



CE Linux Forum

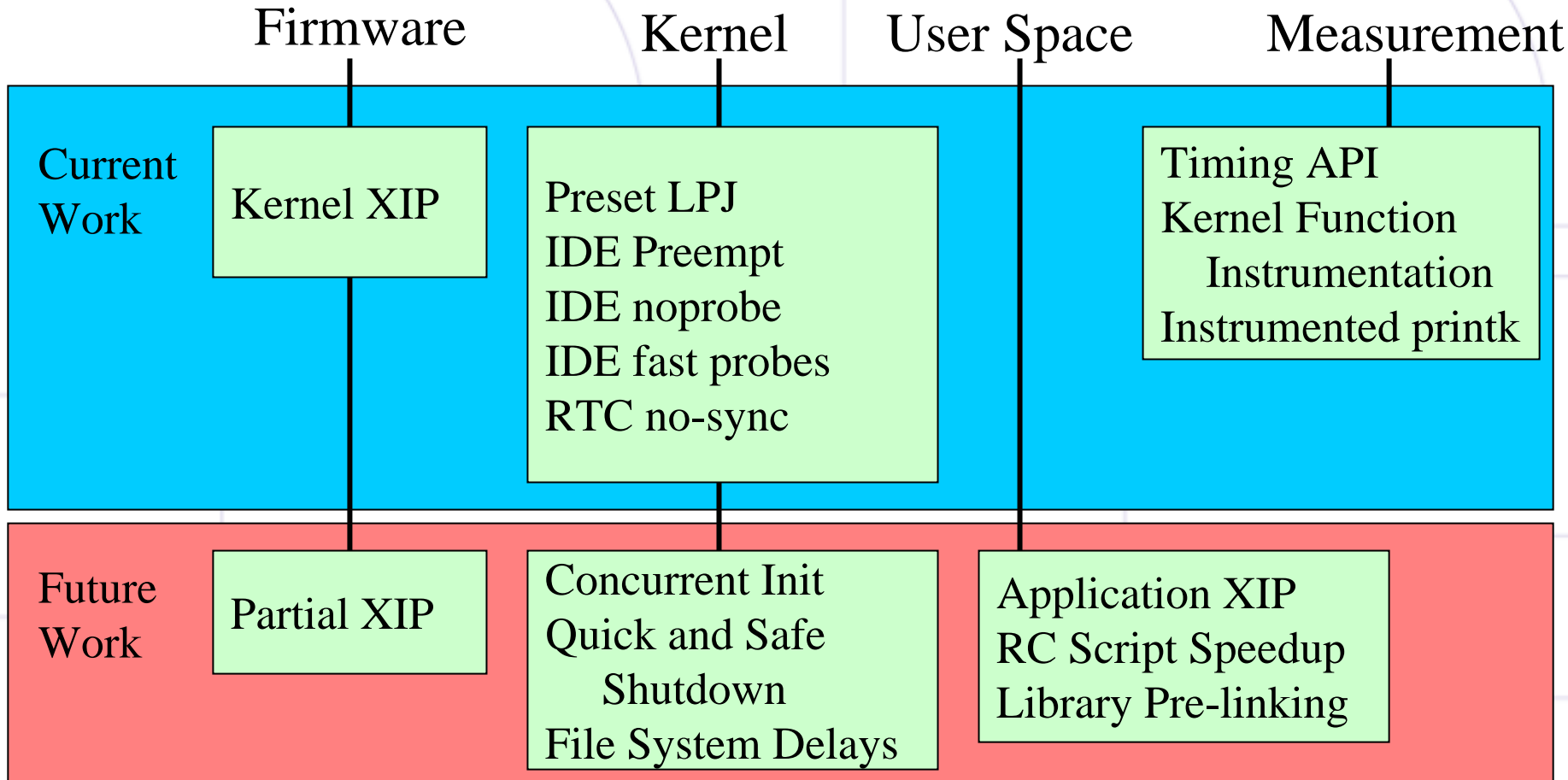
Bootup Time Working Group

Discussion

Tim Bird, Chair



Technology Overview





Patches for Bootup Time

Patch	Status
Kernel XIP (CRAMFS-linear)	In patch archive (2.6.9)
Preset-LPJ	In mainline (since 2.6.8)
Short IDE delays	In patch archive (2.6.8-rc2)
RTC nosync	In patch archive (x86: 2.6.7, ppc:2.6.10)
Printk-times	In patch archive (2.6.10)
Kernel Function Instrumentation	In patch archive (2.6.9)
Linux Trace Toolkit	In patch archive (2.6.10 and 2.6.11-mm)



