Bootlin training course evaluation

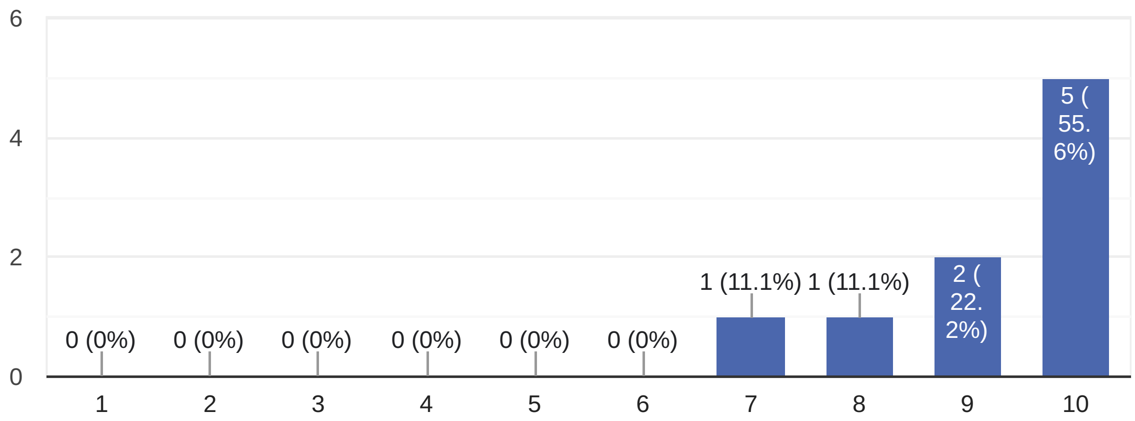
9 responses

Publish analytics

Overall rating of the course

Copy

9 responses



Comments and suggestions

5 responses

Overall, very professional and competent course. Lectures were interesting and labs were thorough, with in-depth knowledge from Alexis.

It was an excellent course facilitated by a competent and friendly trainer.

Days 1 and 2 were okay -- lots of material that I was exposed to in the past and was able to follow the labs via userspace / on my Linux host. (rating 3/5)

Day 3 is when new material started getting exposed to me and the embedded board was necessary for the labs (rating 4/5)

Days 4 - 5 is a chunk of new material accompanied with involved demonstrations and lots of screenshots on my side

These labs are super useful, and I really think having 1 or 1.5 extra days just to accomplish the labs with a partner and instructor providing guidance would be of high value. I understand this is possible in the in-person sessions; I was curious if there could be a "premium" version of the course where there is that extra day provided just to focus on lab work?

I think one improvement would be to involve the students more, as in asking open-ended questions and inquiring what commands to run, what are we looking at, how would we use this, etc. This helps engage the audience more and makes the content stick longer.

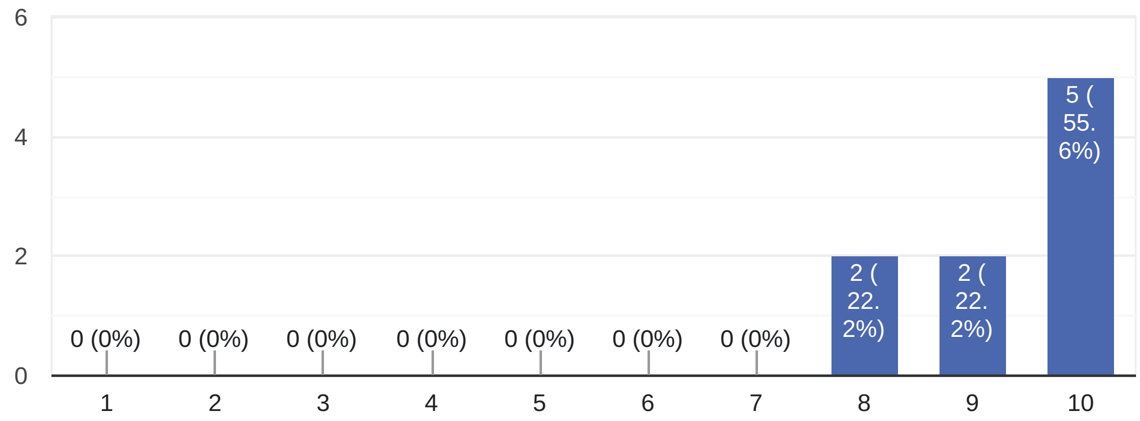
Great course!

