

SO2 Session 27 - Lab 03 (Character driver: from Register to Delete)

Registration and the unregistration of character devices

- specifying the major and minor
- the dev_t used to keep the identifiers
- get the major and minor via MKDEV macro

1) primary line for all code
include <linux/fs.h>
include <linux/cdev.h>

struct my_device {
 struct cdev cdev;

};

2) Registration
int register_chrdev_region (
 dev_t first ,
 unsigned int count,
 char *name)
int unregister_chrdev_region (
 dev_t first ,
 unsigned int count);

3) Initialization
void cdev_init (
 struct cdev * cdev ,
 struct file_operations *fops);
int cdev_add (
 struct cdev *cdev,
 dev_t nm,
 unsigned int count);
int cdev_del (
 struct cdev *cdev);

init → register_chrdev_region
init (fops [open, read, ...])
add

exit → del
unregistration

Access to the address space of the process

Kernel space map some part of the user space to use it (but be careful of the raising error of buffer)

direct access can lead to problem so that's why we adopt the use of this micro :
copy_to_user (
 void __user *to,
 const void *from ,
 unsigned long n);

copy_from_user (
 void *to,
 const void *from ,
 unsigned long n);

Open and release

A problem were occur when implementing the open function is the access control :
opened once a time , for example :
d1 → d2
 ↔ (read one time)
=> do not allow the second open before the release

solution is -EBUZY