G51CSA – Computer Systems Architecture

Operating Systems (Linux)

Red Hat

Jon Masters <jcm@redhat.com>

redhat.

About the speaker

- Jon Masters is a Senior Software Engineer at Red Hat
- History in embedded devices with Real Time requirements
- Professional author, including titles “Professional Linux Programming” and “Building Embedded Linux Systems”
- jcm@redhat.com

redhat.

Agenda

- A little background about the presenter
- Computer Systems Architecture(s)
- What is an Operating System?
- A brief introduction to Linux
- Where is Linux headed?
- Questions?

redhat.

My experiences with Linux

- Started with a single-floppy disk “distribution” (~13 years ago)
- Downloaded Slackware onto 200 floppy disks
- Slackware, Red Hat, SuSE, Debian, Ubuntu, Fedora/RHEL
- Linux User Groups, Conferences, Community
- Ported Linux to scientific instrumentation
- Worked with MontaVista
- Wrote a book on Linux
- Joined Red Hat
- Maintainer
- Another book
- Real Time Linux, Device Drivers

redhat.

Computer Systems Architecture

- A modern computer system is built from many parts:
  - CPU – 32/64-bit, big/little endian, RISC/CISC, Harvard, Von Neumann., etc.
  - Buses – HyperTransport, PCIe, PCI-X, etc.
  - Memory – Caches (I/II/III), RAM, VRAM (GART), etc.
  - Peripherals – Hard disks (SATA, SCSI, SSD),DVD, Graphics, Sound, WiFi, etc.
  - IO – External buses (USB), Firewire, “legacy buses” (serial), etc.
  - Flash memory – firmware, microcode, BIOS, etc.
- Can you name more examples?

redhat.

Computer Systems Architecture

- Time for some handy definitions:
- Architecture (arch) – a family of microprocessors that can be used to build complete and compatible(!) computer systems.
  - Intel IA32/IA64,
  - ARM,
  - Xscale,
  - MIPS.
  - Think of some examples? What was the first compatible architecture?
- Platform – a system built upon a particular configuration of microprocessor and certain other components
  - “PC”
  - Macintosh
  - iPod, iPhone...

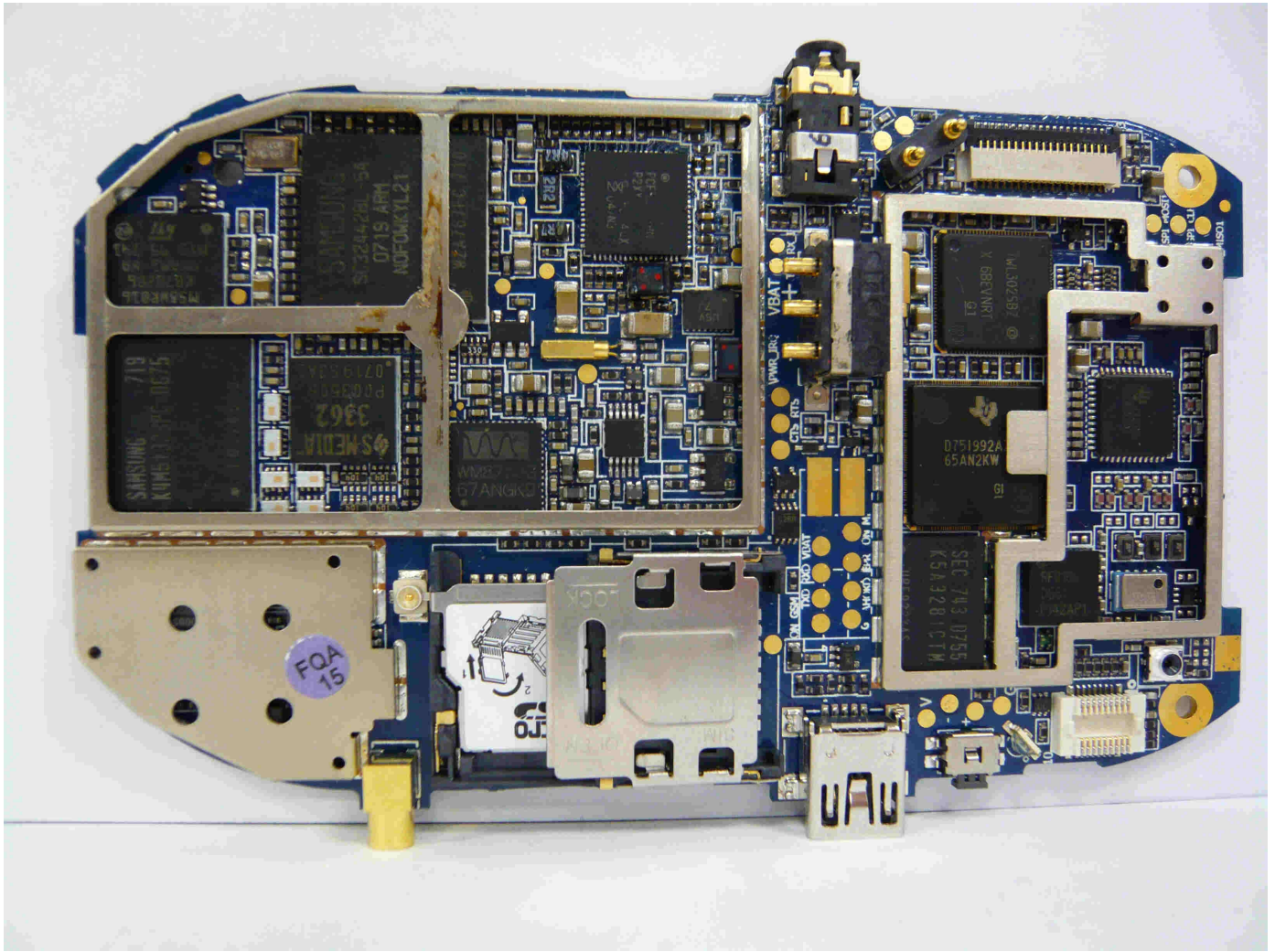
redhat.

