

Bootlin training course evaluation

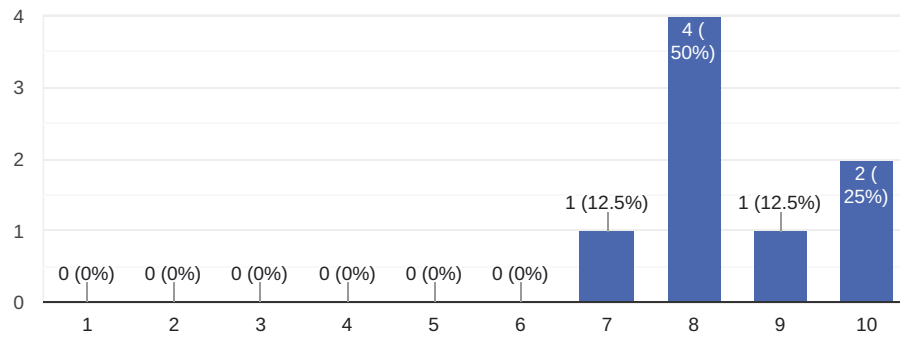
8 responses

[Publish analytics](#)

Overall rating of the course

 Copy

8 responses



Comments and suggestions

4 responses

Clear course, well prepared slides and labs

Deep and clear content

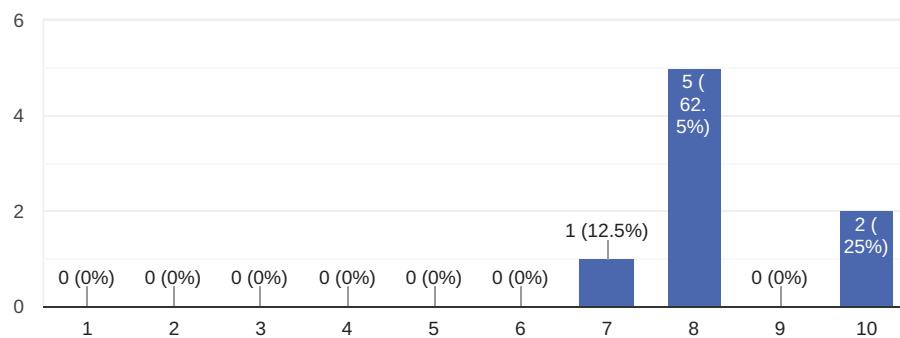
Great split between lectures and demos. Nice that the demos were done soon after learning about the tools in the slides.

To provide a video guide of the practical section

How useful were the lectures?

 Copy

8 responses



Comments and suggestions

4 responses

Good lectures, great delivery

It would be beneficial to add a table summarising the tools and their uses for quick & easy referencing in the future when we are faced with an issue to debug.

Introduction/Linux Application Stack quite long. I understand this make the course more accessible, but these notions were already acquired on my side.

I would have like a summary/index/cheatsheet or something to have an overview (more graphical maybe ?) of what was done during the course as a lot of things were covered. Personally I have done an index like this:

Theoretical

[1](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=21>) Explanation on:

- Process
- memory system
- scheduling
- interrupts
- ELF

[2](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=58>)Pseudo filesystems (proc/sys/debugfs)

Tools

- ELF - LDD and binutils [3](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=63>)
- System monitoring [4](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=67>)
- ps / top
- free / vmstat
- mpstat / pmap
- iostat iotop
- Debugging [5](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=78>)
- GDB / Remote GDB / Python extension
- Tracing (strace / ptrace / ltrace / LD_PRELOAD / perf) [6](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=99>)
- Memory issues (valgring / vgdb / libfence) [7](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=113>)
- Profiling
- Memory usage [8](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=132>) :
- only heap : massif / heaptrack
- Memusage
- Execution [9](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=141>):
- perf [stat|list|report] (performance counter)
- cachegrind (cache problems)
- callgrind
- System-wide / Kernel [10](<https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=150>):
- kprobes / kretprobe
- perf [list|probe|report|script|trace]
- ftrace (kernelshark GUI / trace-cmd CLI)
- irqsoff
- hwlat