Very good overview on tools for Kernel debugging, also Lab sessions were quite interesting

How useful were the lectures?

Copy

9 responses



Comments and suggestions

2 responses

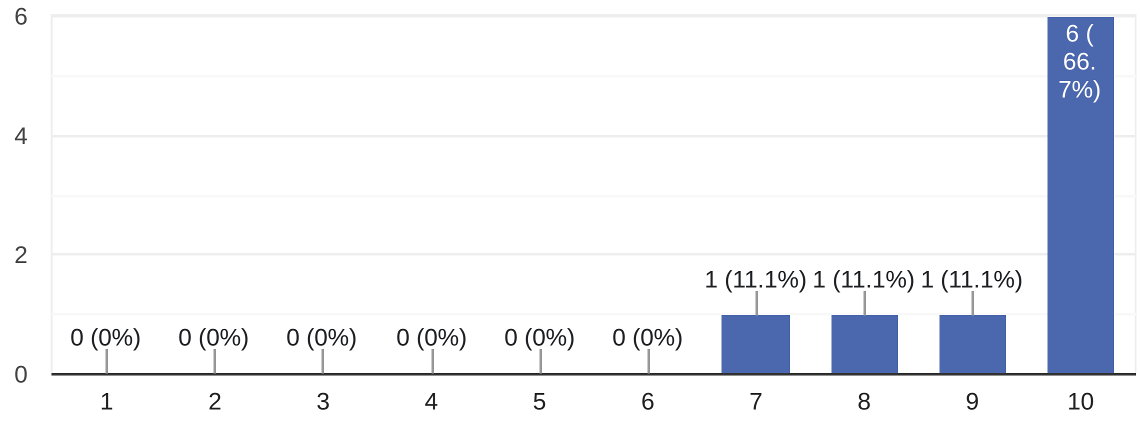
The material and labs were well thought out and executed. Alexis had good knowledge on the topics discussed and actively responded to questions and feedback during the sessions.

Awesome to have to an overview compendium of tooling.

How useful were the practical demos?

Copy

9 responses



Comments and suggestions

3 responses

Considering the time spent on them, they were very useful. Clear instructions for compiling and setting up the hardware. Labs actually demonstrated all the discussed tools and methods.

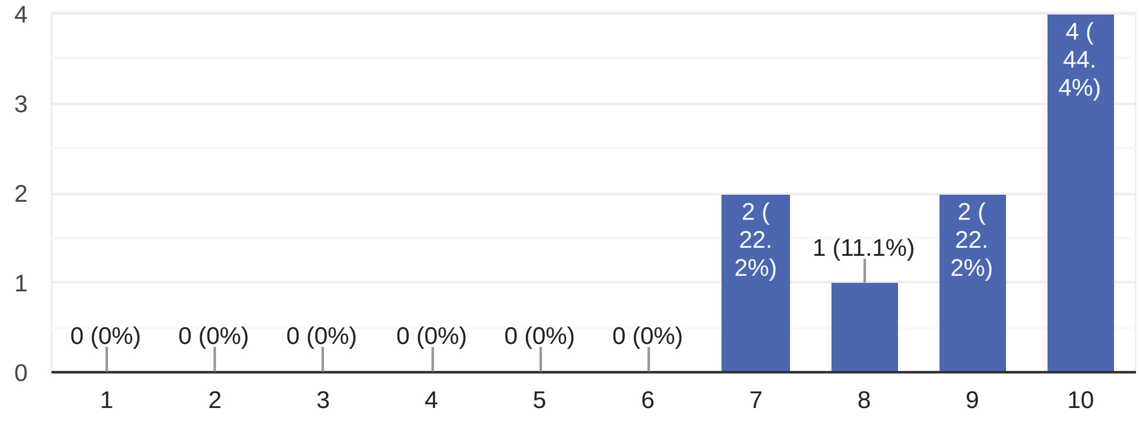
Re-iterating the above that it would be great to have an extra day dedicated to labwork.

Very useful as startpoint to these useful tools.

How would you rate the overall organization of the course?

Copy

9 responses



Comments and suggestions

3 responses

No problems here, course organization was excellent.