Computer Systems Architecture

- Platforms – a standard base upon which to build Operating Systems
- Most modern platforms are heavily standardized
  - PowerMac vs. Intel Macintosh
  - iPod vs. iPhone
  - Sun OpenBoot and OpenFirmware
- “PC” is a poor example of a standard platform
  - Original IBM PC was very non-standard in many ways
  - Used non-configurable, inflexible bus technology (ISA)
  - Lack of information provided to Operating System
  - Later added EISA, PCI (PCI-X, PCIe), ACPI
- ACPI, OpenFirmware, Device Trees
- Development Boards

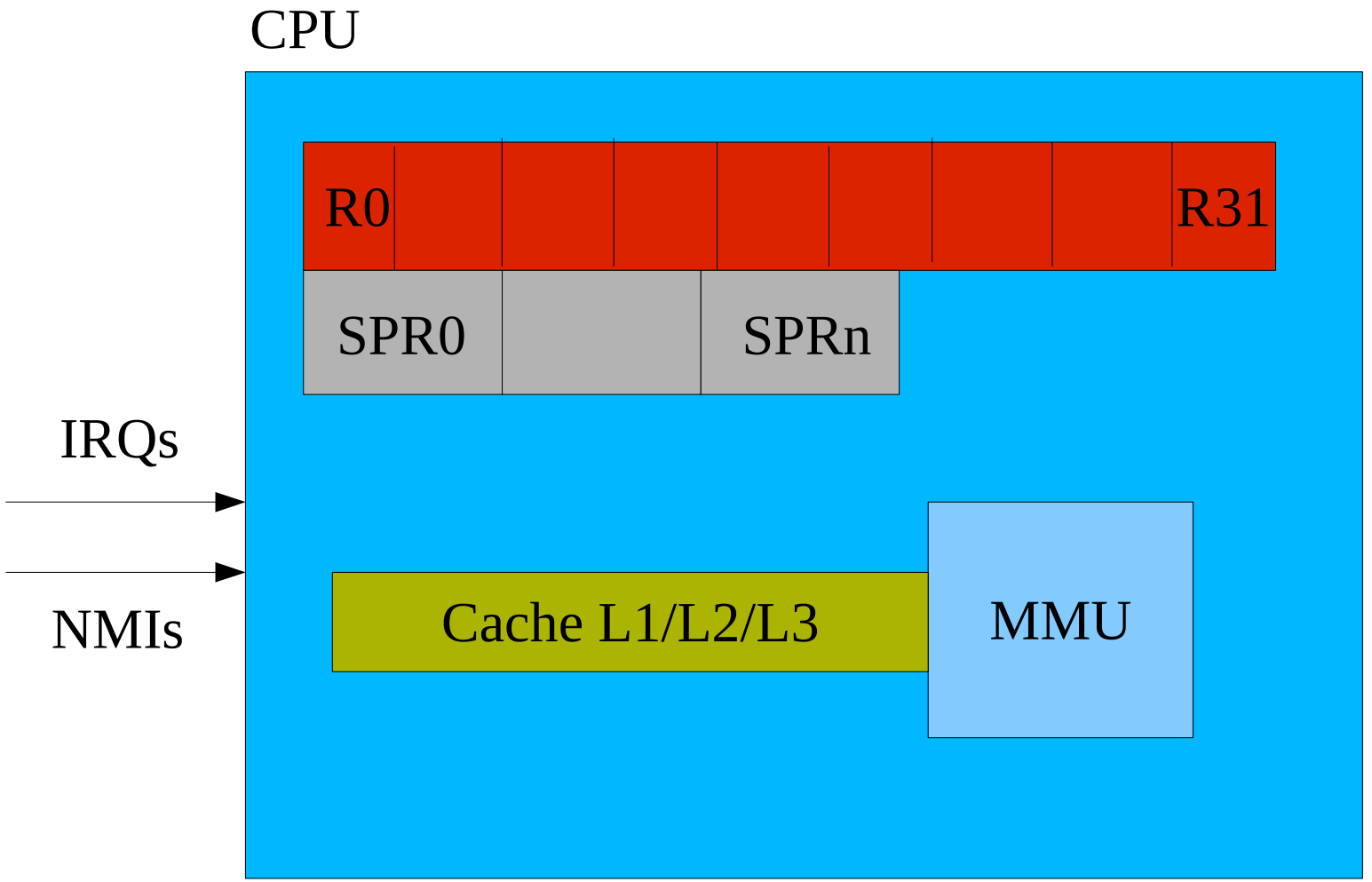
redhat.

Computer Systems Architecture



redhat.

