CE Workgroup

Status of Embedded Linux

September 2012

Tim Bird

Architecture Group Chair

LF CE Workgroup

Outline

Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Resources

Outline

Kernel Versions

Technology Areas

CE Workgroup Projects

Other Stuff

Resources

Kernel Versions

- Linux v3.1 – 24 Oct 2011 – 95 days
  - Larger due to kernel.org breakin
- Linux v3.2 – 4 Jan 2012 – 72 days
- Linux v3.3 – 18 Mar 2012 – 74 days
- Linux v3.4 – 20 May 2012 – 63 days
- Linux v3.5 – 21 July 2012 – 62 days
- Linux v3.6-rc6 – 16 Sep 2012
  - Expect v3.6 ANY DAY NOW

Linux v3.1

- Watchdog timer core
- New framework for handling power management domains was added
  - See driver/base/power/domain.c
- Multiple ARM SoCs now have device tree support

Linux v3.2

- New pin control subsystem
  - Allows control of multiple pins as named groups, with multiplexing
  - See Documentation/pinctrlt.xt
  - See ELC 2012 talk by Linus Walleij
- devfreq – DVFS for non-cpu devices
- PM QOS now supports per-device constraints
  - See Documentation/power/pm_qos_interface.txt
  - See http://lwn.net/Articles/466230

Linux v3.3

- ARM large physcial address extensions
  - See Catalin Marinas talk at ELC Europe 2011
- ALSA support for compressed audio
- New “charger manager” subsystem
  - Can partially resume to poll battery and re-suspend
- Android patches in staging
  - This is really cool

Linux v3.4

- Universal Flash Storage host controller drivers
  - See Documentation/scsi/ufs.txt
- Common clock framework
  - Unifies handling of subsystem clocks
  - See Documentation/clk.txt
- HSI (High-speed synchronous serial interface) framework
  - Used for communication between CPU and cellular modem engines

Linux v3.4 (continued)

- DMA buffer sharing API
- Remoteproc subsystem
  - Allows for control of other CPUs through shared memory
  - Rpmsg is a new mechanism for communicating with other CPUs (running non-Linux)
  - See Documentation/remoteproc.txt and rpmsg.txt

Linux v3.5

- Kernel log rework
  - Structured printk (new format), with tags
  - http://lwn.net/Articles/492125/
- Support for writing NFC drivers
- Integration of ramoops and pstore
  - Part of work to support Android ram_console
- Uprobes
  - User-space probes
  - https://lwn.net/Articles/499190/
- Autosleep

Linux v3.6 (probable)

- Android RAM console functionality integrated into pstore
- Haven’t found much else yet…

Things to watch

- Device trees
- Android features
  - Volatile ranges
  - ARM FIQ -> KDB glue
- big.LITTLE
- Single kernel image for ARM
  - Result of lots of device tree and ARM refactoring work
  - See LinuxCon Japan talk by Deepak Saxena

Outline

Kernel Versions

Technology Areas

CE Workgroup Projects

Other Stuff

Resources

Bootup Time

- Free-electrons presentation
  - Great overview of known techniques
  - http://free-electrons.com/pub/conferences/2011/genivi/boot-time.pdf
  - Free-electons service:
    - Audit, Report, Knowledge transfer
- Systemd in embedded
  - Systemd starts services and daemons on-demand
  - Saw first demo of systemd on Angstrom at ELCE 2011

Bootup Time technologies

- Snapshot boot
  - Old topic, but still very popular
  - Requires work both inside and outside kernel
    - Not much mainlined
  - See ELC 2011 presentation by Kang Dongwook
- Suspend-to-both
  - Suspend to both RAM and disk
  - If RAM loses power, can unhibernate from disk

Graphics

- Nothing new here at the API layer (?)
- 3D
  - OpenGL ES is de-facto standard everywhere
- 2D
  - Android had Skia, but is moving to HWUI
  - Other platforms can use Clutter, Qt, and X
  - Framebuffer is going away, with acceleration required for larger screens

Graphics (cont.)

- Lots of work around memory management between kernel, user-space and GPU
- Android has /dev/ion
  - A unified approach to buffer management and sharing between display, GPU, camera, codecs, etc, new in Ice Cream Sandwich
  - Replacement for pmem
- Mainline has Contiguous Memory Allocator (CMA) and dma-buf
  - http://lwn.net/Articles/468044/ - CMA
  - http://lwn.net/Articles/470339/ - dma-buf

File Systems

- Traditional flash-based:
  - UBIFS
    - Replacing JFFS2 as default raw flash FS of choice
    - Still needs some boot time improvements
  - AXFS
    - Advanced XIP File system – developed by Intel/Numonyx but never mainlined
    - Sony uses this, we've been preparing it for a mainlining effort

File Systems (cont.)

