АВРОРА
СВОЯ СИСТЕМА

Lab 4 Kernel Modules

Ildar Kamaletdinov – team lead, Open Mobile Platform

with Dmitrii Alekhin – junior software developer as TA

TASK, part 1

- You need to implement a chardev kernel module **int_stack.ko**. This kernel module must implement a **stack<integer>** data structure with push/pop operations and also support stack size configuration via ioctl

- Implement a stack data structure:
  1. Allocate memory dynamically
  2. Use synchronization mechanisms in order to make your module ready for multithreading access (i.e. solve a classic readers-writers problem)

- Implement the following **file_operations**:
  1. **open()** and **release()** - initialization and deinitialization of the stack
  2. **read()** - *pop* operation
  3. **write()** - *push* operation
  4. **ioctl()** - configure max stack size

- Error codes must be the same as described in stack(3) manual (see Return codes section). Precisely:
  1. popping from the empty stack -> return **NULL**
  2. pushing into the full stack -> return **-ERANGE** errno (error codes must be negative according to kernel modules coding style, you can read this here).
  3. **ioctl()** errors codes are described in the manual as well

- Ensure that every edge case is handled properly (i.e. stack is empty, stack is full, etc.)

TASK, part 2

› Implement a small userspace utility **kernel_stack** that wraps your module functionality to the following user-friendly CLI:

```
$ kernel_stack set-size 2
$ kernel_stack push 1
$ kernel_stack push 2
$ kernel_stack push 3
ERROR: stack is full
$ kernel_stack pop
2
$ kernel_stack pop
1
$ kernel_stack pop
NULL
```

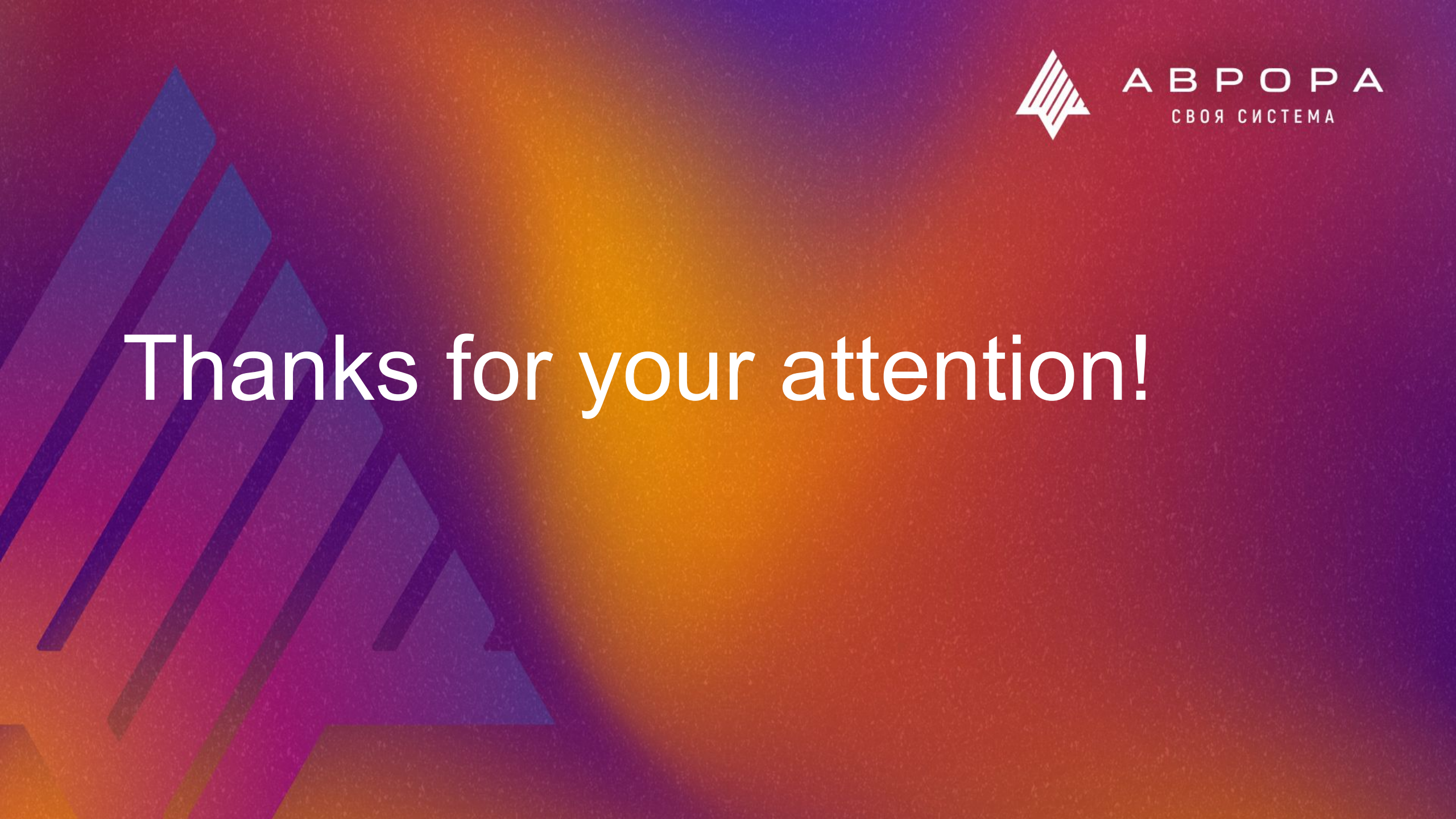
```
$ kernel_stack set-size 3
$ kernel_stack push 1
$ kernel_stack push 2
$ kernel_stack push 3
$ kernel_stack unwind
3
2
1
```

```
$ kernel_stack set-size 0
ERROR: size should be > 0
$ kernel_stack set-size -1
ERROR: size should be > 0
$ kernel_stack set-size 2
$ kernel_stack push 1
$ kernel_stack push 2
$ kernel_stack push 3
ERROR: stack is full
$ echo $?
-34       # -ERANGE errno code
```

› Graded output: source code with report including screenshots. (in PDF)

Acceptance criteria

- **A (20 points) –** kernel module properly implemented, dynamic memory is used, data structure is protected for simultaneous access (mutexes, spinlocks, etc.), error processing implemented, edge cases properly handled. Compilation warnings, some minor and style issues are acceptable
- **B (15-19 points)** – minor issues not related to overall usability (for ex. loctl() call does not return correct error code).
- **C (10-14 points)** – major issues (for ex. not enough locks, wrong locks usage and etc.)
- **D (<10 points)** – module more or less works but no locking implemented or no dynamic memory allocations are used.


АВРОРА
СВОЯ СИСТЕМА
Thanks for your attention!

About US

Open Mobile Platform, LLC

Shortly:

- Founded in 2016
- Offices in Moscow, Nizhny Novgorod, Innopolis and St.Petersburg
- 300+ qualified IT specialists

Main products:

- OS Aurora + Aurora SDK
- Cloud Platform
  Aurora Center (Enterprise Mobility Management)
- Aurora TEE & Trusted Boot