

KDLP: Expanding the Linux Kernel Talent Pipeline

Rado Vrbovsky & Denis Aleksandrov

Background of KDLP

- Kernel Development Learning Pipeline
 - Aging senior kernel engineers
 - Lack of fresh talent
 - An educational and mentoring program aiming to produce new kernel engineers
- Began as a for-credit course at the University of Massachusetts Lowell (UML)
- Expanded to three other universities around the world

Value of KDLP

- All materials are open sourced
- Various iterations of the course resulted in nine full time hires
- Forms bonds between international universities
 - UMass Lowell (UML) in Lowell, MA USA
 - Masaryk University (MUNI) in Brno, Czech Republic
 - University of Texas in San Antonio, Texas USA
 - Technion in Haifa, Israel

Connections to KDLP

- Rado:

- Met with Joel Savitz at LPC2023
- Started a new course for Brno FI MUNI
- Teaching with colleagues across Red Hat
 - KDLP: Content creator, lecturer, project lead

- Denis:

- Participated in a free online version of the course
- Invited to participate in an internship at Red Hat
 - KDLP: content creator, grader, and administrator
- Finished bachelor's degree
- Invited back for a full time position at Red Hat
 - KDLP: liaison

Problem statements

- Varying requirements
- Content divergence
- Scaling

Varying Requirements

- Pace of teaching, depending on the cohort
- Different instructors focus on different topics
- Rules are determined by the university or educational system
- Solution: create adaptable content
 - Cover more topics
 - Content modularity
 - Sort content by difficulty

Content Divergence

- Different iterations deliver different material
 - Natural, but we're aiming to converge the content
 - We have a way to measure the divergence
- Solution: standardize content
 - Create a common pool of materials
 - Larger than one semester
 - Instructor compiles syllabus depending on the students' level

Scaling

- Teach at more universities
- Reach more students
- Improve content quality
- Invite more experienced kernel engineers as guest speakers
- Solution: harness more community involvement

Current program skill-path

| ID | Skill | KDLP MUNI | KDLP US |
|----|-------------------------------------|-----------|---------|
| 22 | Writing character device drivers | N | Y |
| 23 | Writing block device drivers | N | N |
| 24 | Interrupt handling | Y | Y |
| 25 | Synchronization primitives | Y | Y |
| 26 | Race conditions and memory barriers | N | Y |
| 27 | Virtual memory areas and mappings | Y | Y |
| 28 | Filesystems and VFS | Y | Y |
| 29 | Networking stack internals | Y | N |
| 30 | Kernel debugging basics | Y | Y |
| 31 | Using ftrace and perf | Y | Y |
| 32 | Using crash and kdump | Y | N |
| 33 | Dynamic tracing with eBPF | Y | Y |
| 34 | Submitting patches | Y | Y |
| 35 | Mailing list etiquette | Y | Y |

TLDR

- Generate more flexible content to fit varying knowledge bases
 - Cover more topics
 - Content modularity
 - Sort content by difficulty
- We need content standardization
 - What are the candidates expected to know when joining the industry?
 - What skill-set makes a good and valuable kernel engineer?
- Please work with us!

Ways to contribute

- Teaching
- Grading
- Proofreading
- Generating and/or updating content

Upcoming conversations

- Collaboration with SUSE
- Discussions are starting in January 2026
- Reach out if you want to join the discussion!

Resources

Past course iterations:

<https://kdlp.underground.software/course/index.html>

Reach out to us:

rvrbovsk@redhat.com

daleksan@redhat.com