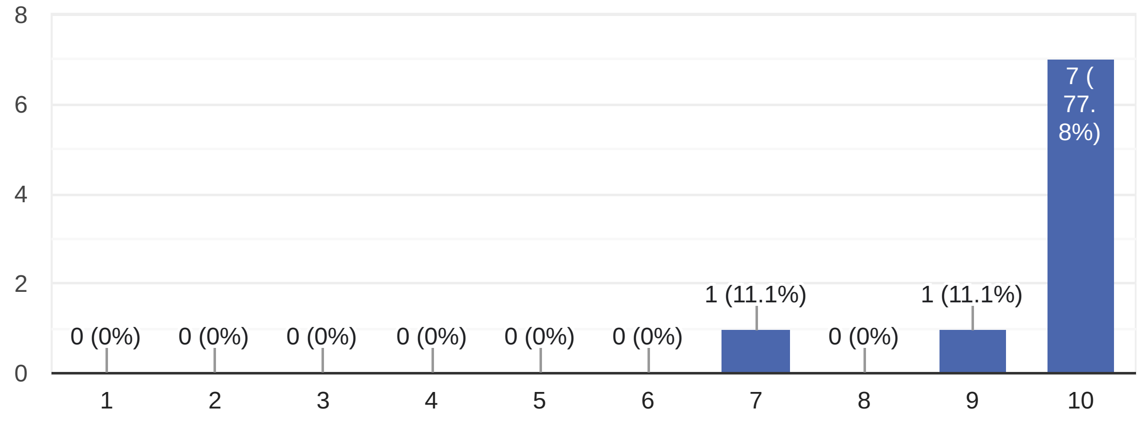
I would slightly change days 1 and 2 a little such that the user can apply / follow the labs on their host machines if available

I feel that maybe splitting the course in two parts (like, 2 days on the first week and 2-3 days on the second week) would be better for two reasons: (1) easier to interleave with the work commitments; (2) it gives some time to go over the course material, experiment and to come up with some deeper questions.

How would you rate the trainer?

Copy

9 responses



Comments and suggestions

4 responses

I had some issues from poor internet connection on some of the lectures, but I cant pinpoint at where the failures originated (likely my end).

Answered questions thoroughly in lecture and after lectures / offline. Provided resources, and attempted different things in demonstrations when asked about a possibility

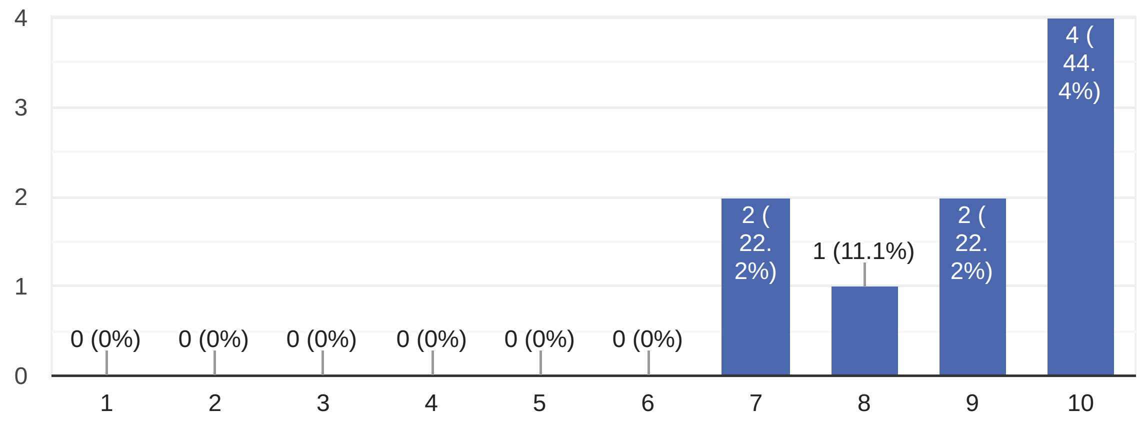
I think Alexis is a great trainer. I enjoyed his presentation style.

Alexis is a highly knowledgeable and committed trainer, delivering the material with clarity and in a well-organized manner.

How did the course meet your learning objectives?

Copy

9 responses



Comments and suggestions

4 responses

Personally, I was hoping for a bit more tracing/profiling methodology (wasn't advertised, so it's totally fine)

Was looking for an overview of tools and a demonstration / feasibility of how debugging in kernel space would apply to my workflow, now need to implement and try it myself :)

The course was a great overview of available technology!

