Bootlin training course evaluation
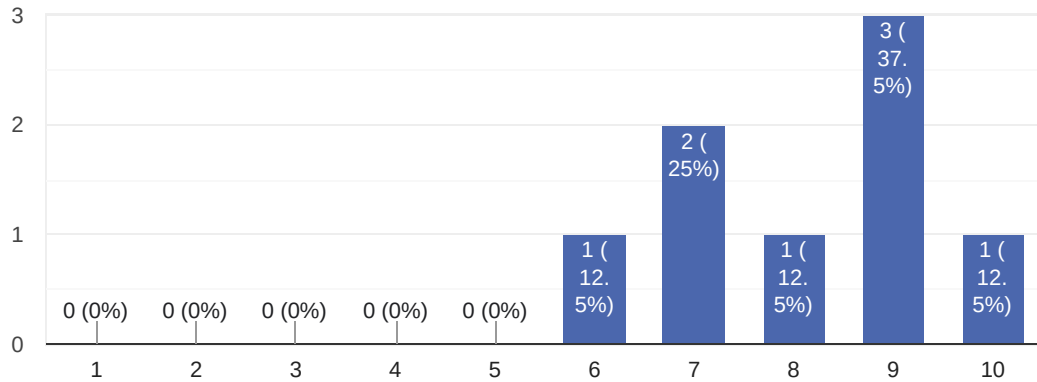
8 responses

Overall rating of the course

Copy

8 responses

Comments and suggestions

4 responses

A lot of materials covering more topics than needed - basics theory IMHO could be shorter giving more time for practical demos / exercises

A lot of talk on theory around graphics and not enough on Linux Graphics (IMO)
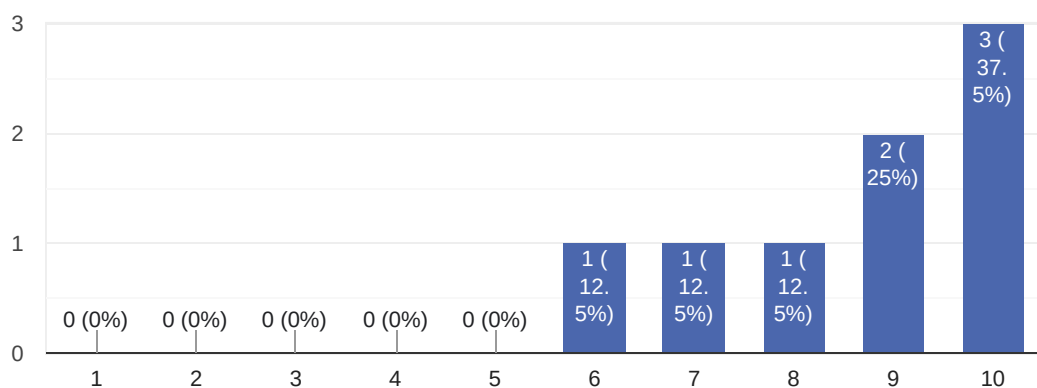
better insight with kernel subsystem maybe have more practical tasks

Perhaps for the online course 5 days could great instead of 4 days of duration.

How useful were the lectures?

Copy

8 responses

Comments and suggestions

3 responses

Some more drawings, diagrams would be appreciated on the slides for better understanding instead of only bullets

it would be helpful to see other frameworks at low level(dmabuff, v4l2, etc) where we can see how data flow within a graphic pipeline, there wasn't too many practical examples. There are some cheap HW that users can get in order to practice and actually have hands on hardware and monitor tools

It is good that an instructor guides through the lectures in order to iunderstand them.

How useful were the practical demos?

[□] Copy

8 responses
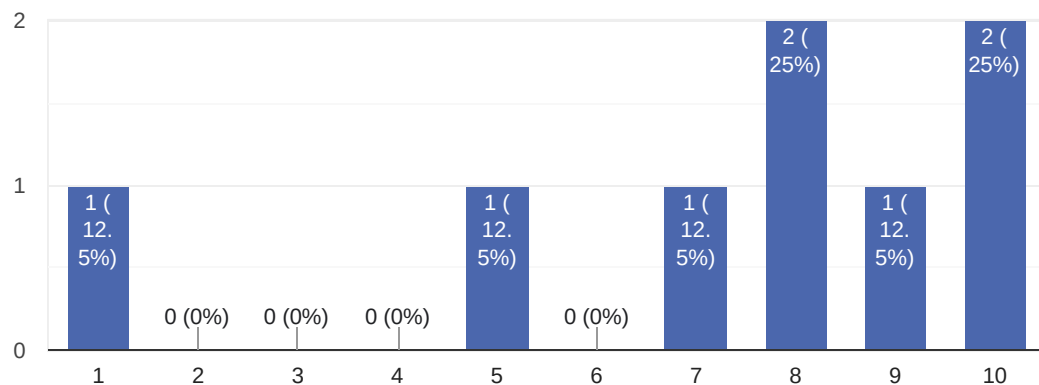
Comments and suggestions

5 responses

Some small exercises done by students would be appreciated to be able to take more from the training

