Status of Embedded Linux
October 2013

Tim Bird
Architecture Group Chair
LF CE Workgroup
Drinking from a firehose
Outline

Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Best of …
Resources
Outline

Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Best of ... Resources
Kernel Versions

- Pace of versions is consistent and good
- Kernel processes are working well
Kernel Versions

- Linux v3.6 – 30 Sep 2012 – 71 days
- Linux v3.7 – 10 Dec 2012 – 71 days
- Linux v3.8 – 18 Feb 2013 – 70 days
- Linux v3.9 – 28 Apr 2013 – 69 days
- Linux v3.10 – 30 June 2013 – 63 days
  - I predicted July 7, 2013 – (7 days off)
- Linux v3.11 – 2 Sep 2013 – 64 days
- Linux v3.12-rc6
  - I predict 3.12 on …
Kernel Versions

- Linux v3.6 – 30 Sep 2012 – 71 days
- Linux v3.7 – 10 Dec 2012 – 71 days
- Linux v3.8 – 18 Feb 2013 – 70 days
- Linux v3.9 – 28 Apr 2013 – 69 days
- Linux v3.10 – 30 June 2013 – 63 days
  - I predicted July 7, 2013 – (7 days off)
- Linux v3.11 – 2 Sep 2013 – 64 days
- Linux v3.12-rc6
  - I predict 3.12 on … 8 Nov 2013 – 68 days
Outline

Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Best of ...
Resources
Bootup Time

- Kernel can be quick (under 1 second)
  - But it takes a lot of work, per product
- Lots of resources available for tuning
  - See http://elinux.org/Boot_Time
  - Good recent presentation:
    http://www.slideshare.net/righiandr/linux-bootime-23817352
- More focus recently on user-space
  - Angstrom uses systemd (yuck)
Bootup Time

- Checkpoint/Restart for Android
  - Jim Huang, 0xlab
  - Android usually takes about 30 seconds
  - Jim achieved about 15 seconds
  - See http://www.slideshare.net/jserv/implement-checkpointing-for-android-elce2012
  - Also http://www.slideshare.net/jserv/tweak-boot
- Other commercial systems are available for snapshot booting
Graphics

- Movement to higher resolutions for some embedded (e.g. Android)
- These cases demand good graphics performance
  - Movement away from frame buffer
  - Crazy rendering stuff from Google
    - LLVM renderscript
  - Buffer management a big issue
    - Need to eliminate data copies
Graphics

- Still hoping for open source drivers for embedded GPUs
- Lots of SoC GPU OSS driver projects
  - Lima, Etnaviv, Grate, Freedreno
    - See http://lwn.net/Articles/567611
- Nvidia even helping with Nouveau
  - http://lwn.net/Articles/568038
Graphics

- Shakeup in GPU market
  - ARM Mali and Vivante gaining market share

<table>
<thead>
<tr>
<th>GPU</th>
<th>1H-2012</th>
<th>1H-2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imagination</td>
<td>52%</td>
<td>37.6%</td>
</tr>
<tr>
<td>Qualcomm</td>
<td>29.3%</td>
<td>32.3%</td>
</tr>
<tr>
<td>ARM Mali</td>
<td>13.5%</td>
<td>18.4%</td>
</tr>
<tr>
<td>Nvidia</td>
<td>4.9%</td>
<td>1.4%</td>
</tr>
<tr>
<td>Vivante</td>
<td>0.3%</td>
<td>9.8%</td>
</tr>
</tbody>
</table>
File Systems

- UBIFS is taking over as de-facto standard for raw flash
  - YAFFS2 doesn’t scale to large NAND
- Rise of eMMC (block-based flash)
  - New techniques needed to address this type of hardware
  - Flash Filesystem Tuning guide
  - F2FS
Flash Filesystem tuning

- CE Workgroup project to analyze filesystem performance on eMMC
- Tested different block-based filesystems on flash media (ext4, btrfs, f2fs)
- Measured the effect of different kernel tuning options
  - IO scheduler, flash geometry vs. flash part attributes and workload characteristics
- Result document is NOW available at:
  - http://elinux.org/File_Systems#Comparison_of_flash_filesystems
- Executive summary: Correct filesystem and tuning options results depend on workload (no single winner)
F2FS