Computer Systems Architecture

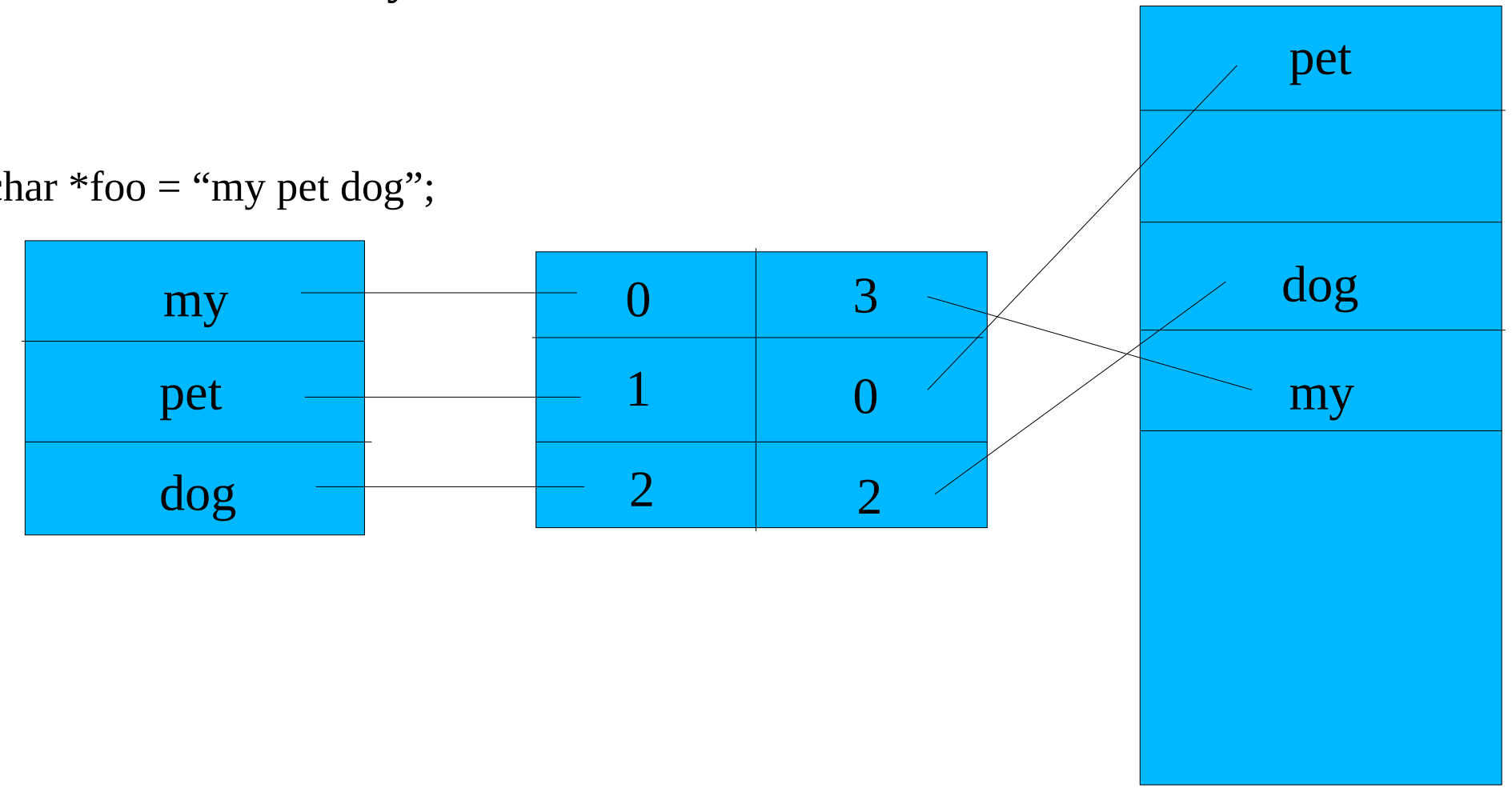


redhat.

