- **Regan Robertson** Mod • 14 days ago • edited
  Good Morning,
  The video will start on this page at 11:00am, and you may have to hit play or unmute. As a reminder this site works best in a Chrome Browser. You can write in comments through this feature to continue the conversation or ask questions. Please login or sign up for a Disqus account to participate in the discussion boards.
  - Reply
  - Share ›

James Potts • 14 days ago
Hi, all. Welcome to our presentation. One thing I'd like to point out, we're purposefully vague on the hardware requirements we were operating under.
- Reply
- Share ›

Darren Cofer James Potts • 14 days ago
Wait, who is this now? Future James or Past James?
- Reply
- Share ›

James Potts Darren Cofer • 14 days ago

Ironically, with the audio issues, we could have just uploaded the original, and pretended Mark's audio was just cutting out. :)

Reply

Share ›

Stuart Card James Potts • 14 days ago

Your test pattern & voice over reminded me of https://en.wikipedia.org/wi...

Reply

Share ›

Jason H Li • 14 days ago • edited

Question to presenters and maybe CoE as a whole - what would you think a viable + sustainable solution to support BSP for the community? I mean more boards than what you mentioned and beyond.

Reply

Share ›

Olin Sibert Jason H Li • 14 days ago

My team wrestled with all these issues recently on Blackjack, and while there is a lot of knowledge within the community about what's practical and what's not, it's quite fragmented and mostly buried behind NDAs. I'd like to see the CoE pick a handful of target hardware platforms and produce the kind of robust BSPs (and documentation, and porting guides) that were discussed here--as open source.

2

Reply

Share ›

Noah Evans Jason H Li • 14 days ago

To add to what **Olin Sibert** said. From Sandia's perspective we'd really like to settle on (or maybe even design ourselves) as a community boards that are representative of our various classes of needs. Sandia tends to have very resource constrained systems for mission purposes and other more capable systems for other needs.
One way to go might be to design a "small" board with limited resources, a "medium" board with traditional specs and a "large" board with full VM support and a more capable system.

You'd probably want all of them to have a CAN/Spacewire/etc packetized bus and something like AXI local to the processor itself.

The Draper guys might also have some input here.

- Reply
- Share ›

Ihor Kuz Noah Evans • 14 days ago

And don't forget that they should all be supported by the proofs.

- Reply
- Share ›

Noah Evans Ihor Kuz • 14 days ago

Very good point. The follow up to that is that the BSPs should have open source board designs and HDL for verification. e.g. verifiable with Yosys or in languages designed for verification like Bluespec/Kami but that's more of a stretch.

- Reply
- Share ›

Olin Sibert Noah Evans • 14 days ago

I'd say more than three, less than ten. Driving factors are multiple CPU architectures and readily accessible (i.e., cheap) development hardware. Removing barriers to adoption (cost, complexity) is really important--that's why Raspberry Pi is a great target, even though it's unlikely to end up in an avionics rack.

Another key issue is that a lot of the solutions that are produced privately are targeted to a particular application, and aren't quite general-purpose. I think having a CoE-funded effort to produce a coherent set of BSPs would be much more effective than simply open-sourcing a bunch of privately-developed but not-quite-complete versions.

Reply

Share ›

James Potts Olin Sibert • 14 days ago

On the bright side, most families of SoCs have fairly consistent peripherals. For example, if there was full support for the peripherals on the Zynq Ultrascale, and full support for Freescale's arm peripherals, it would probably go a long way .

Reply

Share ›

Ihor Kuz Olin Sibert • 14 days ago

I think "funded" is the key word here :-)

Reply

Share ›

James Potts Jason H Li • 14 days ago

As I responded "above" to **Ihor Kuz**, targeting broader support of a select number of SoCs would go a long way. If we could know that any board with a Xilinx MPSoC on it would have all its standard peripherals supported, or any board with a Kintex Whatever™ would mostly work, it would be a big start.