- Flash-friendly filesystem by Samsung
- Mainlined in Linux version 3.8
  - Support for security attributes in 3.12
- Log-structured, with lots of tweaks
  - E.g. hot vs. cold data separation
- I heard that Moto X uses it (successfully)
- See https://lwn.net/Articles/518988/
- See ELCE/ELC talks about it
The exFAT incident

- Weird sequence of events
- Background: exFAT filesystem is covered by Microsoft patents
  - Used for sd cards – almost a requirement to support it
- exFAT code released by independent Russian developer
  - “Liberated” from Samsung
  - Not sure about license
    - But some code may have been derived from kernel
- Samsung released code a few weeks later
- I wouldn’t use this code
Memory Management

- ION memory allocator
Ion memory allocator

- Allows sharing of memory areas between kernel subsystems (and devices)
  - Which reduces copies
- Different devices have different memory constraints (cached, contiguous, etc.)
  - ION can select memory areas matching the least-common-denominator of the constraints
  - ION can manage cache relationship to memory
- But, it uses arm-specific page accessors, and allows hardware-specific optimizations
  - It will have difficulty getting mainlined
Power Management

- Evolution of power management in Linux
  - Suspend/resume, voltage and frequency scaling, longer sleep (tick reduction), runtime device power management, race-to-sleep (wakelocks/autosleep)
  - New stuff starting to get crazy
Power Management

- Autosleep
- Power-aware scheduling
  - Big.LITTLE scheduling
- Memory power management
- Full tickless
Autosleep

- Default state of platform is sleeping, rather than awake
- Wakelock-compatible solution by Rafael Wysocki
  - Rafael: “This series tests the theory that the easiest way to sell a once rejected feature is to advertise it under a different name”
- http://lwn.net/Articles/479841/
- Mainlined in v3.5
Power-aware scheduling:

- Small-task packing
  - Try to migrate tasks to allow more CPUs to go idle
- Task placement on mixed cpu_power systems
  - Move large tasks to faster CPUs
- Resources:
  - [http://lwn.net/Articles/546664](http://lwn.net/Articles/546664) - overview
  - [http://lwn.net/Articles/552885](http://lwn.net/Articles/552885) - some resistance
  - Ingo Molnar wants to consolidate this power stuff in the scheduler – rather than spread out into power/cpufreq/cpuidle/scheduler systems
big.LITTLE

- Crazy system with small, slow, power-efficient processors, alongside big fast, power-hungry processors
- Requires some tremendous feats of scheduling to save power
  - Power-aware scheduling on steroids
big.LITTLE scheduling

- Overview: https://lwn.net/Articles/501501
- Multi-cluster power scheduling
  - https://lwn.net/Articles/539082/
- In-kernel-switcher work
  - https://lwn.net/Articles/549473/
- See talk at LCJ by Nakagawa-san of Renesas
  - One User Space Approach to big.LITTLE MP System on Real Silicon
- Still waiting for real-product results
Memory Power Management

- Is a form of device PM
  - With memory regions as the devices
- Restrict or migrate allocated memory into regions so that some banks/chips can be powered off
- Don’t have good measurements of power savings yet
- See http://lwn.net/Articles/568891
Full tickless

- Also known as “full dynamic tick”
  - Under some circumstance, some processors may run with no periodic ticks at all

Some restrictions:
- Boot CPU cannot be ‘full’ tickless
- A CPU cannot be full tickless with more than one process

See https://lwn.net/Articles/549580/
System Size

- Kernel size
- Library size
- Automated reduction research
Kernel size

- Cooperative memory relinquishment
  - Volatile Ranges
  - Lexmark work (membroker and ANR malloc)
  - See talk at ELC 2013 – "SystemWide Memory Management without Swap"
Library reduction

- **libc** – bionic libc
  - Has good features from Android, and is smaller and more configurable than glibc