- Lots of companies using EXT4 on eMMC
- Want to optimize Linux block filesystem layers for flash
  - See Arnd Bergmann's talk at ELC Europe 2011 on filesystem performance on cheap flash media
  - See Ken Tough's ELC 2012 talk
- CE WG project to analyze filesystem performance on eMMC

Power Management

- Runtime Power Management
  - Relatively new ability to suspend and resume individual system components
  - See http://lwn.net/Articles/347573/
- See Magnus Damm's slides at: http://elinux.org/ELC_2011_Presentations
- Device power domains
  - Set of devices sharing power resources (clocks, power planes, etc.)
  - See Rafael Wysocki's talks at LinuxCon Japan 2011 and ELC Europe 2011

Power Management

- Autosleep
  - Wakelock-compatible solution by Rafael Wysocki
    - http://lwn.net/Articles/479841/
  - Rafael: "*This series tests the theory that the easiest way to sell a once rejected feature is to advertise it under a different name*"
  - Mainlined in v3.5
- Power-aware scheduling:
  - http://lwn.net/Articles/512487/

System Size

- Good talks recently:
  - Darren Hart at ELCE 2011 – poky-tiny
- Kernel size
  - Andi Kleen's Link-Time Optimization patches
  - CE WG project for kernel dynamic memory analysis
  - LLVM compilation of the kernel
- User space is memory problem area now
  - OOM killer or OOM avoidance is big issue
    - Application lifecycle
    - Application hinting
  - Volatile Ranges = the new hotness

Link Time Optimization

- See http://lwn.net/Articles/512548/
- Newer gcc (4.7) supports adding extra meta-data about routines (gimple) at compile time
- Linker can now do whole-program optimization at link time
- Andi Kleen has 74 patches that add support to the Linux kernel for LTO feature
  - Mark functions as 'visible' to avoid dead-code elimination
  - Adjust compilation flags to be consistent
  - Add dependencies to avoid conflicts for features which can't conform to LTO requirements (ftrace)

LTO (cont.)

- Cost:
  - Longer kernel builds (4x)
  - More memory during build (up to 9G required for allyesconfig)
  - Subtle bugs from optimizations
    - E.g. duplicate code elimination caused a pointer comparison failure
- Benefits:
  - Right now - NO size benefit
  - Performance: (very preliminary)
    - Hackbench – 5%, network benchmark – up to 18%

LTO (cont. 2)

- Why am I so excited about this?
- I have recently been studying automatic kernel reduction techniques
  - It is not tractable to reduce kernel manually
  - Whole system optimization is a critical part of automatic reduction
  - LTO and LLVM represent first systematic approach to problem
- Note: This work obsoletes -ffunction-sections
- Takes Linux-tiny in a whole new direction

Possible LTO benefits

- Can automatically drop unused global functions and variables
  - Could cut down on ifdefs
- Partial inlining
  - Inline only parts of a function like a test at the beginning.
- Optimize arguments to global functions
  - Drop unnecessary args, optimize input/output, etc.
- Detect function side effects and optimize caller
  - e.g. Caller can keep some globals in registers over calls.
- Detect read only variables and optimize them
- Replace indirect calls with direct calls, enabling other optimizations.
- Do constant propagation and specialization for functions.
  - If a function is called commonly with a constant it can generate a special variant of this function optimized for that
    - e.g. kmalloc_GFP_KERNEL()

Volatile Ranges

- Work by John Stultz
  - Inspired by Android feature in ashmem
    - http://lwn.net/Articles/468896/
    - http://lwn.net/Articles/500382/
- Allows cooperation between the kernel and applications on "volatile" memory usage
- Overview:
  - Application notifies kernel about re-claimable memory areas
  - Not mainlined yet

Volatile Ranges Use Example

- Application allocates memory and uses it
- Kernel notifies app that memory is running low
- Application marks areas that can be re-created (like image caches or layout areas) as volatile
  - Kernel can free those areas if needed
- If application wants to use the data, it tries to unmark it as volatile
  - If area was freed, the call fails – the application must regenerate the data
  - If area was not freed, the call succeeds – the application can use the data as is