Computer Systems Architecture

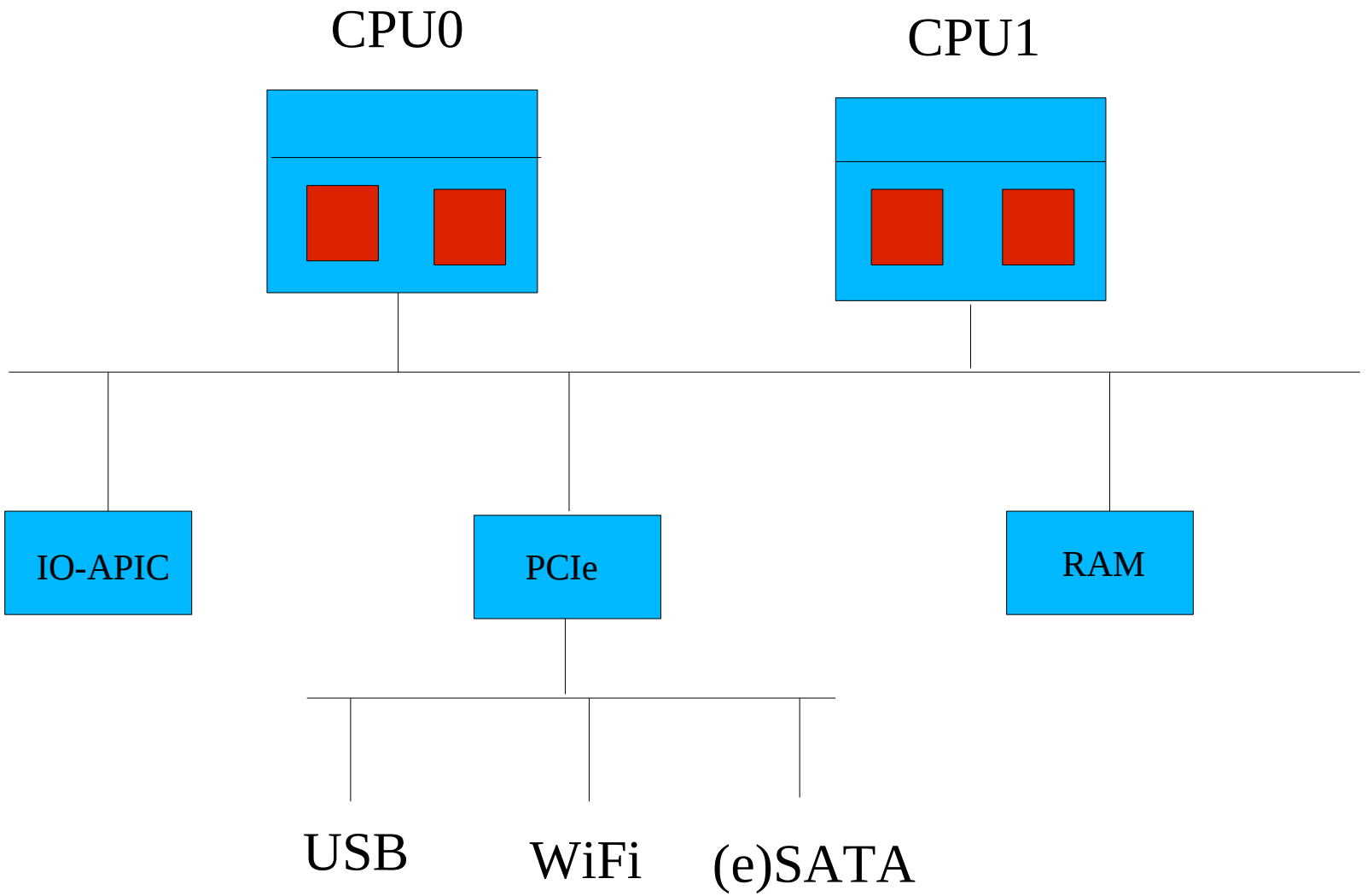
Virtual Memory

char *foo = “my pet dog”;



redhat.

Computer Systems Architecture



redhat.

What is an Operating System?

- Just a bunch of privileged library functions with supporting code
  - Bringup
  - Housekeeping
  - Applications
- A resource broker that manages access to underlying hardware
  - Finite resources
  - Virtualized/abstraction
  - Standardized interfaces
- Built for a set of platforms based on a particular architecture(s)
