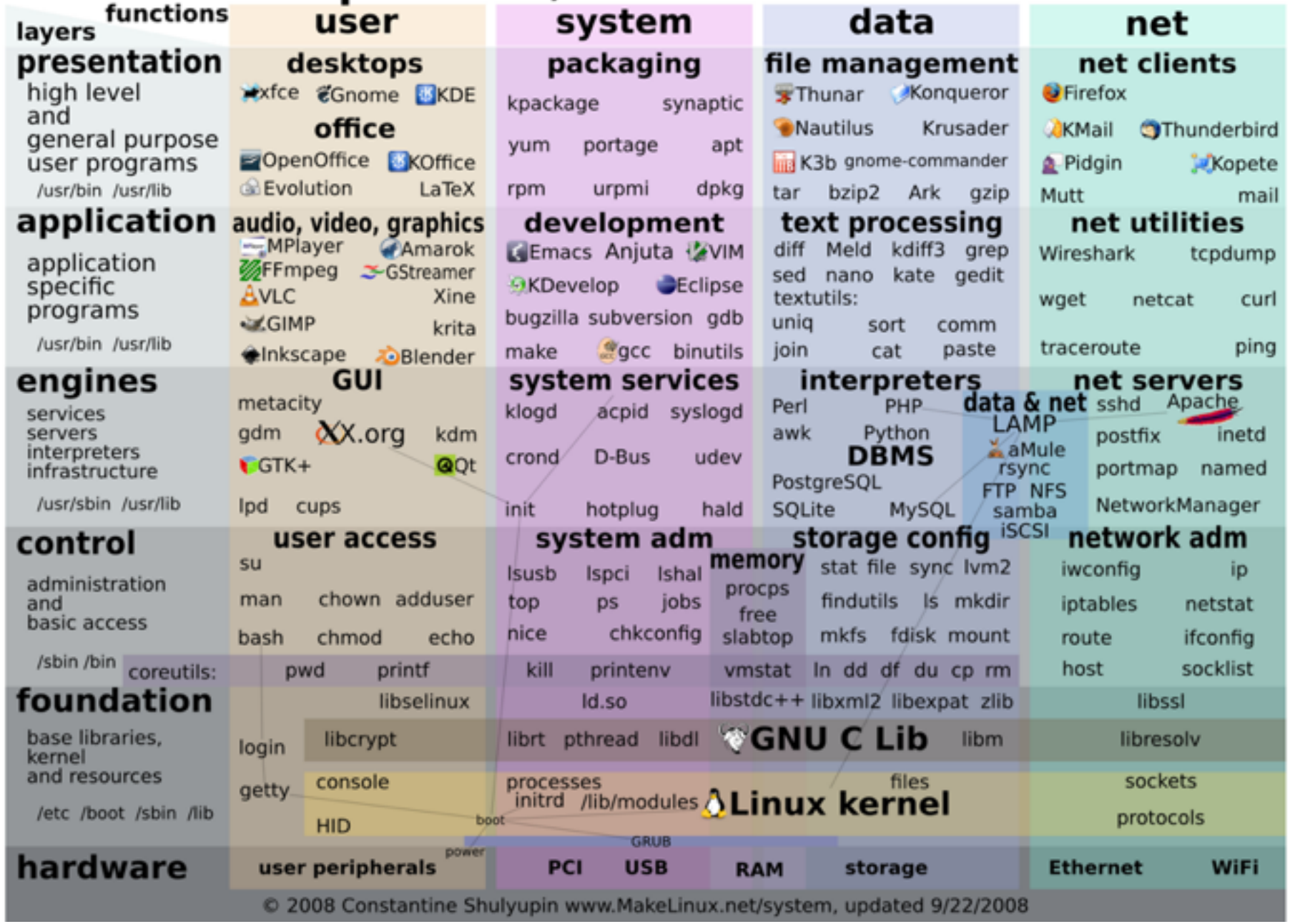
https://en.m.wikipedia.org/wiki/Linus_Torvalds