Outline

Kernel Versions

Technology Areas

CE Workgroup Projects

Other Stuff

Resources

CEWG Contract Work 2012

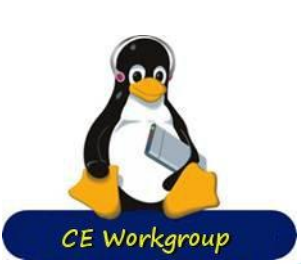
- eMMC tuning
- Dynamic memory reduction
- Mainline FIQ debugger
- ConnMan support for WiFi direct
- Improve kexecboot
- Measure systemd and udev
- UBIFS robustness work
- U-boot log buffer sharing

eMMC tuning guide

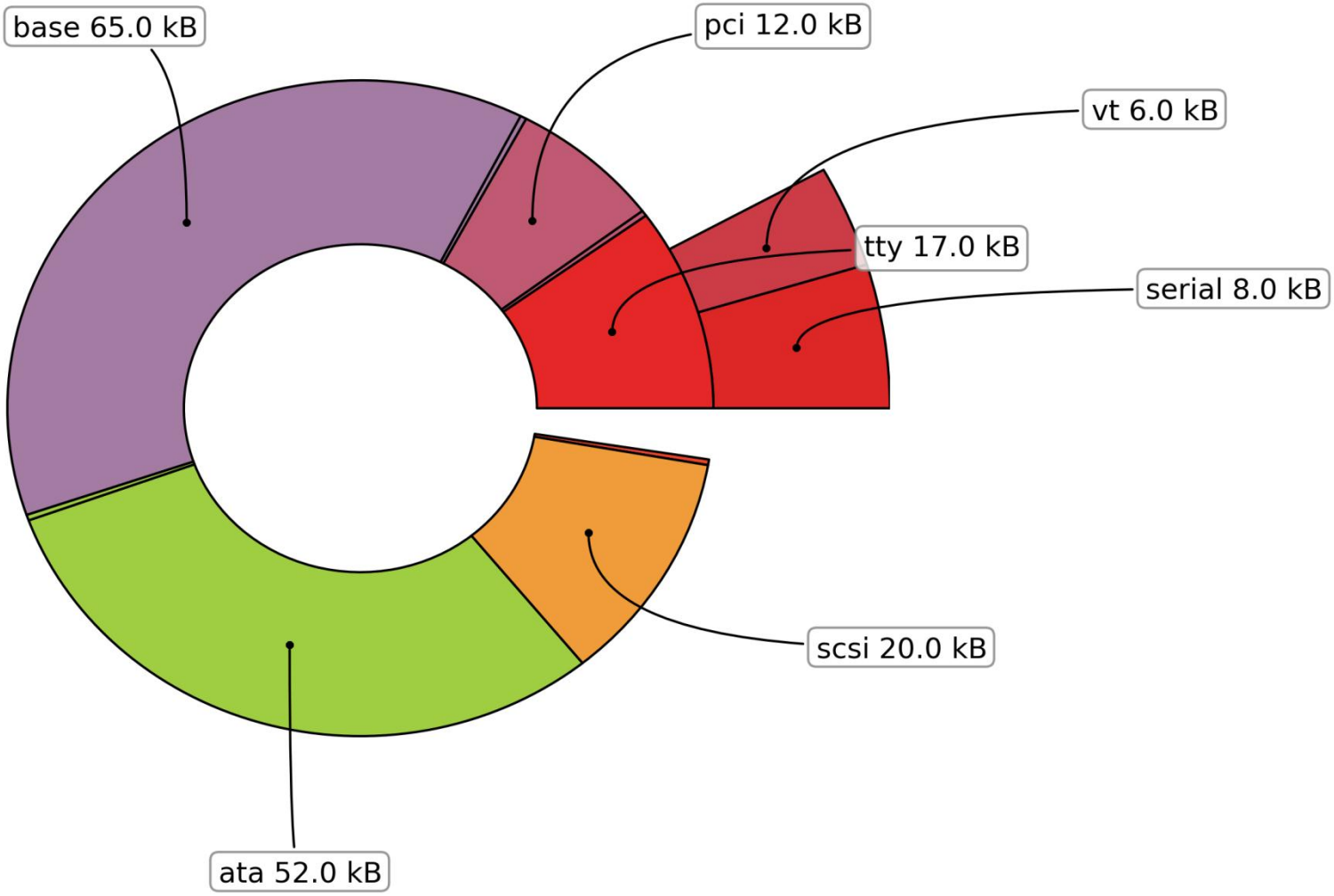
- Description:
  - This project will analyse EXT3, EXT4 and BTRFS on a variety of block-based flash parts on a few different development boards
  - Output will be a document describing best practices for tuning Linux block-based filesystems for block-based flash filesystems
- Contractor: Cogent Embedded
- Status: work has just begun

