

Character device drivers

Praktikum „Kernel Programming“

December 17, 2014 - Albrecht Oster, Walter König

Outline

What is a character device driver?

How can we use it?

What does it look like?

Let's write our own!

What is a character device driver?

two common types: character devices and block devices

character devices are byte-oriented

transfers a stream of bytes directly from kernel to user space

unbuffered and synchronous access

basic I/O (real serial devices, virtual devices)

Examples for character devices

most common type of device driver

is represented as a device file (e.g. `/dev/ttyS0`)

can be used like a regular file from user space

open, read, write, close

How does it fit in the Linux architecture?

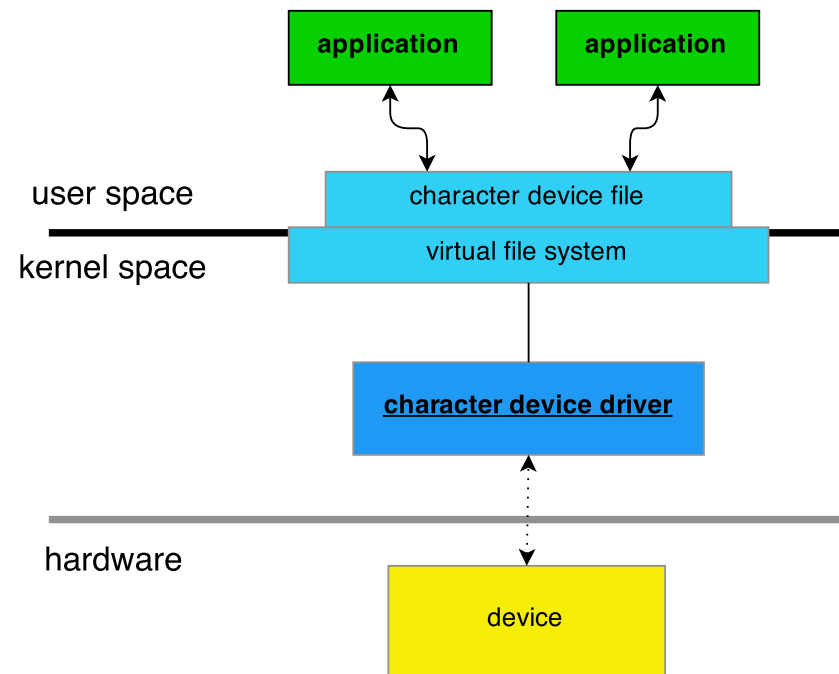
remains in kernel space

used by user space apps through the device file in the VFS

usual file operations

outcome may differ

the driver has final low level access to the actual device



How does it fit in the Linux architecture?

four entities involved

user space app

character device file (virtual file system)

character device driver (kernel space)

actual character device

Loading / Unloading

`module_init(function_ref)`

`module_exit(function_ref)`

`insmod` loads the driver

`rmmod` unloads the driver

Registering a device file

devices are referred to by major and minor numbers

```
int register_chrdev(unsigned int major, const char *name, struct
file_operations *fops);
```

driver is registered to one major number

several instances of the same driver are distinguished by minor numbers

can define file operations it supports

ioctl

represents a way to control the device itself

every device can have its own `ioctl` commands

defined in file operations struct

used by a user space app with a file descriptor and macro from header files

```
int ioctl(int file_handle, int request, char *data);
```

blocking I/O

restricting access to the device

single-user lock

system-wide lock

„device busy“

Literature

„Character Device Drivers“

<http://linux.die.net/lkmpg/x569.html>

„Enhanced Char Driver Operations“

<http://www.xml.com/ldd/chapter/book/ch05.html>

Wikipedia: „Device File“

http://en.wikipedia.org/wiki/Device_file

„Decoding Character Device File Operations“

<http://www.opensourceforu.com/2011/05/decoding-character-device-file-operations/>

„I/O Control in Linux“

<http://www.opensourceforu.com/2011/08/io-control-in-linux/>

„Mknod“

<http://linuxwiki.de/mknod>