Timeline of Unix and Linux

- 1991: Torvalds announces project
  - 50k LoC
- 1992: Licensed under GNU Public License (GPL)
- 1992: “Linux is obsolete”
  - Tannebaum/Torvalds debate: micro- vs monolithic kernels
- 1993 and on: Slackware, Debian, Red Hat, etc distributions
- 2003: SCO lawsuits: claimed Unix rights used against Linux distros
- 2007: Android announced: uses Linux kernel
- 2011: Linux 3.0
- 2019: Linux is over 10mil lines of code, in 100s of millions of devices

https://www.linux-netbook.com/linux/timeline/

Map of GNU/Linux OS and FOSS



http://www.makelinux.net/system/

https://en.m.wikipedia.org/wiki/GNU/Linux_naming_controversy

GNU/Linux Naming Controversy

- Linux filled a gap in free software
  - GNU Hurd is still rarely used
- Most Linux-based OSes use GNU system software
  - Except Android, some routers, etc
- GNU/Linux

  "Most of the tools used with linux are GNU software and are under the GNU copyleft" – Torvalds, 1992

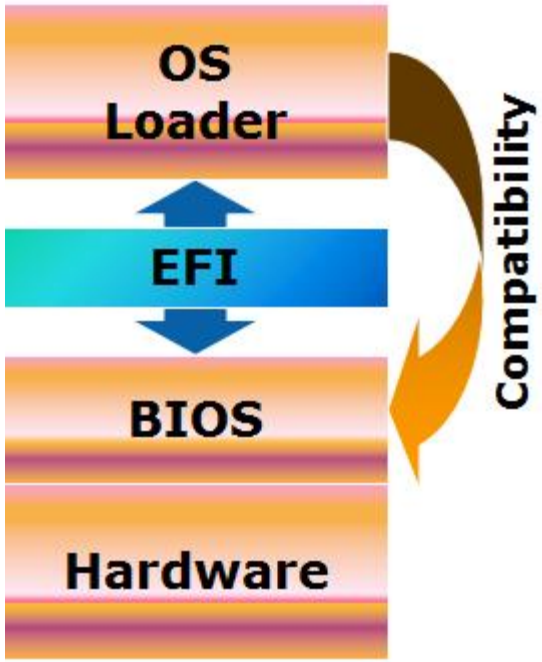
  "Today tens of millions of users are using an operating system that was developed so they could have freedom—but they don't know this, because they think the system is Linux and that it was developed by a student 'just for fun'." – Stallman, 2012
- Linux

  "This claim is a proxy for an underlying territorial dispute; people who insist on the term GNU/Linux want the FSF to get most of the credit for Linux because [Stallman] and friends wrote many of its user-level tools." – Raymond