Dynamic memory reduction

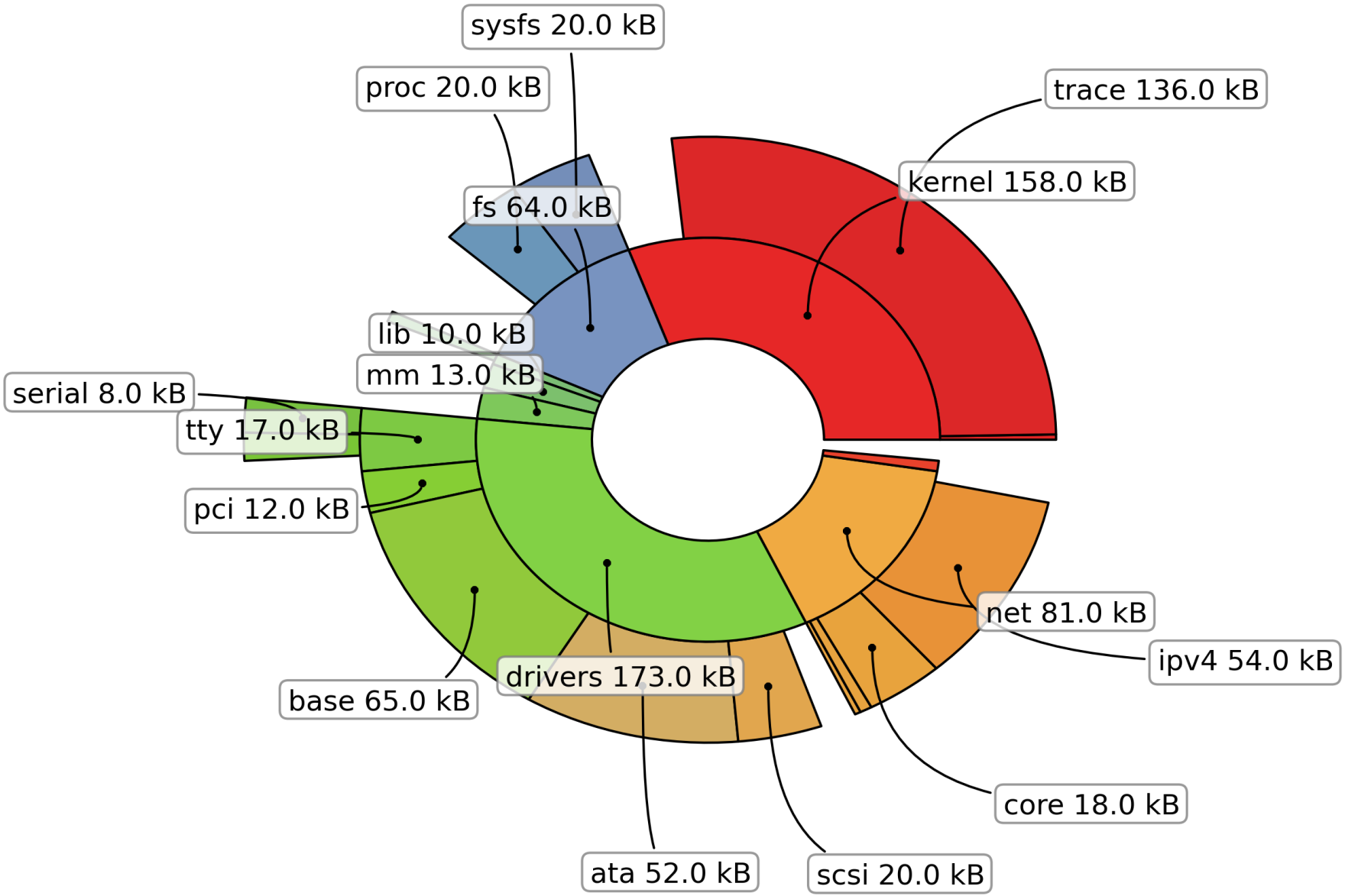
- Description:
  - Instrument and collect data on kernel dynamic memory allocations
  - Make recommendations for areas where dynamic kernel memory usage could be reduced
- Contractor: Ezequiel Garcia
- Status:
  - Work is under way to use existing kmem_events (ftrace) infrastructure to report dynamic memory usage in the kernel
  - Some patches already accepted upstream to improve memory tracing infrastructure
  - Work is in progress to create tool for visualization of kernel memory usage
  - See http://elinux.org/Kernel_dynamic_memory_analysis



