Linux Device Driver

LKDD – Framework & VFS

https://community.ruggedboard.com

LKDD-Framework & VFS

What is a device driver?

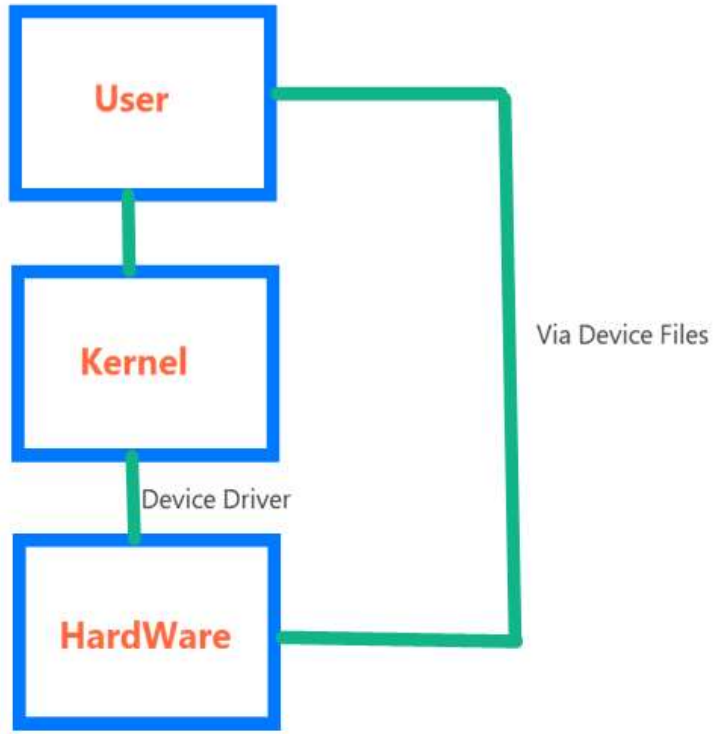
A device driver (often referred to as driver') is a piece of software that controls a particular type of device which is connected to the computer system

- A device driver is a piece of code that configures and manages a device.
- The device driver code knows, how to configure the device, sending data to the device, and it knows how to process requests which originate from the device.
- When the device driver code is loaded into the operating system such as Linux, it exposes interfaces to the user-space so that the user application can communicate with the device.
- Without the device driver, the OS/Application will not have a clear picture of how to deal with a device

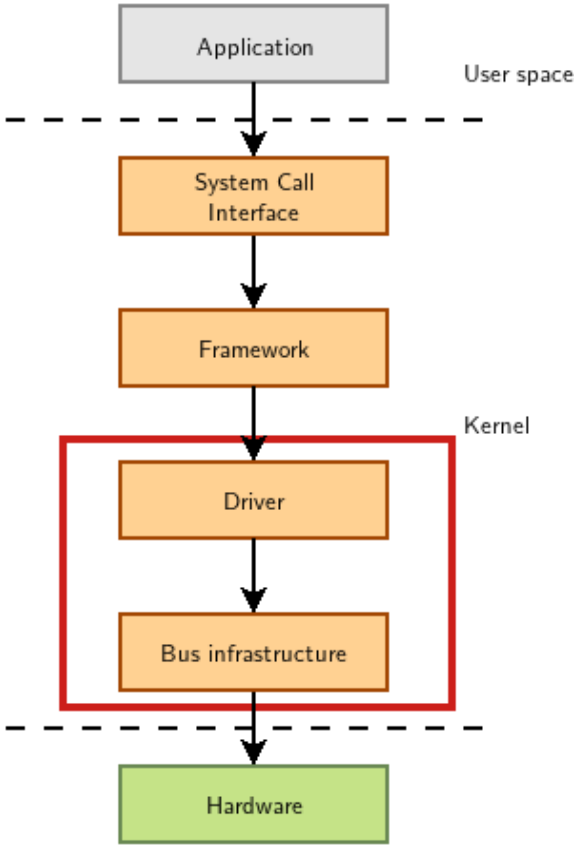
LKDD-Framework & VFS

A device driver has three sides:

- one side talks to the kernel
- one talks to the hardware,
- one talks to the user



LKDD-Framework



In Linux, a driver is always interfacing with:

- a *framework* that allows the driver to expose the hardware features in a generic way.
- a *bus infrastructure*, part of the device model, to detect/communicate with the hardware.