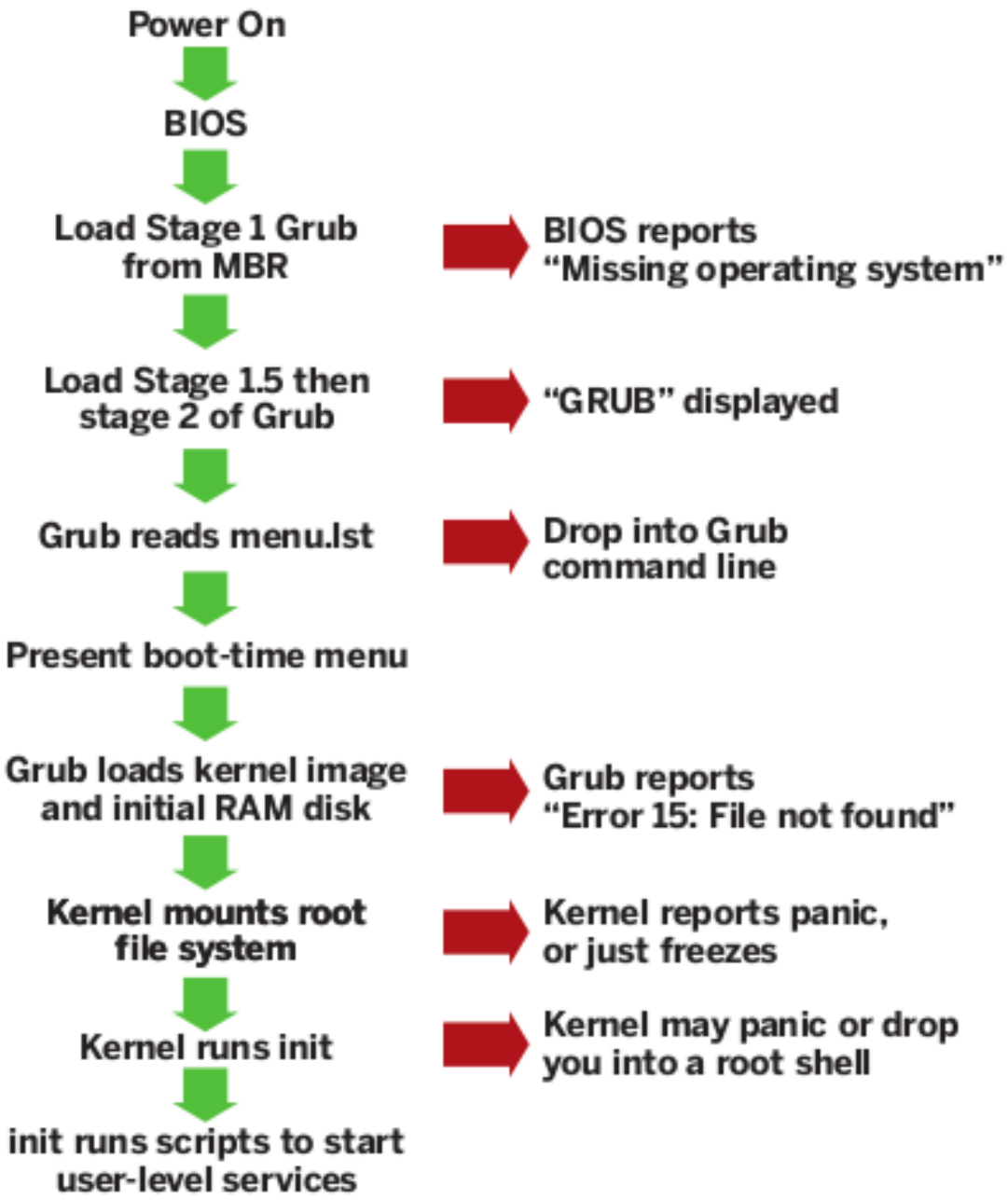
  "Well, I think it's justified, but it's justified if you actually make a GNU distribution of Linux ... because if you actually make your own distribution of Linux, you get to name the thing, but calling Linux in general 'GNU/Linux' I think is just ridiculous" – Torvalds, 2001

Booting: How Does the First Program Start?

- From “bootstrap”
  - ”Pull yourself up by your own bootstraps”
- PC firmware runs on startup
  - BIOS for decades
  - UEFI these days
- BIOS loads and runs 512 bytes of HD
- UEFI has device drivers
  - Runs bootloader program from disk
- Bootloader runs kernel