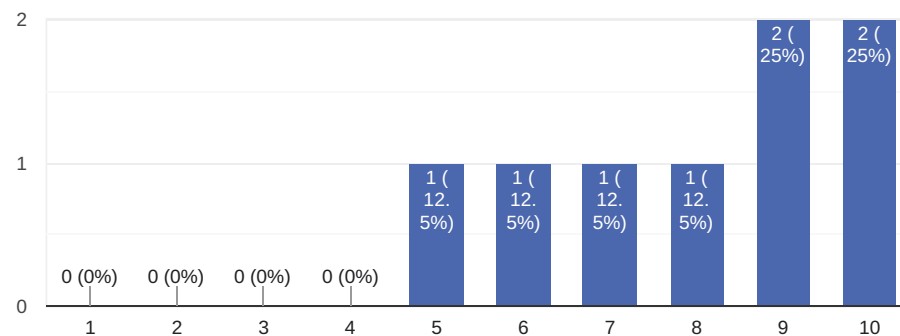
- eBPF / BCC / bpftrace
- LTTng
- Kernel Debugging [11](https://bootlin.com/doc/training/sessions/online.debugging.sep2023/debugging-slides.pdf#page=205)
- `pr_*()` / `trace_printk` / `dev_*()` / `*_ratelimited()` / `[printk_formats]`(https://www.kernel.org/doc/html/latest/core-api/printk-formats.html)
- Understand kernel oops: `addr2line` / `decode_stacktrace.sh`
- Kexec / kdump
- kmemleak: scan whole memory for dangling memory allocation
- KASAN: use after free / out of bounds
- UBSAN: runtime checker for UB
- KGDB / KDB
- [crash](https://github.com/crash-utility/crash) (taking coredump as inputs)
- locking/concurrency debug strategy
- sparse - static analysis

To provide more links for external resources

How useful were the practical demos?



8 responses



Comments and suggestions

5 responses

Maybe the demos would be more efficient if it required the participant to actively do it instead of just looking at someone doing it. For example in the final test include a question which answer would be a result of a lab.

Difficult to rate this because i didn't had the dev kit. Frustrating

Sometimes it was a bit fast, but really useful.

Practical demos were very good, but the real value for me was doing it at the same time than the trainer. This made more interactive/interesting, but the course is not really planned around that and I understand this bring a whole can of worse logistically and in term of support.

Maybe some remote access to demos system could maybe help ?

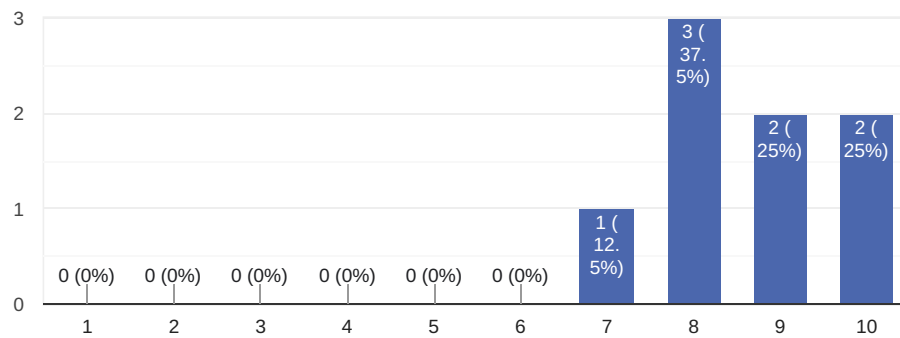
To provide a video guide of the practical section



How would you rate the overall organization of the course?



8 responses



Comments and suggestions

2 responses

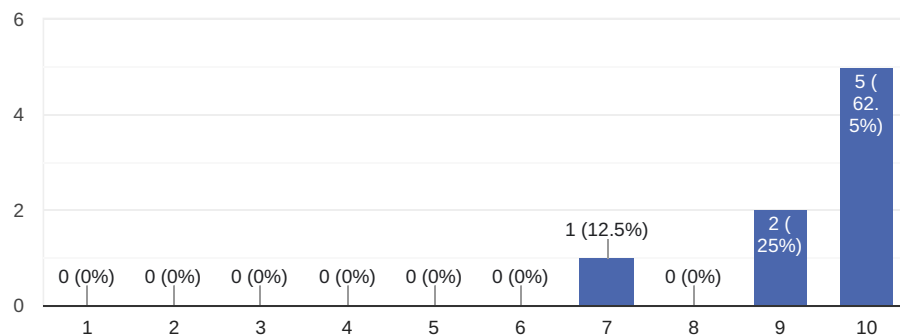
Everything about the process, training materials with all the links were sent before the training, in time so there was no misunderstanding. The organization was excellent.