LKDD-Framework & VFS

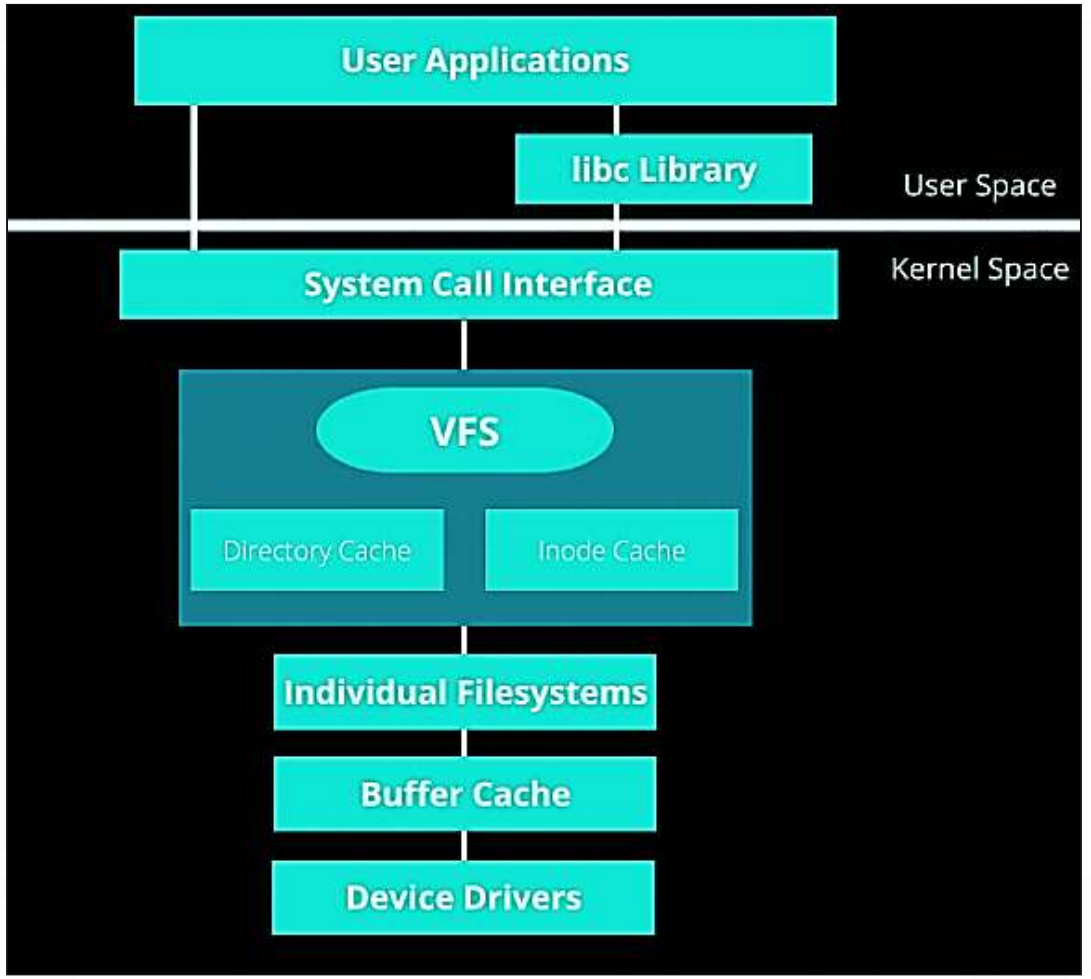
VFS

A **virtual file system** (**VFS**) is an abstract layer on top of a more concrete file system.

The purpose of a VFS is to allow client applications to access different types of concrete file systems in a uniform way.

A VFS can, for example, be used to access local and network storage devices transparently without the client application noticing the difference.

