

CS333 - Project 1

Linux kernel module driver

Thành viên nhóm:

1751044-Ngô Lê Phương Trinh

1751027-Huỳnh Thanh Vy

1751112-Nguyễn Thị Thanh Trúc

1. Function

File RanNum_drv.c

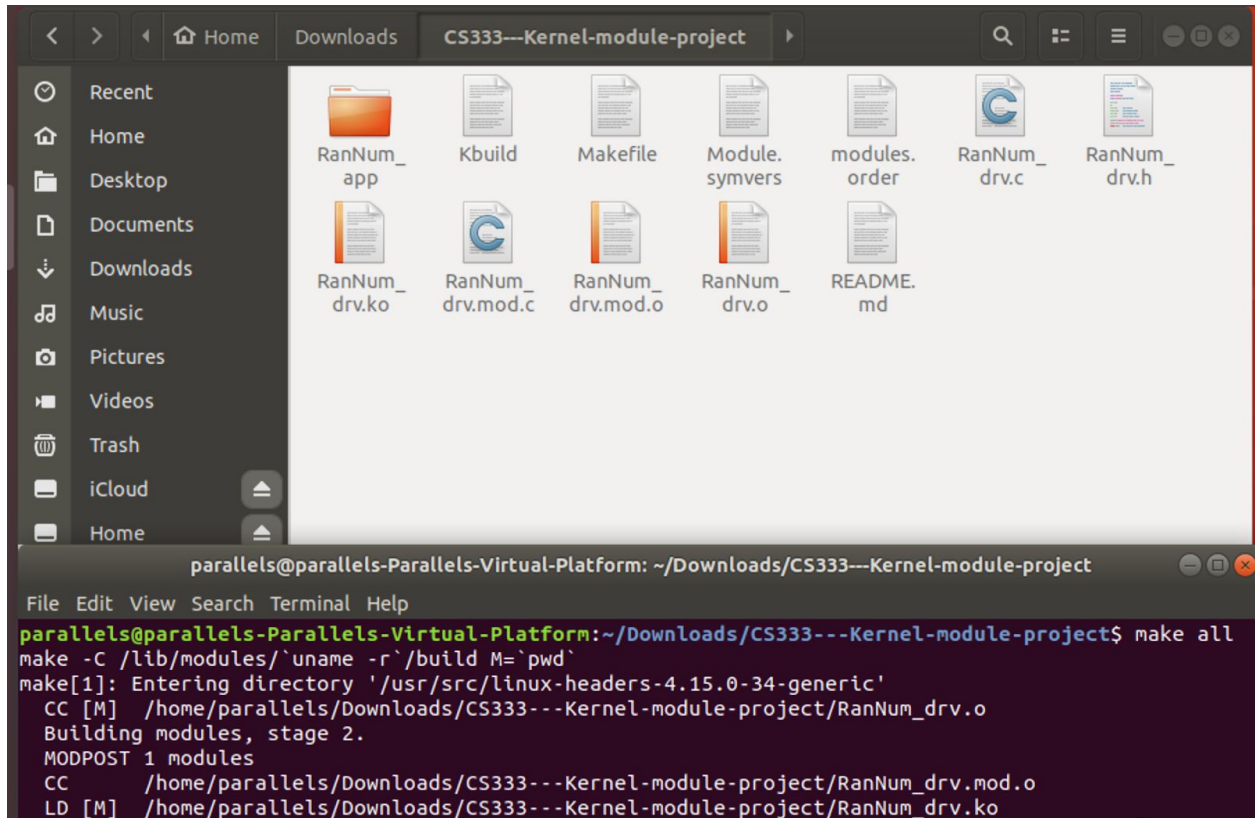
	Function	Description
OS specific	<pre>// Ham entry point Open static int RanNum_driver_open(struct inode *inode, struct file *filp)</pre>	The Linux kernel has a structure used to describe character devices - <code>cdev</code> . Whenever the char driver wants to control a character device, the char driver has to "submit" its <code>cdev</code> structure to the Linux kernel. The <code>cdev</code> structure has a field of <code>file_operations</code> structure type that describes what functions do, corresponding to a system call. Each function is called an entry point.
	<pre>// Ham entry point Release static int RanNum_driver_release(stru ct inode *inode, struct file *filp)</pre>	This function is invoked to cause the device to stop working. If it returns 0, then releasing the device file is successful. Otherwise, fail to release the device file when it returns an integer different from 0.
	<pre>// Ham entry point Read Random Number static ssize_t RanNum_driver_read(struct file *filp, char *user_buffer, size_t len, loff_t *offset)</pre>	Use <code>get_random_bytes(&randomNumber, sizeof(randomNumber))</code> to generate a random number with the size is sizeof(char) which is assigned to parameter randomNumber . Then split up each character and append it to kernel_buffer string. Input the use_buffer string and size of len which is received from User Space. Output the string that contains numbers to kernel