Reply

Share ›

James Potts James Potts • 14 days ago

Unfortunately, that means less priority for some popular boards. For example, the Raspberry Pi is a fantastic little board. But you won't find its SoC on any other solution. So while it's convenient to target them for development, it does nothing as far as increasing support for systems used in industry. Not only can we not buy an industrial/hardened board containing its processor, but Broadcom likely wouldn't even sell them to us for custom boards at the volume that we produce.

Reply

Share ›

Aleksey Nogin • 14 days ago

My "RFE" for this from a few years back was to create a script that would run on target HW under Linux, look over the HW, and spit out some sort of report on what peripherals are supported by seL4 and to what level. Ideally, maybe even spit out some stub camkes configuration for the target HW.

1

Reply

Share ›

Ihor Kuz Aleksey Nogin • 14 days ago

We currently have better support for using device trees (in the kernel and in CAmkES) so that probably handles some of this. But it's definitely not complete yet.

Reply

Share ›

James Potts Ihor Kuz • 14 days ago

And in the case of the Tegra K1, it wouldn't have helped, because the issues we ran into were hard-coded into the kernel code, and not even called out in the K1 device-tree. (Not much you can do when kernel devs ignore proper standards.)

Reply

Share ›

Ihor Kuz • 14 days ago

The problem of platform support and the need to support a wide variety of existing platform is difficult. Currently I see two options:
- limit the range of supported hardware (to 1 or 2 boards), and fully support it
- provide limited support for a wider range of hardware, and make it a 'community problem' to provide more complete support.

Neither are ideal. :-(

But which would people prefer?

1

Reply

Share ›

Ihor Kuz Ihor Kuz • 14 days ago

Note that better documentation of what *is* supported is definitely necessary. Something we are slowly working on improving.

2

Reply

Share ›

Aleksey Nogin Ihor Kuz • 14 days ago

Both ;). But I think the primary issue is really to be clear what is supported and to what degree.

2

Reply

Share ›

Nathaniel Husted Ihor Kuz • 14 days ago

In a way couldn't there be a bit of a combo? The foundation focuses on a set of "pristine" platforms while the others are left to the community. This ties back to **Jason H Li**'s earlier comment about what's viable/sustainable for BSP: You really want a list of "features" of what it means to be supported -- and keep those up to date.

Reply

Share ›

Jason H Li Ihor Kuz • 14 days ago

For the DoD community, there may be a need to support more than 1 or 2 boards. So IMHO it is a CoE problem, to support and fully support.

Reply

Share ›

Nathaniel Husted Jason H Li • 14 days ago

There's probably a commonality challenge as well given engineering activities may want to start on dev boards for prototyping than move out into a more bespoke platform later that would arguably need to be kept "Distribution D"/ITAR.

Reply

Share ›

Jerry Dussault Ihor Kuz • 14 days ago

Great discussion here - clearly a topic worthy of follow-up, and maybe building a strategy among the community (Foundation, CoE, +...).

1

Reply

Share ›

Ihor Kuz Ihor Kuz • 14 days ago

My experience is that whenever we focus on better supporting one platform, someone always comes along and says "but I need to use this other platform". And as the presenters said, this is a legitimate requirement based on the application requirements and constraints.

- Reply
- Share ›

Robbie VanVossen Ihor Kuz • 14 days ago

A better defined driver model would help this as well. If you support one platform really well then it can be used as a reference, along with the driver model, by the community to more easily add new drivers for new platforms.

- Reply
- Share ›

Ihor Kuz Robbie VanVossen • 14 days ago

I have a talk about that tomorrow...

- 1
- Reply
- Share ›

Ihor Kuz Ihor Kuz • 14 days ago

Anyhow, defining a more limited, but better supported, base of hardware and software for seL4 is something that the Foundation aims for and that Gernot will talk about on the last day. https://www.sel4summit.com/...

Reply

Share ›

James Potts Ihor Kuz • 14 days ago • edited

Ignoring x86 for the moment, choosing a small number of SoC families to support would probably go a long way. Say, for example, Xilinx, Freescale, and [insert a third cheaper option here: Allwinner? Broadcom?].