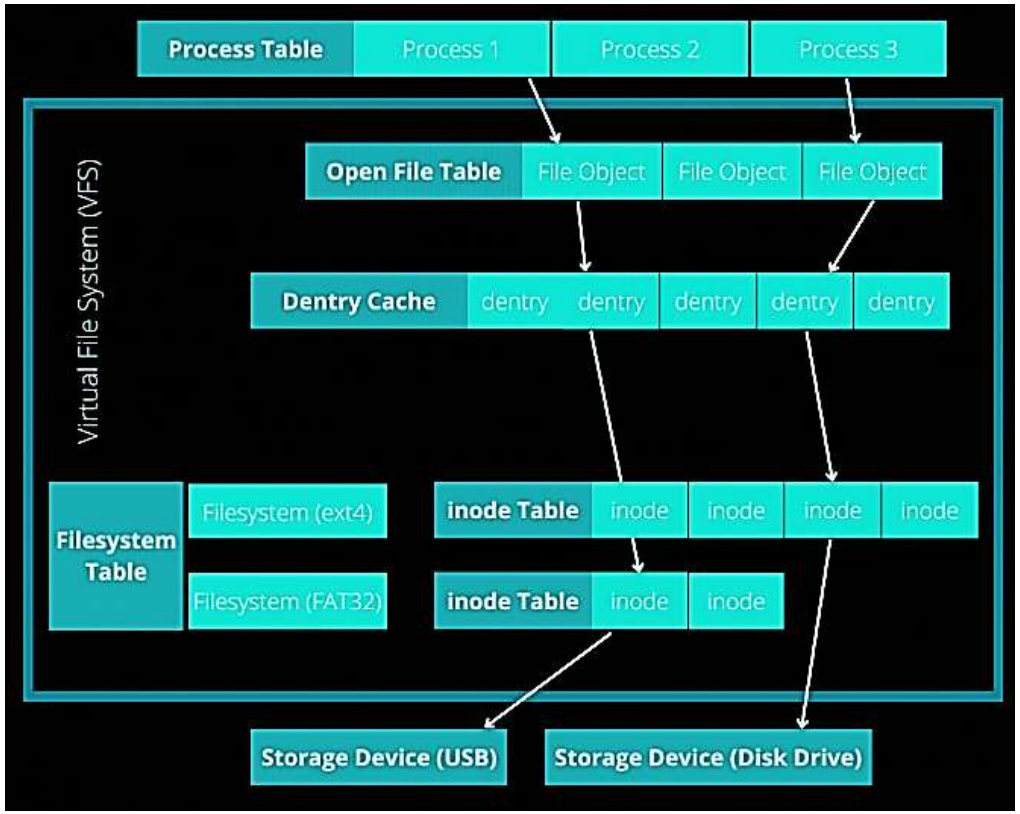
It can be used to bridge the differences in Windows, classic Mac OS/macOS and Unix filesystems, so that applications can access files on local file systems of those types without having to know what type of file system they are accessing.



LKDD-Framework & VFS

Data Structures we focus in VFS are:

- Filesystem Types
- Superblocks
- Inodes
- Dentries



- The VFS maintains a list of superblocks.
- A superblock represents a mounted filesystem.
- Each superblock maintains a list of inodes.
- An inode represents a file system object (i.e. file).

LKDD-Framework & VFS

A file system object can be one of the following types:

- socket
- symbolic link
- regular file
- block device
- directory
- **character device**
- FIFO

LKDD-Framework & VFS

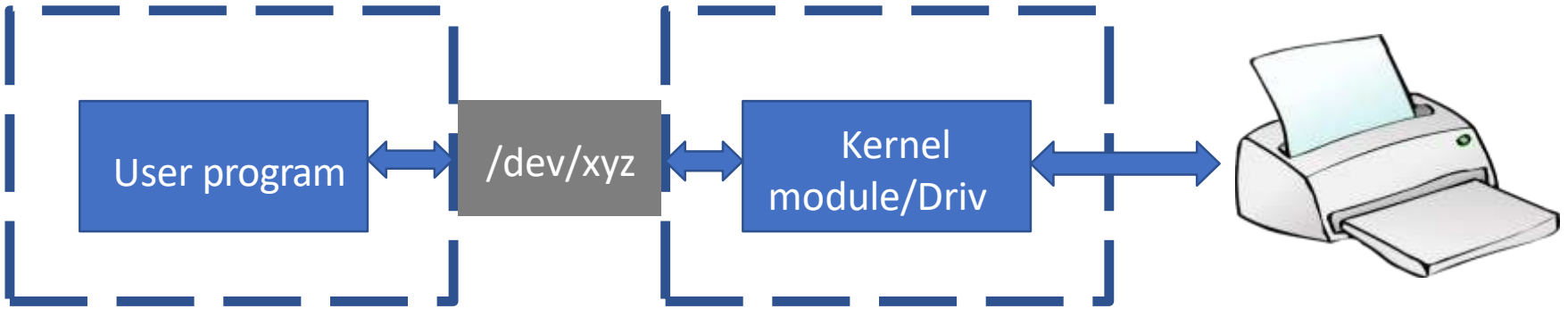
- In Linux everything is considered to be a file so devices are also considered to be a file.
- In Linux, hardware devices are accessed by the user through special device files.
- These files are grouped into the /dev directory, and system calls open, read, write, close, lseek, mmap etc. are redirected by the operating system to the device driver associated with the physical device.
- A device driver is a piece of software that operates or controls a particular type of device.
- A device file is an interface for a device driver that appears in a file system as if it were an ordinary file.