I would have enjoyed going into slightly more complex performance issues or another example like the one with the cache line. Possibly instead of exploring LTTng in more detail.

Another thing I would be interested in, are real-life stories. That is summaries of stories (more complex than the examples in the lab) that were solved by Bootlin staff or whoever using the presented tools. Possibly including any a-ha moments or lessons learned. However, I also acknowledge that the time was already well filled and the pace shouldn't be any faster and I would not know what else to drop from the content.

I personally would prefer to have more lab time on eBPF/more complicated perf/ftrace usage scenarios

What part(s) of the course did you like most?

6 responses

I really appreciated the labs and demonstrations were clear and practical.

The tracing, profiling, and kernel debugging portions were great as I haven't been exposed to that in the past.

The labs.

User-space debugging tools, but maybe in the future I will need kernel-space tools.

Userspace debugging possibilities

The kernel part

What part(s) of the course did you like least?

6 responses

The length of daily sessions and the pause to content time ratio. I wish the ratio was closer to 45 min of lecture / 15 min of pause.

I liked them all.

Wish we could do the labs while the instructor is there guiding us!

Nothing.

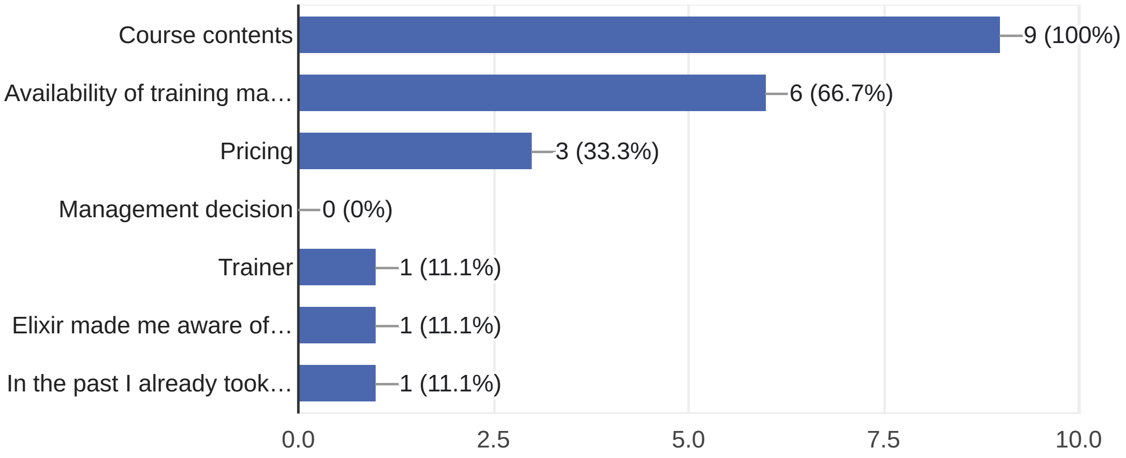
none

The beginning of the course

What reasons prompted you to choose a Bootlin course?

Copy

9 responses



Comments

4 responses

Thanks for Bootlin and Alexis for arranging the course!

Thank you so much for the training, it was great and I would recommend it to my colleagues!

Great course. Thank you!

Also more content which is also relevant for x86 systems would also be helpful. Matrix server is not accessible, so it would be good, if there are any alternatives. Thanks.

Further training needs?

6 responses

All your other courses are relevant to me, and I will work my way through most of them.

tracing/profiling methodology (regarding real-time)

Already signed up for a bunch of courses, hoping to run through the entire curriculum within the next 2-3 years! Excited to see y'all in the next one!

Would love a code-oriented eBPF course for writing and exploring systems via eBPF programs, manipulating data before other parts of the kernel stack receive it, divert code paths away from buggy code (or hot-patching buggy code), and debugging eBPF program issues

It would be nice if there was a course about programming C for embedded. Using these courses tools for improve and to learn how to refactor programs to improve CPU load, memory usage ....

Realtime Linux

Like you have a course on the graphics stack, it'd be very useful to have a course covering Linux network stack;

Also, an advanced course for "experienced" kernel developers - suppose somebody is fluent in the material covered by the LDD book, then a short course talking about the intricacies of RCU/memory barriers/etc etc would be very beneficial.