Current Status

- Some patches are being maintained
 - Need assistance extending/maintaining patches for other architectures
- WG activity has fallen off, due to:
 - Move to 2.6
 - Laziness by WG Chair
- Need to restart WG activity



Goals

- Verify instrumentation tools (printk-times, KFI, LTT) on other architectures
- Submit existing patches to LKML
- Focus on user space:
 - RC scripts
 - pre-linking



Current Implementations



Short IDE delays

- IDE init code delays over 70 times during startup
- Patch changes IDE delay from 50 milliseconds to 5 milliseconds.
- Patch was submitted to LKML in August, but rejected
- Need to determine reason for so many calls to delay during IDE init
- Some calls to delay can be avoided using IDE “noprobe” on the kernel command line
- Need to evaluate risks of short IDE delays



IDE-noprobe

- IDE “noprobe” option on kernel command line allows skipping some IDE probing during init.
- Option is sometimes disabled (overridden) by PCI init code
- This was true in 2.4, we need to confirm this behaviour in 2.6
- Need to find out why noprobe is being overridden
- Current patch is a quick hack to identify problem.
- Need to ask LKML about behaviour



RTC No-synch

- Avoid synchronization with RTC clock edge during first load of system time
- Have patches for x86 and PPC
- X86 patch was submitted – apparently ignored
- Need to submit again



Kernel XIP

- Execute kernel directly from flash
- Don't know current status in kernel



Application XIP

- Execute applications and libraries directly from flash
- CRAMFS-linear patch – what is status?
- **RC script speedups –Library prelinking**
–



Timing API

- Is a Cross-platform API for timing measurement code
- Implementations were completed on x86 and PPC
- Code was never submitted to LKML
- Code needs to be tested and compared with `sched_clock()`



Printk-times

(a.k.a Instrumented Printk)

- Shows timing information on each printk line
- Works on any platform supporting `sched_clock()`
- Need to submit to LKML
- Should add printks at key init checkpoints
 - Checkpionts were identified in April



KFI

- Measures time and duration of each kernel function
- Works on any platform supporting `sched_clock()`
- Is currently being extended:
 - untimed traces
 - more complex triggers
 - `proc fs` output (eliminating need for user-space tool)
- Need to submit to LKML
- Also want additional post-processing tool to show different trace summary



Research

- Library Pre-Linking
 - Link applications to shared libraries ahead of runtime
 - Need prototype system to measure performance
- RC script speedups
 - Reduce fork cost and improve use of busybox builtin programs
 - busybox patch needs to be submitted to busybox mailing list for comment



Other Resesearch

- Concurrent Init
 - Need to find drivers with spare (busywait) cycles
 - Need to identify prototype mechanism for overlapping driver initializations
- Filesystem measurements
 - Completed initial tests
 - Need to find spots that might be improved.



Project	Status						
	Research	Spec	CELF Impl.	2.6 Impl.	Evaluated	Doc	Accepted
Calibrate delay/Preset-LPJ	done	done	done	done	done	3	done
IDE-noprobe	3	done	3	0	3	3	no
Short IDE delays	done	0	0	3	0	3	no
Kernel XIP	done	done	4 @	0	0	2	no
Application XIP	3	0	4	4	0	0	no
Timing API	done	3 - WIP	0	3 @	0	0	no
KFI	done	0	done	4	done	2	no
Printk-time	done	3	3 @	done	4	0	no
RTC no-synch	done	0	done	done	3	0	no
RC script speedups	3	0	n/a	n/a	0	0	no**
Concurrent Init	2	0	2	0	0	0	no
Quick and Safe Shutdown	3	2	0	0	0	0	no
File system delays	2	0	0	0	0	0	no
Pre-linking/lazy linking	0	0	n/a	n/a	0	0	no*



Core Group

- Noboru Wakabayashi – Hitachi
- Narm Gadiraju – Intel
- Kiyoyasu Maruyama – Mistubishi
- Todd Poynor – Monta Vista
- Yoshinori Nagai – Sharp
- Tim Bird – Sony
- Keisuke Yasui - Toshiba