https://software.intel.com/en-us/articles/uefi-framwork-course-contents



http://linuxtroops.blogspot.com/2010/12/rhel-boot-sequence.html

Object Files: Managing Compiled Programs

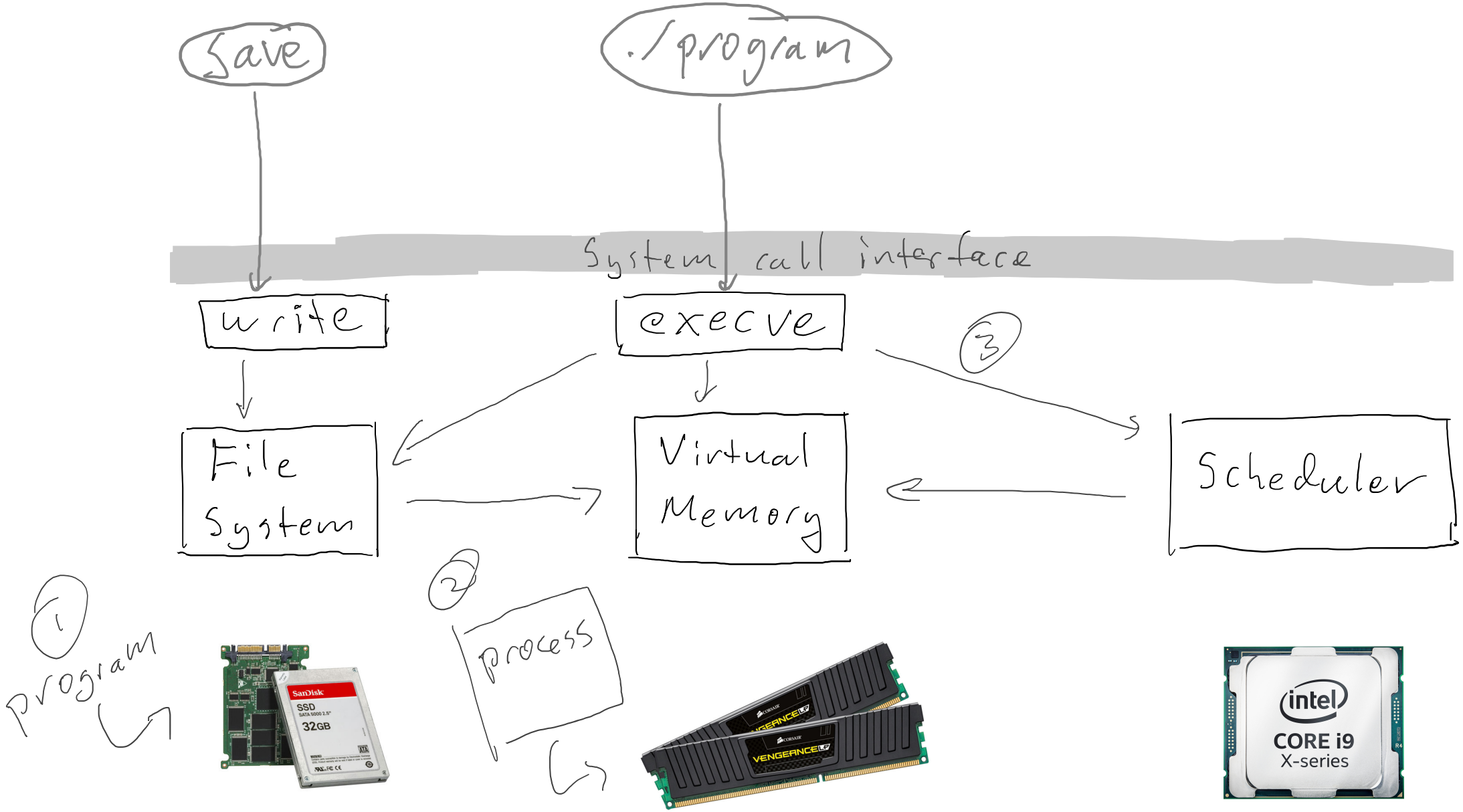
- Assembly
  - Mnemonics (ascii text) for machine instructions
  - Assembler converts to machine code
  - Machine code is raw binary interpreted by CPU
- Linking
  - Libraries
  - Bundling libraries at compile-time
- Dynamic libraries
  - Shared libraries
  - Linking at run-time

Demo: Working with Object Files

- Tools
  - gcc -S
  - objdump
- Compiling to assembly
- Linking
- Inspecting object file

Loaders: How Do Applications Run?

- Kernel initialized first process (init or systemd)
- Your program is read into memory and “main” is called
- Program vs process
  - Program: file containing instructions
  - Process: running program in memory


save
./program
System call interface
write
execve
File System
Virtual Memory
Scheduler
3
1
program
2
process
SanDisk
SSD
32GB
CORSAIR
VENGEANCE
intel
CORE i9
X-series

Demo: Loading Object Files

- Tools
  - objdump
  - ldd
- http://tldp.org/LDP/LG/issue84/hawk.html