

Linux Input & Output

By: Sagar Janagonda

Contents

- Introduction
- What is Device Driver?
- Block Structure of Device Driver
- Linux Device Driver Classes
- References

Introduction

The Input & Output in Linux looks much like it does in any Unix.

All device drivers appear as a normal files.

Devices can appear as objects in the file system.

System administrator can create special files within a file system that contains references to a device driver.

Administrator can also set permissions for each device.

Device Driver

- Also called as software driver.
- Computer program allowing higher-level computer programs to interact with a device.
- Communicates with the device through the computer bus or communications subsystem to which the hardware is connected.
- Linux device drivers can be built either as parts of the kernel or separately as loadable modules.

Device Driver Applications

Because of the diversity of modern hardware many ways exist in which drivers can be used.

- Printers
- Video adapters
- Network cards
- Sound cards
- Scanners and digital cameras etc.

Linux device-oriented file system accesses disk storage through two caches:

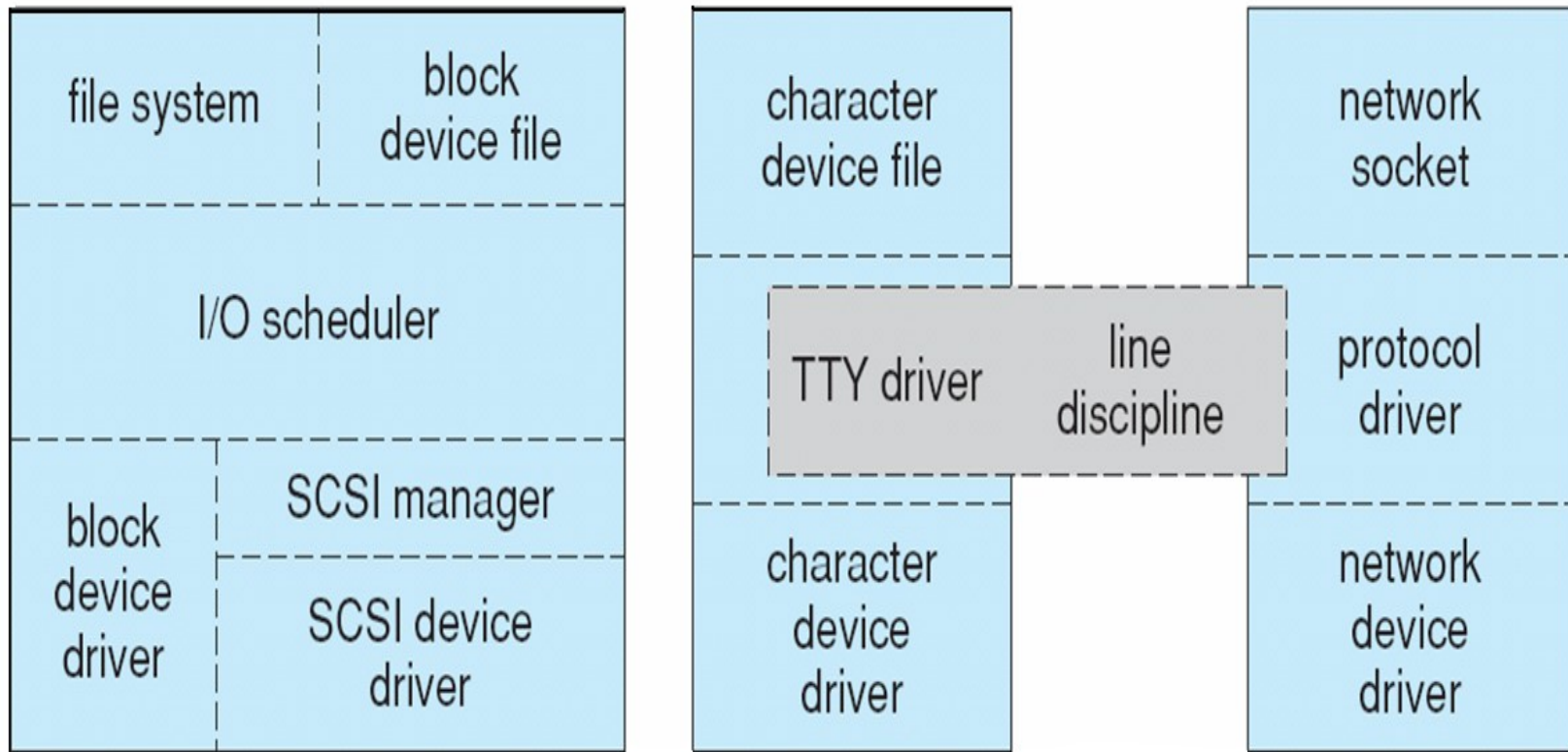
- Data is stored in the page cache, which is unified with the virtual memory system.
- Metadata is stored in the buffer cache, a separate cache indexed by the physical disk block.

Linux splits all devices into three classes:

- ***Block devices*** allow random access to completely independent, fixed size blocks of data.
- ***Character devices*** include most other devices; they don't need to support the functionality of regular files.
- ***Network devices*** are interfaced via the kernel's networking subsystem.

Device-Driver Block Structure

user application



SCSI : Small Computer System Interface

Is a set of standards for physically connecting and transferring data between computers and peripheral devices.

TTY

Is a Unix Command that prints to standard output the name of the terminal connected to standard input. The name of the program comes from teletypewriter.

Line discipline is a layer in the terminal subsystem.

It processes the data it receives from the hardware driver.