<table>
<thead>
<tr>
<th>Library</th>
<th>Size Before</th>
<th>Size After</th>
</tr>
</thead>
<tbody>
<tr>
<td>glibc 2.11 : /lib/libc.so</td>
<td>1,208,224 bytes</td>
<td>→</td>
</tr>
<tr>
<td>uClibc 0.9.30 : /lib/libuClibc.so</td>
<td>424,235 bytes</td>
<td>→</td>
</tr>
<tr>
<td>bionic 2.1 : /system/lib/libc.so</td>
<td>243,948 bytes</td>
<td>→</td>
</tr>
</tbody>
</table>

- See ELC 2013 talk by Jim Huang
- **Kconfig** for eglibc
  - Ability to configure parts of libc to use

<table>
<thead>
<tr>
<th>Library</th>
<th>Original Size</th>
<th>Reduced Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>libc-2.17.so</td>
<td>1200K</td>
<td>830K</td>
</tr>
<tr>
<td>ld-2.17.so</td>
<td>128K</td>
<td>120K</td>
</tr>
<tr>
<td>libm-2.17.so</td>
<td>610K</td>
<td>580K</td>
</tr>
</tbody>
</table>

- See ELC 2013 talk by Khem Raj
Advanced Size Optimization of the Linux Kernel

- “Auto-reduce” project
- Find automated ways to reduce the kernel
  - Link-time optimization – 380K “free” reduction from compiler flag
  - System call elimination
  - Kernel command-line argument elimination
  - Kernel constraint system
- Additional research - 50% of kernel code is unexecuted
  - Link-time re-writing
  - Cold-code compression
- See Tim Bird’s presentation on advanced size optimization of the kernel
  - Notes and slides available at: http://elinux.org/System_Size_Auto-Reduction
Security

- SMACK
- SE-Linux
- Embedded integrity
SMACK

- SMACK for Tizen
  - Simplified rule set (3 tiers, 40,000 rules)
  - See http://lwn.net/Articles/55278
SE-Linux

- SE-Android
  - Implementation of SE-Linux for Android systems
- SE-Linux was previously too big for embedded
  - Early embedded SE-Linux required 2M
  - Desktop SE ruleset is 900,000 rules
- However, SE-Android only has 1658 rules and 263 types (71K policy size)
- [http://selinuxproject.org/page/SEAndroid](http://selinuxproject.org/page/SEAndroid)
  - Especially: [http://www.internetsociety.org/sites/default/files/Presentation02_4.pdf](http://www.internetsociety.org/sites/default/files/Presentation02_4.pdf)
Embedded Integrity

• David Safford’s talk at Linux Security Summit
  • Some nice simple things to do to lock down a device
  • Cheap or free mechanisms (without having to resort to TPM chip), to achieve:
    • Detect firmware modification
    • Prevent firmware modification (lock it)
    • Signed updates
    • Trusted boot

• http://lwn.net/Articles/568943
Tracing

- **Ktap**
  - Dynamic tracing, without the overhead of compiling into a module
  - Adds an interpreter to the kernel
  - Single module, that leverages ftrace, kprobes, etc.
  - Prints results in ASCII
  - Good session in LinuxCon Japan by Jovi Zhang
Device Tree
Device Tree (cont.)

• Let me cut right to the chase…
  • I don’t like device tree – there, I said it
• Supports single Zimage
• Requires drivers to separate hardware configuration from code
  • Pushes code away from platform data structures, to runtime configuration
    • Ugh – it offends my embedded sensibilities
• Is a royal pain
Device Tree

- New requirements for implementing ARM board support and drivers
- I have found it complicated to use
  - Not mature yet
    - E.g. dma, pinctrl still being developed
  - Everyone defining their own bindings
  - Not enough documentation and examples
  - No type-checking or compile-time optimization
Device tree (cont.)

- Change in maintainership
  - Grant Likely transferred maintainership to others
  - Not enough review of bindings
- Discussion about having device tree be long-lived ABI to kernel
  - Should be usable by other operating systems
  - Maybe move out of kernel repository
- Lots of discussions planned at ARM mini-summit/Kernel Summit
  - Lots of presentations at ELC Europe this year
- See http://elinux.org/Device_Tree
Things to watch

- Android features
  - Volatile ranges
  - ION memory allocator
- Device-tree churn/maturation
- Power-aware scheduling
Things to watch (longer-term)