  - Microsoft Windows vs. Windows CE/Mobile – IA32/X86_64, PowerPC.
  - Linux – IA32/IA64, PowerPC, ARM, Xscale, S390, MIPS, etc.
  - Need for standardized platform(s)

redhat.

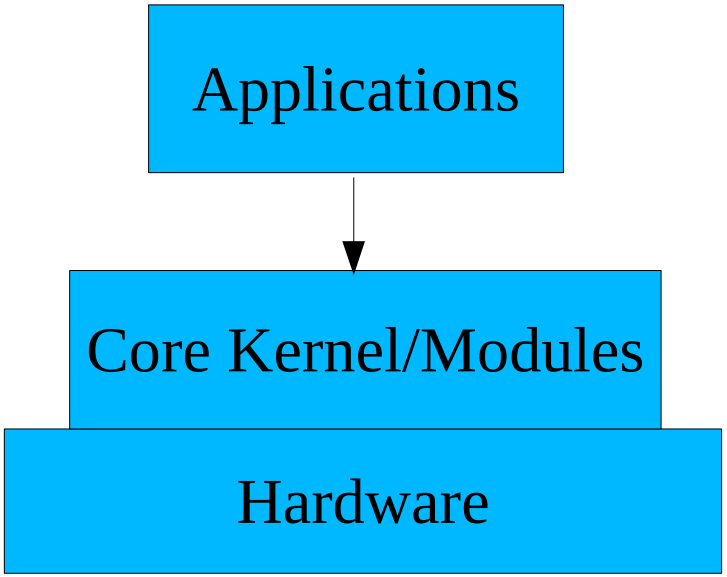
What is an Operating System?

- Reliant upon certain architectural/platform features:
  - Memory Map
  - Virtual Memory
  - Platform descriptor(s)
  - Hardware
- Must perform/provide the following:
  - System initialization
  - Device Drivers
  - Libraries
  - Graphical Desktop