Would have loved to have a proper "Labs" PDF like in others bootlin formation
To hava a guided practical experiment. The github repos are great as it provides nice example but it lacks a list of exercise that we could do to practice what we learned during the course

Very useful to understand the complicated theory

not many practical demos, maybe include debugging/monitor tools ans some examples on common hazards that a kernel developers face whenever there's a problem with graphics
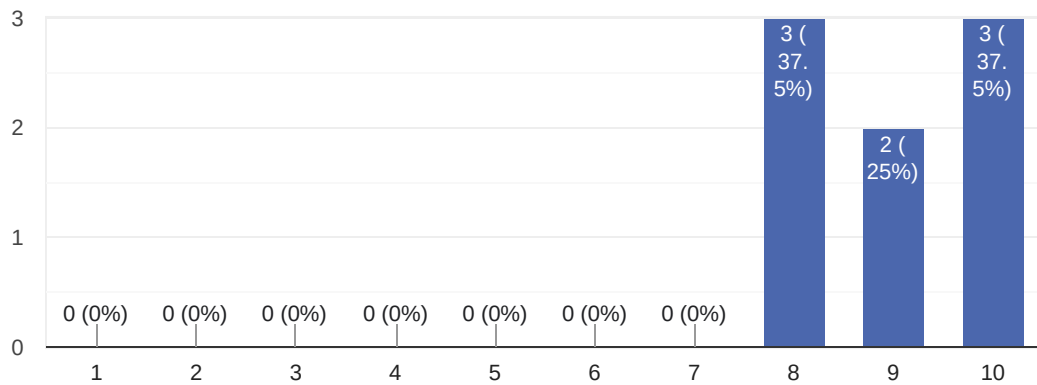
They were enough.

How would you rate the overall organization of the course?

8 responses

Bar chart:
- 1: 0 (0%)
- 2: 0 (0%)
- 3: 0 (0%)
- 4: 0 (0%)
- 5: 0 (0%)
- 6: 0 (0%)
- 7: 0 (0%)
- 8: 3 (37.5%)
- 9: 2 (25%)
- 10: 3 (37.5%)

Comments and suggestions

2 responses

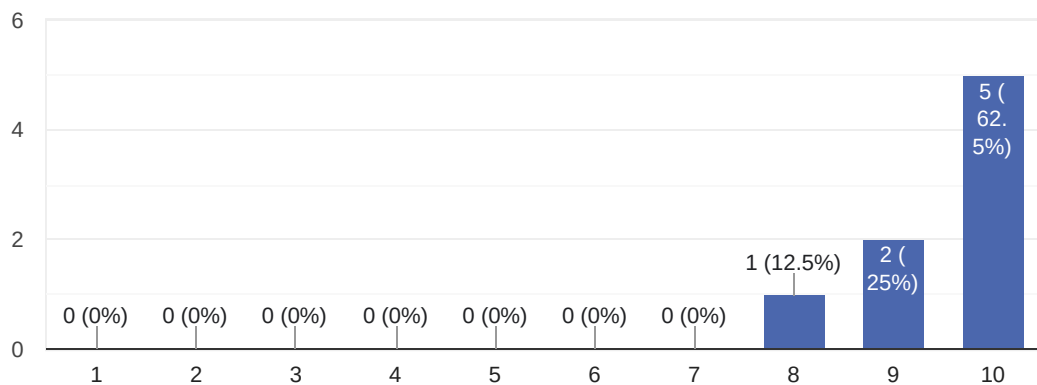More time should be reserved for software stack.

Overall organization was good.

How would you rate the trainer?