• Non-volatile mass memory
  • Interesting remarks by Linus in LinuxCon 2012 panel
  • Won’t change a lot of kernel algorithms
  • Will mostly change filesystems
    • Byte-addressable storage has big implications for long-term storage
• Applications will still segregate data between persistent and non-persistent groups
• Things take longer to change than people think
• And, persistent RAM seems to always be 5 years out
Outline

Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Best of …
Resources
- eMMC tuning guide completed
- Open Project Proposal period

CEWG Contract Work 2013
eMMC tuning guide

• Description:
  • This project analysed EXT4, BTRFS and F2FS on a variety of block-based flash parts on a few different development boards
  • Output is a document describing best practices for tuning Linux block-based filesystems for block-based flash filesystems
  • Also, methods and scripts for filesystem testing

• Contractor: Cogent Embedded
• Status: Complete in May, 2012
  • Document at: http://elinux.org/File_Systems#Comparison_of_flash_filesystems
Open Project Proposals

• Proposal period was held recently
• See http://elinux.org/CEWG_Open_Project_Proposal_2013
• Follow link to see project list
• Was discussed at Architecture Group meeting
  • We selected 8 projects to fund, but still need to go through Steering Committee for final approval
• Selection should be finalized this week
Other Projects

• Long Term Support Initiative (LTSI)
Long Term Support Kernel for Industry

- LTSI 3.4 is available now
- Held workshop at LinuxCon Japan
  - Discussed testing phase of project
  - Discussed promotion of project
- New White Paper released:
  - See http://lwn.net/Articles/569634
- Linux 3.10 is next community Long Term Stable kernel
Outline

Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Best of ...
Resources
Other Stuff

- Tools
- Testing Frameworks
- Build Systems
- Distributions
- Wiki
- Miscellaneous
Tools

- Cortex
  - Coredump filter
  - Generates sparse core dump
  - See ELC 2013 presentation by Tristan Lelong
    - "Debugging for production systems"

- Debugging techniques
  - Good overview by Kevin Dankwardt at ELC 2013
    - "Survey of Linux Kernel Debugging Techniques"
Testing frameworks

- Autotest
  - Simple framework
  - Not cross-compiler aware?
- LAVA
  - Linaro test framework
- "Kernel Testing Tools and Techniques" BOF by Matt Porter at ELC 2013
- CE workgroup probably starting a test activity for LTSI soon
  - Need input…
Build Systems

• Yocto project
  • Lots of talks at ELCE (and previous ELCs)
  • Tutorials now online
• Buildroot
• Android

• An embarrassment of riches for build systems
Distributions

- Tizen – may be a serious competitor in embedded distros
  - Needs to open up a bit more (but it looks like it’s happening)
  - Replacing Bada at Samsung
  - Shipping in phones??
- Android use in non-CE embedded
  - Headless android
- Yocto Project = the new in-house distro
- Angstrom = packaged embedded distro
  - Very common on development boards
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 wiki projects:
    - Video transcription project
Miscellaneous

- Kernel Community Civility
- Embedded Contribution status
- Hardware
Kernel Community civility

- Recent discussion about being nicer to people on LKML
  - Sarah Sharp complained about abusive language and attitude on LKML
  - Some say harshness is needed to maintain quality
  - Others say system works OK as is
  - Is being discussed at kernel summit
Hardware

- Intel Quark processor
  - Power-efficient 486
  - Galileo board – arduino compatible
  - Signal of Intel getting into low end

- Apple M7 – separate, always on processor for location/motion services
  - Attempt to provide continuous location service without power overhead of main CPU
Embedded contribution status

- Contributions are improving, especially from embedded CPU vendors
  - See charts for embedded contribution status on LWN.net (top 3.11 contributors)
  - [http://lwn.net/Articles/563977/](http://lwn.net/Articles/563977/)
- Kernelnewbies.org/OPWfirstpatch – great document on the mechanics of a first patch contribution
- Still would be good to get a “best practices” document describing how to work with OSS
- Version gap – still with us for CE companies
  - Maybe device-tree will help us get the stable kernel API we’ve always wanted (ha ha)
Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Best of …
Resources
Best of …

- **Rules:**
  - Must be actual shipping product
  - Must do something useful
  - Not a contest – just for information

- **Categories**
  - Smallest
  - Fastest booting
  - Longest battery life
Smallest?