Drivers kmalloc



Linux kmalloc



Mainline FIQ debugger

- Description:
  - Add ARM FIQ glue code and integrate with existing kernel debugger
  - Allows use of ARM FIQ (non maskable interrupt) to activate a kernel debugger
  - Android used it's own debug monitor, and has phones that are configured to trigger this on the earphone jack (also supplying a serial console on the earphone jack)
- Status:
  - Project is on hold because Anton Vorontsov is apparently already doing this work
    - See https://lkml.org/lkml/2012/7/30/124
    - This should show in mainline soon

ConnMann WiFi direct

- Description:
  - Add support for WiFi direct to ConnMann wireless connection manager
- Contractor: ProFusion
- Status: not engaged yet

Improve kexecboot

- Description:
  - Make improvements to kexecboot booloader
  - Support load from network
  - UI improvements
- Contractor: Yuri Bushmelev
- Status: Finalizing contract

Measure systemd and udev

- Description:
  - Measure the overhead and performance of system and udev, as used in embedded systems
- Status: Not started yet

UBIFS robustness work

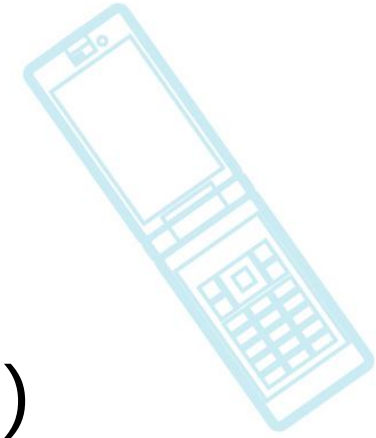
- Description:
  - Add support for "power cut" simulations to UBIFS, to allow for finding and fixing filesystem bugs that occur when power is lots
- Status: Not started yet

U-boot log buffer sharing

- Description:
  - Add support for U-Boot and the Linux kernel to share their log buffer, to allow for easier collection of joint logs
- Status: Not started yet

Long-term Projects

- Android mainline project
- Long Term Support Initiative (LTSI)



Android mainline status

- 3.3 kernel (with 12 lines of patches) boots AOSP
- eLinux status page:
  - http://elinux.org/Android_Mainlining_Project
- Was reported on at Kernel Summit:
  - http://lwn.net/Articles/514901/

Mainline status (cont.)

- Specific pieces:
  - Wakelocks => autosleep
  - Ashmem => (partly) volatile ranges
  - Ram console => persistent RAM
  - Android USB gadget driver
  - Alarm-dev => POSIX alarm timers
  - FIQ glue code (in progress)
  - GPIO timers => LED triggers (??)
  - Low memory killer => vmevents (??) in progress

Mainline status (cont. 2)

- What's not been done?
  - Logger – a few cleanups, but nothing to generalize it for other users
  - Binder – a few people talking about
  - IO memory allocator
    - Work in progress to adopt features into dma-buf
  - Network security – may stay out-of-tree forever

Android Meta-Issues

- Social issues have largely been worked out
  - Colin Cross was at Kernel Summit
  - Nobody complains like they used to
  - Linaro doing lots of "proxy" work on the features
- Android not using a continuous stream of kernels any more
  - Will use selected kernel versions longer
    - Currently plan to use 3.4 in next generation products
- Nobody really worries about "Android fork" anymore
  - Still lots of work left, though

Long Term Support Kernel for Industry

- Ueda-san will have more information later
- Small report from LTSI meeting at LinuxCon US
- Kernel version: 3.4 is the next big thing
  - Wind River supporting LTSI kernel
  - Yocto Project supporting LTSI kernel
    - Officially supported (very big news)
  - Android using 3.4 kernel
  - Next community long-term = 3.4
  - LTSI 3.4 kernel is now open for contributions