The lectures were good, however I would tweak the ratio a bit. Trading less lectures for more practical demos.

How would you rate the trainer?



8 responses



Comments and suggestions

5 responses

Nice trainer, answers all questions. Sometimes didn't have the answer but came back the next session with the answer (or on the chat)

English easily understandable (for a French). Clear.

The trainer was very friendly and approachable, able to answer all applicable questions and able to explain concepts to a group with varying levels of experience.

Helpful trainer with a great understanding of the training material and good communication skills.

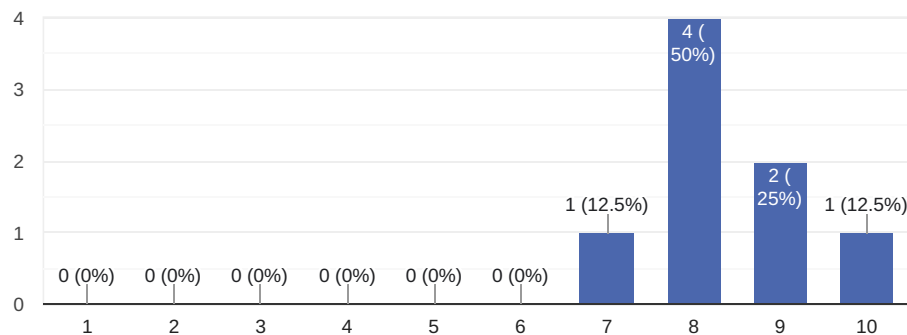
He was explaining well, knowledgeable and was still good after 4 hours talking with a remote group, that's clearly not easy.



How did the course meet your learning objectives?



8 responses



Comments and suggestions

2 responses

It definitely met my learning objectives, it was my first introduction to nearly all the tools and I feel that it has set me up with a good understanding of what tools to use in various situations.

I would have like more kernel focused lessons, but clearly the user space tools are going to be usefull too.

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

5 responses

This course is like a list of several tools that one can use to debug different parts of an embedded system

ftrace

The application debugging and kernel debugging section along with their demos.

Parts related to performance analysis and tracing, profiling.

KGDB

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

5 responses

The labs were a bit too fast and one could get lost easily since there is no hands on from the participant

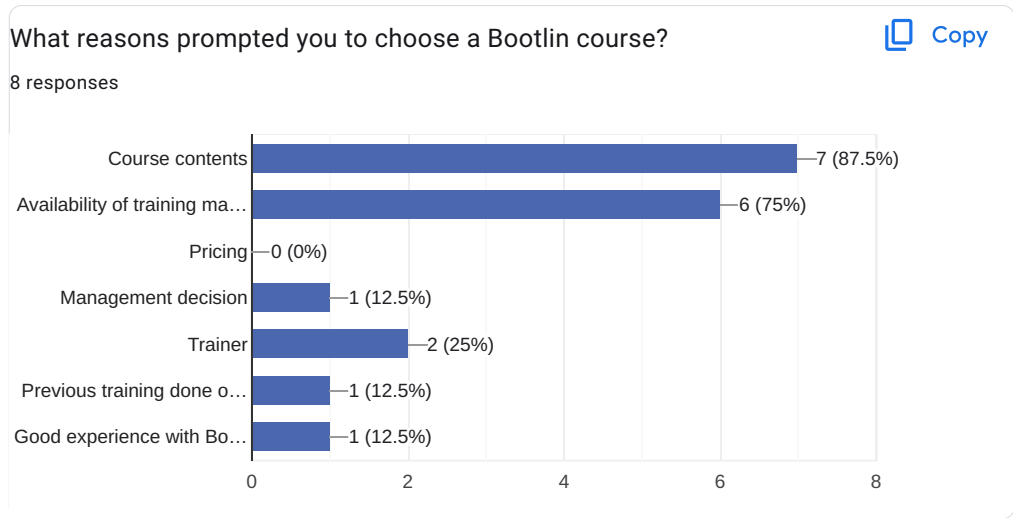
eBPF

The application profiling section

Linux Application stack/Introduction

last day of practical section





Comments

1 response

Thank you, I learnt a lot from this course and hope to do another one of your courses in future!

Further training needs?

1 response

I'm interested in boot_time optimization that you already offer.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

