```
// ...
#include  <linux/slab.h>:
// ...
struct _instance_data {
    int counter;
};

static ssize_t driver_read(struct file *instanz, char *user, size_t count, loff_t *
offset)  {
    char helloWorld[] = "hello world\n";
    unsigned long notcopied;
    size_t to_copy;
    struct _instance_data *iptr;

    iptr = (struct _instance_data*) instanz->private_data;

    to_copy = iptr->counter;
    to_copy =  min(to_copy, count);
    if (to_copy <= 0) {
        return 0;
    }

    notcopied = copy_to_user(user, helloWorld, to_copy);
    iptr->counter = iptr->counter - to_copy + notcopied;
    return to_copy - notcopied;
}
static int driver_open(struct inode *geraetedatei, struct file *instanz) {
    struct _instance_data *iptr;

    iptr = (struct _instance_data *)kmalloc(sizeof(struct _instance_data), GFP_KERN
EL);
    if (iptr == 0) {
        printk("oops, not enough kern mem hehe\n");
        return -ENOMEM;
    }


    iptr->counter = strlen("hello world\n") + 1;
    instanz -> private_data = (void *) iptr;

    return 0;
}

static int driver_release(struct inode *geraetedatei, struct file *instanz)  {
    if (instanz->private_data) {
        kfree(instanz->private_data);
    }
    return 0;
}
// ...
```