LTSI support by Yocto Project

- Plan to support multiple kernels:
  - Latest upstream kernel, 6 weeks prior to release
  - LTSI kernel
- Support qemu and 1 physical board per arch (arm, mips, ppc, x86, x86-64)
  - This means kernel and distribution testing on 10 platforms
- Projected kernels for Yocto Project releases (tentative):

YP release	1.3	1.4	1.5	1.6
Kernels	3.2, 3.4	3.4 LTSI, 3.6	3.4 LTSI, 3.8	3.8 LTSI, 3.10

LTSI release schedule

- Merge window for LTSI 3.4 open for one month
  - Just opened today (Sep 19 in US)
- Should be released by end of year
- Features:
  - Sony working on AXFS patches
  - Samsung working on F2FS patches

Outline

Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Resources

Other Stuff

- Tools
- Build Systems
- Distributions
- Android
- Industry Organizations
- Events
- Miscellaneous

Tools

- QEMU
  - QEMU is being used everywhere, for device emulation (Android, Yocto)
  - Javascript QEMU implementation (!!)
- Eclipse
  - Is now de-facto “umbrella” tool for development
  - Need to pry seasoned developers away from command line
- Tracing
  - Perf, Ftrace and LTTng 2.0
  - Common Trace Format standard

Build Systems

- Yocto project
  - Some new things:
    - "HOB" graphical interface
    - Builder image – created by Yocto Project
      - Finally can test YP with no external dependencies
  - Sony is adopting Yocto Project

Android

- Android 4.1 (Jelly Bean) released July 2012
- Ice Cream Sandwich unifies mobile, tablet and TV platforms in one codebase
- Phone activations at 1,00,000 per day
  - 400 million activations total
- Ubuntu for Android
  - Very interesting – use Android device as PC, when connected to dock (large screen and keyboard)

Events

- ELC/Android Builders Summit – Feb 2012
- LinuxCon Japan – June 2012
- Japan Jamborees
- Kernel Summit/LinuxCon US/Plumbers
  - August 2012
- Embedded Linux Conference Europe 2012
  - November 7-9, 2012 – Barcelona, Spain
- Embedded Linux Conference 2013
  - February 20-22, 2013 – San Francisco

Highlights from recent events

- Plumbers
  - Freeing memory under pressure
    - John Stultz – "Letting Go"
    - http://www.linuxplumbersconf.org/2012/wp-content/uploads/2012/08/LettingGo.pdf
  - Mini coredump (see next slide)
- Kernel summit
  - ARM mini-summit
    - Not enough embedded content
      - All about ARM64, big.LITTLE, single system image, etc

Mini core dumps

- Project to dump sparse core images
- Has a configuration-driven user agent
- Core dumps with only requested information:
  - Can save basic register, backtrace, etc.
  - Saves only part of the process image
- On host, backfills the coredump with text, read-only data, etc.
  - Once the mini-coredump is backfilled on the host you can use standard coredump analysis tools (gdb)
- Project by Thomas Gleixner

Miscellaneous

- Increased use of Stack Overflow
  - Great site for answering detailed development questions
  - See www.youtube.com/watch?v=NWHfY_lvKIQ
  - Google developers answer questions here
  - Search: “site:stackoverflow.com <question>”
- Raspberry Pi
  - Extremely low-cost development board - $25
  - Targeted at students and hobbyists

eLinux wiki

- http://elinux.org
  - Web site dedicated to information for embedded Linux developers
    - The wikipedia of embedded linux!
- Hundreds of page covering numerous topic areas: bootup time, realtime, security, power management, flash filesystem, toolchain, editors
- Working on new wiki projects:
  - Video transcription project
  - Topic-by-topic cleanup

Video Transcription Project

- Plan to ask volunteers to provide written versions of presentations from events
  - Makes it easier to search for information
  - Can make it much faster to review a presentation
  - Volunteers can do as little as one minute of video
    - Idea is to crowd-source the effort
- Not advertised yet
  - Still defining process and creating templates
  - Likely announced at ELC Europe
- See http://elinux.org/Video_transcription_project

Kernel Versions

Technology Areas

CE Workgroup Projects

Other Stuff

Resources

Resources

- LWN.net
  - http://lwn.net/
  - If you are not subscribed, please do so
- Kernel Newbies
  - http://kernelnewbies.org/Linux_3.?
- eLinux wiki - http://elinux.org/
  - Especially http://elinux.org/Events for slides
- Celinux-dev mailing list

Thanks!