8 responses

Bar chart:
- 1: 0 (0%)
- 2: 0 (0%)
- 3: 0 (0%)
- 4: 0 (0%)
- 5: 0 (0%)
- 6: 0 (0%)
- 7: 0 (0%)
- 8: 1 (12.5%)
- 9: 2 (25%)
- 10: 5 (62.5%)

Comments and suggestions

2 responses

He showed domain of the subjects but I would love to saw him delivering more practical examples
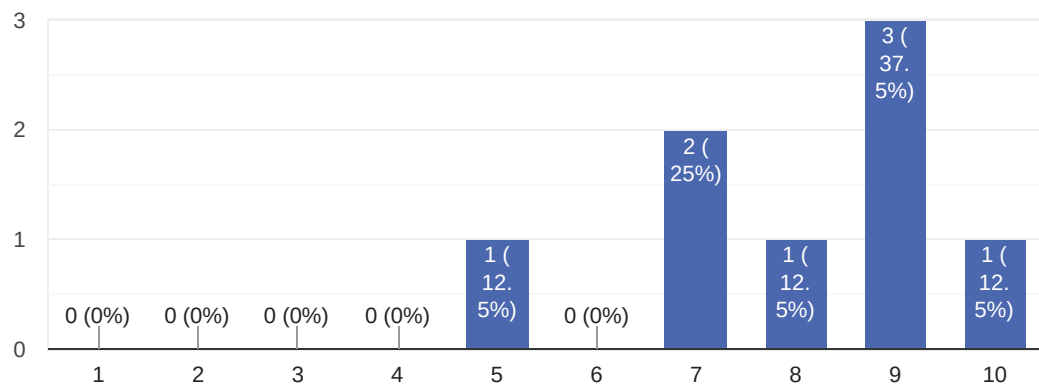
He has knowlegde about the different topics covered in the training.

How did the course meet your learning objectives?

8 responses

Comments and suggestions

1 response

I was looking for a more deep Kernel DRM knowledge, although he delivered some good information I would love to have hands on hardware and actually go and play with kernel subsystem

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

6 responses

system walkthrough

Explanation of different 3D APIs

Practical demos and software stack

DRM examples

linux Kernel DRM

Software aspects => kernel part

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

5 responses

basic theory

Part about picture, anti-aliasing, color models etc.
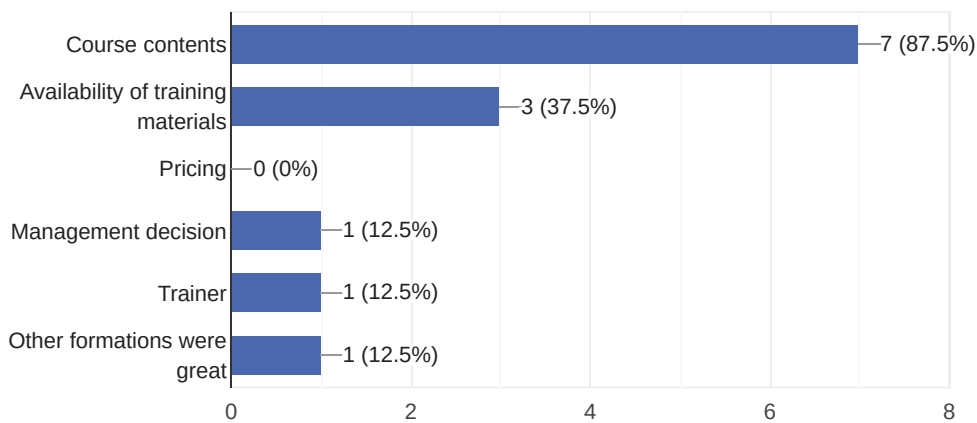I already knew about this stuff, it's generally taught during engineering studies

Nothing

N/A very useful information! maybe next time more practical than theoric

There was not any in particular

What reasons prompted you to choose a Bootlin course?

Copy

8 responses

Course contents — 7 (87.5%)
Availability of training materials — 3 (37.5%)
Pricing — 0 (0%)
Management decision — 1 (12.5%)
Trainer — 1 (12.5%)
Other formations were great — 1 (12.5%)

Comments

2 responses

I would love to see a more practical approach where we can see common practices to solve problems with graphics, what to pay attention whenever writing a kernel driver, tips on how to identify and avoid problems within DRM stack

Other bootlin material are freely available.

Further training needs?

4 responses

Nice introduction to the Linux graphics world. Maybe a formation which really dives deeply into kernel + userspace side for 3D graphics would be great ? With pratical examples and labs (ex: writing a panel kernel driver + integrating 3D graphics via an embedded board)

Linux kernel development
Embedded Linux audio

Kernel graphics subsystem, complex hardware architectures, stream video over PCI, more examples on how to properly use DMA within graphics kernel stack

If there are particular trainings related to software aspects, boot kernel and usersapace.

Google Forms