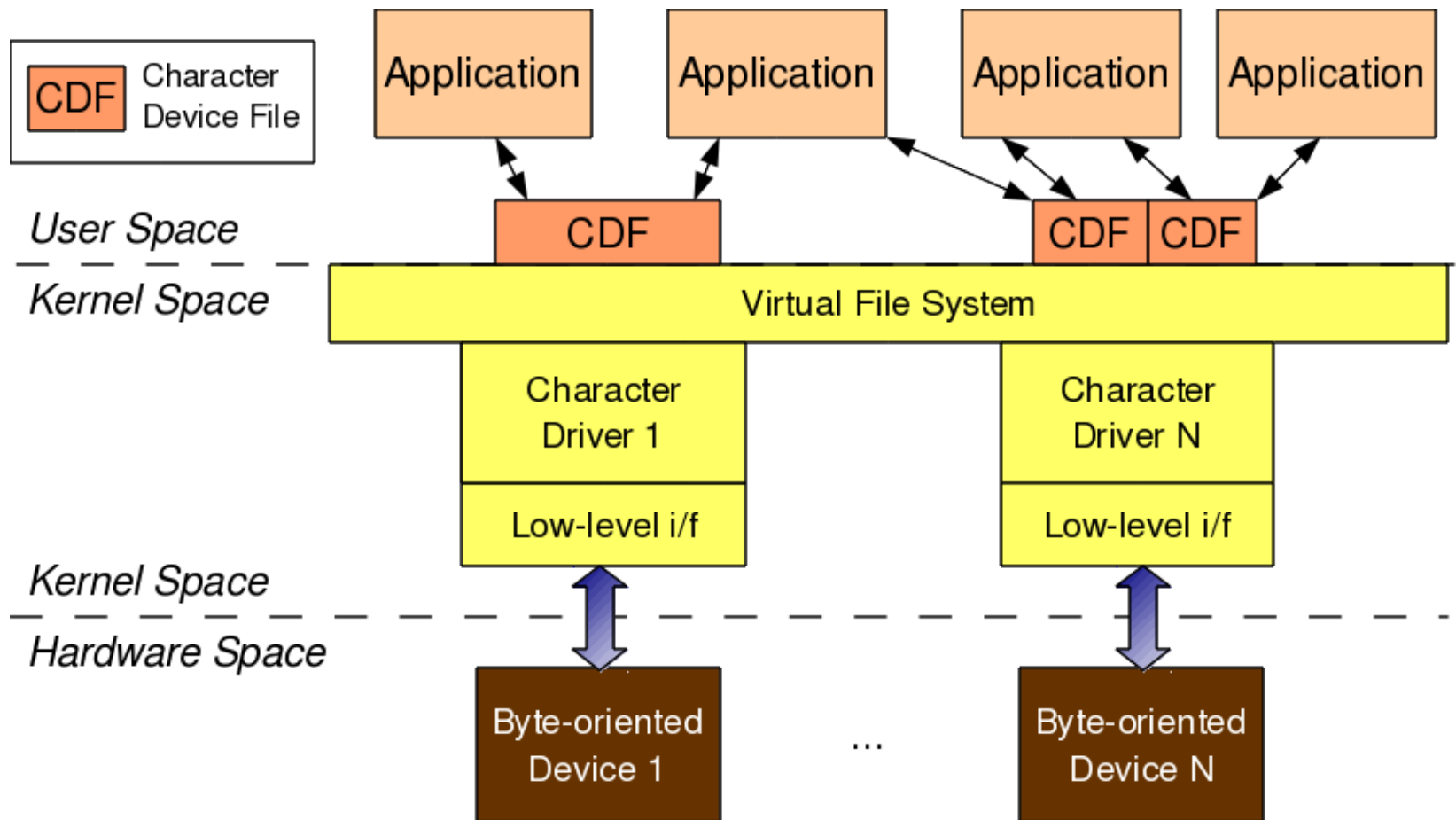
Block Devices

- Provide the main interface to all disk devices in a system.
- Used to store file systems.
- The *block buffer* cache serves two main purposes:
 - it acts as a pool of buffers for active I/O
 - Collection of data
 - it serves as a cache for completed I/O
 - Fast memory holding copy of data
- The *request manager* manages the reading and writing of buffer contents to and from a block device driver
 - e.g. disk drives, CD.

Character Devices

- A device driver which does not offer random access to fixed blocks of data.
- A character device driver must register a set of functions which implement the driver's various file I/O operations.
- The kernel performs almost no preprocessing of a file read or write request to a character device, but simply passes on the request to the device.
- The main exception to this rule is the special subset of character device drivers which implement terminal devices, for which the kernel maintains a standard interface.

e.g. keyboards, mouse, serial ports



- Character driver usage is done through the corresponding character device file.
- Linked through the virtual file system (VFS).
- Application does the usual file operations on the character device file.
- Operations are translated to the corresponding functions in the linked character device driver.
- Those functions then do the final low-level access to the actual device to achieve the desired results.

Network Devices

- Network transaction is made through an interface.
- Device that is able to exchange data with other hosts.
- Network interface is in charge of sending and receiving data packets.
- Network driver handles packets.
- Communication between the kernel and a network device driver is completely different from that used with char and block drivers.
- Instead of *read* and *write*, the kernel calls functions related to packet transmission.

References

Operating Systems by STEVEN HAND

Operating Systems by ABRAHAM SILBERSCHATZ

<http://linuxjournal.com/article/2890>

Thank You...!