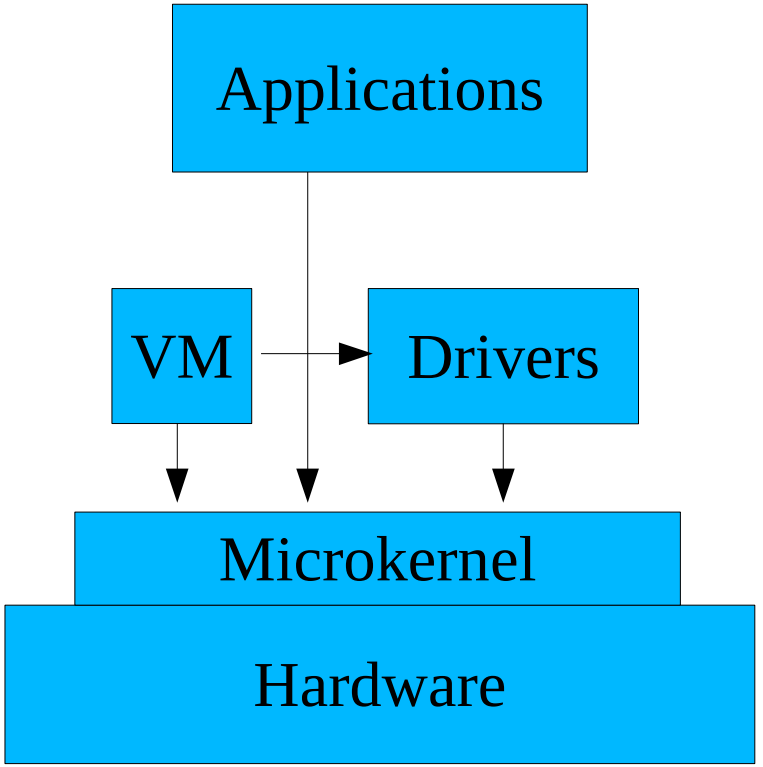
redhat.

What is an Operating System?

- Two kinds of Operating System:



Linux, UNIX



Windows, Mac OS X

redhat.

What is an Operating System?

- System boot process goes roughly:
  - Firmware (“BIOS”) handles POST
  - Bootloader loads Operating System
  - Firmware/Bootloader supply info to Operating System
  - Operating System manages resources
  - Operating System loads applications
- Core of the Operating System is called a “kernel”
  - Provides privileged functions – timers, system calls, etc.
  - Manages hardware devices
  - Schedules user applications
  - Highly event driven

redhat.

What is an Operating System?

- Monolithic vs. Microkernel
  - “Slow” vs. “Fast”?
  - “Stable” vs. “Unstable”?
  - Classical examples?
- Reality!
  - Neither exist
  - Performance?
  - Maintainability
  - Linux vs. Windows vs. OS X

redhat.

A brief introduction to Linux

- Getting ahold of Linux – who here today uses Linux?
  - What does “Linux” mean to you anyway?
  - Fedora, OpenSuSE, Ubuntu?
- Recommendations for all experience levels
  - Distributions
  - Communities
- More than Linux
  - FreeDesktop
  - Firefox
  - Thunderbird, Evolution, etc.
  - Examples?

redhat.

A brief introduction to Linux

- Traditional style UNIX-like kernel used to build a complete system
- Project started in summer of 1991 by some Finnish guy :)
- Linux vs. Minix vs. flamewars!
- Originally supported only the (shiny!) Intel 80386
- Today many millions of lines of code – co-ordination?
- Big Corporations
- Research

redhat.

A brief introduction to Linux

- Playing with the Linux kernel
  - Visit kernel.org and download the source code
  - Buy a book and sign up at LWN.net
  - Sign up to mailing lists
  - Kernelnewbies
- Kernel development
  - How does it work?
  - Complexity

redhat.

A brief introduction to Linux

- The Linux kernel source
  - Documentation
  - Include
  - Kernel
  - Arch
  - Drivers
  - Filesystems
  - ?
- Building the kernel
  - Create a config
  - Build a kernel
  - Install

redhat.

A brief introduction to Linux

- Did you know...
  - That many Linux developers have real lives?
  - That Linux is growing in popularity – why?
  - That you can contribute and learn?
  - That Linus Torvalds is a manager?
  - Magic numbers used in the kernel

redhat.

Where is Linux headed?

- Enterprise Server Systems
  - Scalability
  - Performance
  - Reliability
- Embedded Devices
  - TiVo, Routers, Mobile Phones, TVs...
  - Real Time Systems
- End Users?
  - Lots of distributions
  - Improved hardware support
  - Faster boot times
  - Feature complete

redhat.

Questions?

- #include <std_disclaimer.h>
- All views and opinions expressed are those of the author and do not necessarily represent those of Red Hat, Inc.