Not all boards will be the same, and not all SoCs within a family will be the same, but most Freescale parts are going to use the same peripherals on the die, regardless of the processor core or address where they're located.

At that point, supporting "boards" becomes much less of an issue. Any board with an MPSoC? Almost turnkey. A new Kintex board? Change some addresses. (Yes, I'm obviously simplifying, but you get the idea.)

In the future, we might see more SoCs using PCIe as a peripheral interface, making for a more generic driver model, but until that time, focusing on a few company's families would go a long way.

Reply

Share ›

Raymond Richards • 14 days ago

How are you managing IP concerns? Do you feel that drivers for commodity peripherals should be released openly? What about the COE restricted repository?

Reply

Share ›

Jason H Li Raymond Richards • 14 days ago

The DoD BSP, before pub release, should reside in the COE restricted repo IMHO.

Reply

Share ›

James Potts Raymond Richards • 14 days ago

That's above my pay scale. ;)
But as a rule, I'd love to see all manufacturers publish baseline drivers for their peripherals. The "protect our secret sauce" reaction that many take is certainly detrimental.

Reply

Share ›

Renato Levy • 14 days ago • edited

what if, instead of using a full Linux, one would have adopted a shim-like OS, like the ones used in cloud systems? They don't assume they have unfettered hardware access.

Reply

Share ›

James Potts Renato Levy • 14 days ago

That might have worked, but for the physical layout of the memory map. Without paravirtualization, we still wouldn't have been able to map devices into the shim without giving the shim pretty unfettered access to the system itself.

Reply

Share ›

Renato Levy James Potts • 14 days ago

I guess this would be an issue for any Cyber-physical system. I am asking because one of my pet projects is to modify a JVM to execute directly on seL4, or even C#. This way one can load a program to execute, with minimum overhead from the unused OS

- Reply
- Share ›

James Potts Renato Levy • 14 days ago

Oooh. C#. :) In a past life excursion, I did a lot of enterprisey development. I might not love Windows, but I learned to really appreciate C#.

- Reply
- Share ›

Renato Levy James Potts • 14 days ago

Me too. Very few know that C# also executes in a virtual machine, which runs on CIL. Easy to prove things too on stack based machines....

- Reply
- Share ›

James Potts James Potts • 14 days ago

Actually, Java and C# are great examples. Both provide an immense amount of support beyond the base language, for all sorts of development needs (as does Python), which led to their rapid adoption.

Linux did the same, in a way. I'd argue it was the first "alternate" OS that truly supported a wide range of peripherals without end-user pain (I'll leave the NetBSD/FreeBSD arguments to the side).

The embedded space, of course, is much harder, due to the lack of standardized interfaces for peripherals.

- Reply
- Share ›

Ihor Kuz Renato Levy • 14 days ago

Problem is that any non-trivial application needs a bunch of device drivers (and other OS functionality - FS, network stack, etc.). That's what the VMs provide.

- Reply
- Share ›

Renato Levy Ihor Kuz • 14 days ago

I am not saying you don't need a VMM, just not the full overhead of a Linux on top of seL4, since the most critical part of management is already done by seL4, and better.

Reply

Share ›

Ihor Kuz Renato Levy • 14 days ago

Agreed. The reason for using Linux is that nothing else generally provides the required support 'out of the box' without significant work.

Reply

Share ›

Robbie VanVossen James Potts • 14 days ago

This is really an issue for all bus type device. PCIe, USB, SPI, I2C, etc. None of these devices currently have any form of hardware virtualization so to divide up devices on the bus, there must be paravirtualization or device emulation.

Reply

Share ›

Stuart Card • 14 days ago

Have you looked at the PolarFire RISC-V / FPGA SoC from Microchip/Microsemi? Their ICICLE developers' board? If so, what is your assessment of its level of support as provided by the DornerWorks port of seL4?

Reply

Share ›

Regan Robertson Mod • 14 days ago

Please join us for the next session that starts in 2 minutes. You can either go back to agenda to get to the next session or at the bottom of the page there is a next session button.

Reply

Share ›