- TP-Link MR3020
  - WiFi hotspot
  - 4M flash chip
    - 128K U-Boot
    - 1M for kernel
    - 2.8M root filesystem
  - 32M DRAM
  - See http://lwn.net/Articles/568943
Fastest Boot

- 630 ms cold boot (beagleboard?)
  - http://www.makelinux.com/emb/fastboot/omap
- MontaVista dashboard boot in < 1 second
Kernel Versions
Technology Areas
CE Workgroup Projects
Other Stuff
Best of...
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
- LinuxCon Japan slides
  - http://events.linuxfoundation.org/events/linuxcon-japan/program/presentations
Status of Industry

- Status = Healthy
- Over 1.5 billion devices shipped with embedded Linux
  - This is a conservative estimate
- Still going strong
- We used to joke about “world domination”
  - We don’t any more
Thanks!
Extra Slides

The following slides are just for reference, for embedded-related features introduced in recent kernel versions.
Linux v3.6

- Android RAM console functionality integrated into pstore
- CANFD support for CAN protocol
  - CAN with flexible data rate
- LED oneshot mode
  - Sysfs interface for certain one-time LED/gpio manipulations
- "Suspend to Both"
  - Create resume image both in RAM and on disk
  - If power dies during suspend, disk image can be used to resume
Linux v3.7

- ARM multi-platform support
  - See http://lwn.net/Articles/496400/
- ARM 64-bit support (Aarch64)
- Cryptographically signed kernel modules
  - See https://lwn.net/Articles/470906/
- Perf trace (alternative to strace)
  - Allows intermingling kernel trace events with syscall events
- Runtime power management for audio
- Kerneldoc system can output in HTML5 format
Linux v3.8

- F2FS – flash-friendly file system
  - Details elsewhere
- New thermal governor subsystem
- Memory control group support for accounting for kernel memory usage
  - Stack and slab accounting and limits
- Cpuidle support for big.LITTLE
Linux v3.9

- Ftrace snapshots
  - Grab a snapshot of a running trace without stopping
- KVM virtualization for Cortex A15 processors
- PowerPC support for transactional memory
- CONFIG_EXPERIMENTAL=y
  - And should be gone soon
- ‘make menuconfig’ now has "save" and "load" buttons
Linux v3.9 (cont.)

- Descriptor-based GPIO
  - Access GPIOs by descriptor
    - By name in addition to by number
  - Allows for grouping GPIOs
    - For “atomic” operations
      - Possibly useful for handling realtime issues
- See [http://lwn.net/Articles/533632/](http://lwn.net/Articles/533632/)
Linux v3.10

- Full tickless (more later)
- Single zImage for ARM
  - Lots more platforms support multi-platform kernels
  - Arnd Bergmann shooting for almost-complete coverage by v3.12
- Multi-cluster power management
  - Partial support for big.LITTLE PM
Linux v3.10 (cont.)

- Multiple ftrace buffers
- Memory pressure control group support
  - Allows for notification if memory gets low
  - [http://lwn.net/Articles/531077/](http://lwn.net/Articles/531077/)
Linux v3.11

- Power-efficient workqueues
  - Allow work to be done on any CPU, to avoid waking sleeping CPUs
- LZ4 kernel image compression
- Checkpatch –fix
  - Attempt to fix some simple errors
- F2FS continues to mature
  - Lots of patches from Samsung
Linux v3.11 (cont.)

- **Zswap**
  - "Zswap is a lightweight, write-behind compressed cache for swap pages. It takes pages that are in the process of being swapped out and attempts to compress them into a dynamically allocated RAM-based memory pool. … This results in a significant I/O reduction and performance gains for systems that are swapping“

- See https://lwn.net/Articles/551401/
Linux 3.12 (probable)

- Full-system idle detection
  - Tricky rcu-based implementation to allow for fast indication of individual CPU idleness (using per-cpu variable), AND fast detection of global CPU idleness (single global variable)

- New cpu-idle driver that builds on multi-cluster power management
  - I.e. Getting closer to support for “big.LITTLE” CPU scheduling

- Lots of device drivers converting over to device tree
  - More on this later