

# Linux Kernel Programming

## Flash Project Prototype Presentation

Pierre Olivier

Systems Software Research Group @ Virginia Tech

April 6, 2017

# General presentation

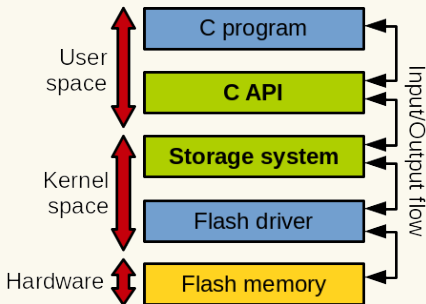
- ▶ **Prototype: in-kernel, on-flash key-value store**
- ▶ Example of ("key", "value") couple: ("name", "pierre")

- ▶ **Operations:**

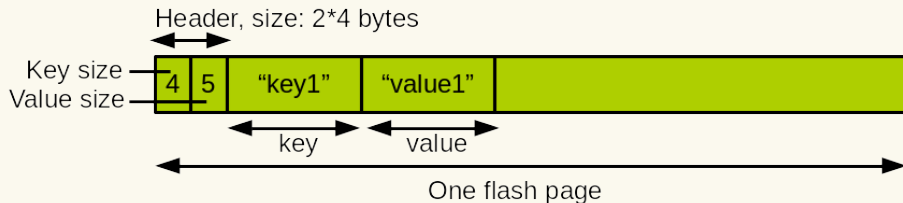
- ▶ `set(key, value)`
- ▶ `value = get(key)`
- ▶ `format()`

- ▶ **Integration in the software stack:**

- ▶ Use the MTD driver to perform flash operations
- ▶ A C API is provided for use from user-space programs



# Storage format



- ▶ One key/value couple per page
- ▶ Header with the key size and the value size (both integers) in the first 8 bytes of the page
- ▶ Followed by the key, then the value

# Operations

## ▶ **set (key, value)**

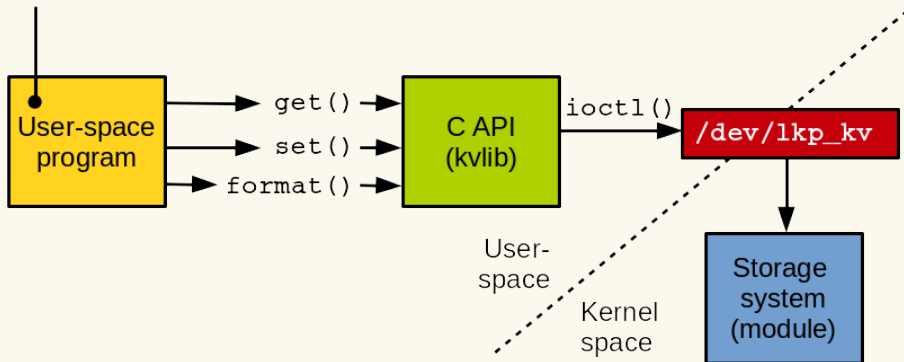
- ▶ Takes two parameters, the key and the value to insert in the storage system
- ▶ Scans the storage system to check that we are not inserting a duplicate key
- ▶ Then writes the couple in a free page
- ▶ The flash memory partition is written sequentially:
  - ▶ A free block (index b) is selected, then written page by page (flash constraint)
  - ▶ Block full? Selects b+1
  - ▶ Flash full? system switches to **read-only mode**, no more set operations possible
- ▶ At module insert time, flash is scanned to determine the next page to write

## Operations (2)

- ▶ **value = get (key)**
  - ▶ Returns the value corresponding to a key (parameter) if present in the storage system
  - ▶ Scans the entire used flash space page by page searching for the key
  - ▶ If found, returns the corresponding value
- ▶ **format ()**
  - ▶ Erase the entire content of the flash memory: all key/value couples stored are lost

# Usage from user-space

```
#include <kvlib.h>
```



# Limitations

- ▶ **Functionalities?**
- ▶ **Performance? Scalability?**