In the Linux world there are three categories of device drivers:

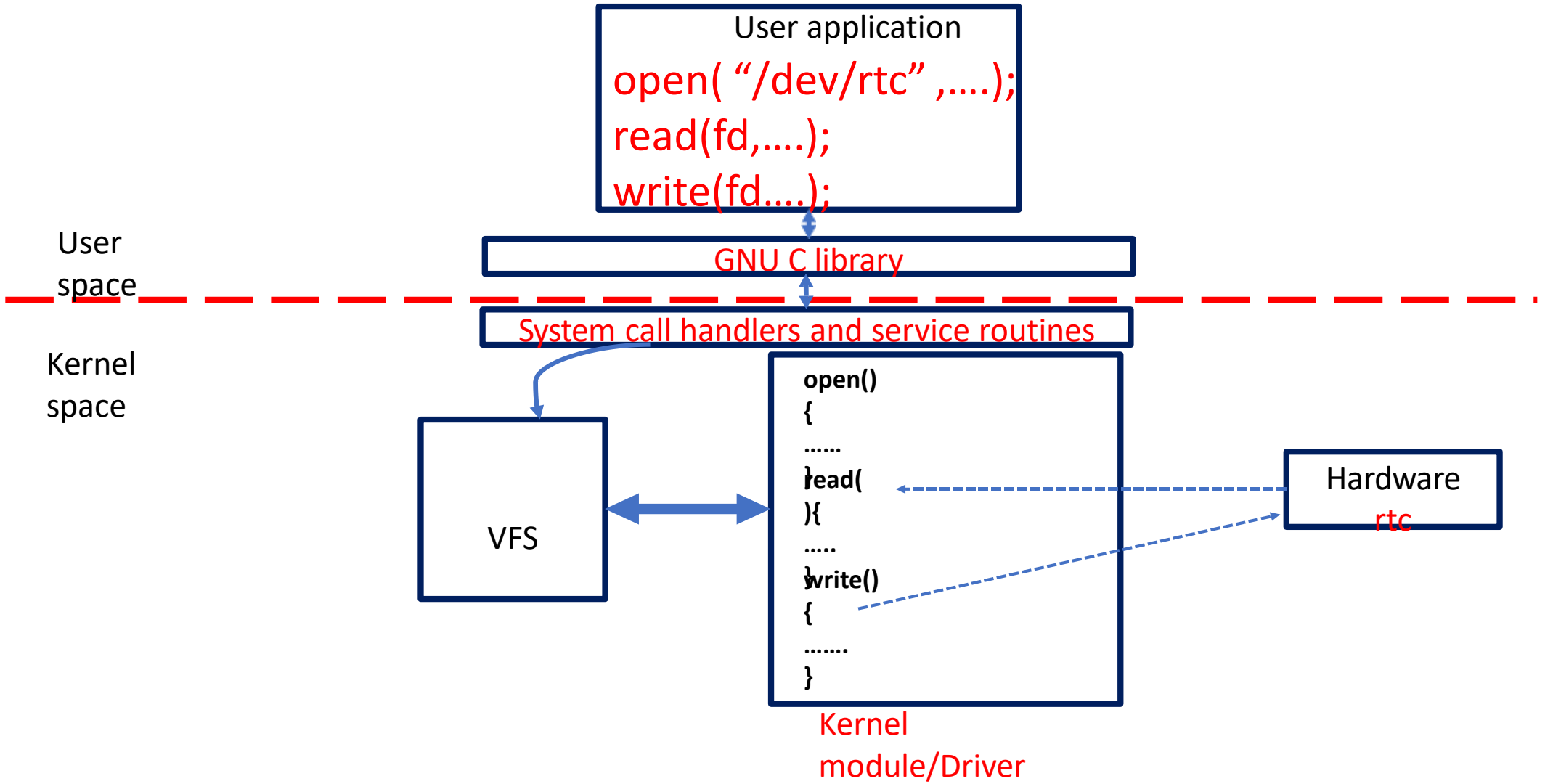
- **Character**
- Block
- Network

LKDD-Framework & VFS

- Devices are accessed as a file in Unix/Linux systems.
- A device file is a special file or a node which gets populated in /dev directory during kernel boot time or device/driver hot plug events
- By using device file, user application and drivers communicate with each other.
- Device files are managed as part of VFS subsystem of the kernel



LKDD-Framework & VFS



Connect-us

Developer Wiki

Open Discussions