A+ will be down for a version upgrade on Tuesday 03.01.2023 at 9-12.

This course has already ended.

« 6. Linux Driver (/os/2022/materials_m06/) 7. Synchronization and deadlocks » (/os/2022/materials_m07/)

CS-C3140 (/os/2022/) / 6. Linux Driver (/os/2022/materials_m06/) / 6.1 A Linux Driver

A Linux Driver

Exercise 1

⚠ The deadline for the assignment has passed (Sunday, 13 November 2022, 23:59).

Writing a Simple Driver for Linux

The following Google Colab includes the code and description of the assignment A Simple Driver with Linux Colab link (https://colab.research.google.com/drive/17BnqHxEeHRHwj6aAd9Jr2YorCS2o9Djd)

1. Writing the Driver

Question 1 2 / 2

TODO 1. What does MAX_DEVICE_NUM represent in our code?

- ○ MAX_DEVICE_NUM is a constant that represents the number of major devices controlled by the driver
- ○ MAX_DEVICE_NUM is a constant that represents the number of minor devices controlled by the driver
- ◉ **MAX_DEVICE_NUM is a constant that represents the maximum number of minor devices controlled by the driver**
- ○ MAX_DEVICE_NUM is a constant that represents the maximum number of major devices controlled by the driver

✔ Correct!

Question 2 2 / 2

TODO 2. Which of the following statements are true regarding struct file_operations?

- ☑ **The first field is a pointer to the module that "owns" this structure and it is used by the kernel to maintain the module's usage count. Setting an owner is used to prevent the module from being unloaded while its operations are in use.**
- ☐ The signature of the functions cannot differ from the system call that the user uses
- ☑ **The file_operations structure holds pointers to functions defined by the driver that perform various operations on the device**
- ☑ **Any member of the structure which you don't explicitly assign will be initialized to NULL by gcc**

✔ Correct!

Question 3 2 / 2

TODO 3. What is the use of alloc_chrdev_region?

- ☐ alloc_chrdev_region tells the kernel how many device numbers we need, but it will not find a starting major number for us (even though there are numbers available)
- ☑ **alloc_chrdev_region allocates and registers a range of char device numbers. The third parameter may be equal to MAX_DEVICE_NUM, to cover all possible minor numbers**
- ☑ **The kernel will dynamically allocate a major number for our character device if this function is called**
- ☐ The first parameter should always be equal to MAX_DEVICE_NUM

✔ Correct!

Question 4 2 / 2

TODO 4. What is true about device initialization?

- [ ] Device initialization after calling alloc_chrdev_region is synonymous to registering it with the kernel
- [ ] Device initialization creates and registers a cdev occupying a range of minors. The function used for initialization has the major as a parameter, and if this parameter is set to 0, this function will dynamically allocate a major and return its number.
- [x] **Device initialization does not make the device available to the system immediately**
- [x] **Device initialization has the file operations for the device as an argument**

✔ Correct!

Question 5 2 / 2

TODO 5. What is true about setting the owner of the device in this case?

- [x] **Owner field of the structure should be initialized in order to protect against ill-advised module unloads while the device is active.**
- [x] **The owner is initialized using a macro defined in <linux/module.h>**
- [x] **Setting the owner of the device is mandatory**
- [ ] Setting the owner of the device is not mandatory

✔ Correct!

Question 6 4 / 4

TODO 6. What is true about adding a device to the system in this case?

- [x] **The function used for adding the device has the number of consecutive minor numbers corresponding to the device as an argument**
- [x] **Together with initialization, the function used for adding a character device registers it to the VFS (Virtual File System)**
- [x] **The device is live immediately to the system after the function used for adding is called and the kernel can invoke its operations**
- [ ] The device is not live immediately to the system after the function used for adding is called

✔ Correct!

Question 7 2 / 2

TODO 7. What is true regarding device_create?

- [x] **After calling device_create, a struct device will be created in sysfs, registered to the specified class**
- [x] **If it succeeds, device_create will create a /dev/chdev-x device (where x is one of the i values) in the /dev directory**
- [x] **The struct class passed to this function must have previously been created with a call to class_create**
- [ ] The major number of the previously generated device is changed at this step

✔ Correct!

Question 8 2 / 2

TODO 8. What is true regarding device_destroy?

- [x] **Its call unregisters and cleans up a device that was created with a call to device_create**
- [ ] One cannot call device_destroy on a device that is still opened by a process
- [x] **/dev/chdev-x (where x is one of the i values) won't be visible anymore after calling device_destroy**
- [x] **The function takes a pointer to the struct class the device was registered with as an argument**

✔ Correct!

2. Installing and Using the Driver

Question 9 2 / 2

TODO 9. What is true regarding the command?

- [x] **insmod is used to insert the LKM file into the Linux Kernel. insmod calls the chdev_init() function**
- [x] **The 'make' command reads the Makefile present in current directory and executes based on the statements in Makefile**
- [x] **The insmod is called in this case only if the make succeeds**
- [ ] The insmod is called in this case even if make does not succeed

✔ Correct!

Question 10 2 / 2

TODO 10. What is true regarding the command?

- ○ We cannot use this command to print all devices we just created
- ○ After running the command, the output consists of three or more lines that start with 'crw'

- ○ After running the command, the output consists of only one line that starts with 'crw'
- ◉ **After running the command, the output consists of exactly two lines that start with 'crw'**

✔ Correct!

Question 11 **2 / 2**

TODO 11. What is true regarding the command?

- ☐ We can use this command to check build & load messages for both devices at the same time
- ☐ After running the command, the output consists of build errors
- ☐ The command outputs nothing
- ☑ **After running the command, the output contains lines with the message 'Hello from the kernel world!'**

✔ Correct!

Question 12 **4 / 4**

TODO 12. What is true regarding the command?

- ☐ After the first read, data from the user is "Writing character device drivers is not that hard!" and the writing device is 0
- ☑ **After the first read, data from the user is "Writing charact" and the writing device is 0**
- ☐ After the first read, data from the user is "Writing charact" and the writing device is 1
- ☑ **After the second read, data from the user is "I do agree!" and the writing device is 1**

✔ Correct!

Submit

« 6. Linux Driver (/os/2022/materials_m06/) 7. Synchronization and deadlocks » (/os/2022/materials_m07/)