		log. Use copy_to_user function to copy from kernel buffer to user buffer so that random number can be displayed in user console.
	<pre>// Ham khoi tao driver static int __init RanNum_driver_init(void)</pre>	This function does some jobs: allocate device number, create device file, allocate data structure of driver, initialize a virtual character device, register entry point with kernel.
	<pre>// Ham ket thuc driver static void __exit RanNum_driver_exit(void)</pre>	The function does some jobs: unregister entry point with kernel, release a virtual character device, deallocate memory for driver data, remove device file, release device number.
Device specific	<pre>// Ham khoi tao thiet bi int vchar_hw_init(vchar_dev_t *hw)</pre>	In order for the char driver to interact with the device "vchar_device", we define the structure <code>vchar_dev_t</code> , and also add the <code>*vchar_hw</code> field into the <code>vchar_drv</code> structure. The process of interacting with the device including device initialization, release, read / write device registers. This function initializes the registers of the device "vchar_device".
	<pre>// Ham giai phong thiet bi void vchar_hw_exit(vchar_dev_t *hw)</pre>	This function releases the registers of the device "vchar_device".

In order to test whether the char device has operated exactly, we create a folder name `RanNum_app`. Inside the folder is the program `RanNum_generate.c`. This is a program on user space which interacts with the `RanNum_driver` through **open and close** system calls to check the operation of **open and release** entry points. The system calls **read** function to read from the driver and show a random number in the console of the application.

2. How to build and run “Generate random number module”

- `make all` to build `RanNum_drv.ko`



- `sudo insmod RanNum_drv.ko` to install module to kernel
- `cat /proc/devices` to check whether device has installed and the major number of driver

```

parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$ sudo insmod RanNum_drv.ko
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$ cat /proc/devices | grep RanNum_drv
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$ cat /proc/devices
Character devices:
 1 mem
 4 /dev/vc/0
 4 tty
 4 ttys
 5 /dev/tty
 5 /dev/console
 5 /dev/ptmx
 5 ttyprintk
 6 lp
 7 vcs
10 misc
13 input
21 sg
29 fb
81 video4linux
89 i2c
99 ppdev
108 ppp
116 alsa
128 ptm
136 pts
180 usb
189 usb_device
204 ttyMAX
226 drm
242 RanNum_device
243 media
244 hidraw
245 aux
246 bsg
247 hmm_device
248 watchdog
249 rtc
250 dax
251 dmmctl
252 ndctl
253 tpm
254 gpiochip

```

Or you could use `cat /proc/devices | grep RanNum_device` to see the direct major number of the device

```

parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$
cat /proc/devices | grep RanNum_device
242 RanNum_device

```

- `dmesg` to view kernel log -> You should see Initial RanNum driver successfully

```

[ 5257.697188] Exit RanNum driver
[ 5260.226560] Allocated device number (242,0)
[ 5260.226784] Initialize RanNum driver successfully
[ 5280.934710] Exit RanNum driver
[ 5294.353257] Allocated device number (242,0)
[ 5294.354897] Initialize RanNum driver successfully
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$

```

- `sudo chmod 666 /dev/RanNum_dev` to give access for user app
- `ls -la /dev/RanNum_dev` to check giving access successfully

```

[ 5257.697188] Exit RanNum driver
[ 5260.226560] Allocated device number (242,0)
[ 5260.226784] Initialize RanNum driver successfully
[ 5280.934710] Exit RanNum driver
[ 5294.353257] Allocated device number (242,0)
[ 5294.354897] Initialize RanNum driver successfully
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$ sudo chmod 666 /dev/RanNum_dev
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$ ls -la /dev/RanNum_dev
crw-rw-rw- 1 root root 242, 0 Apr 29 22:11 /dev/RanNum_dev
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$

```

- `cd RanNum_app`

- `make all` to build `RanNum_generate.c`

```
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$ cd RanNum_app/
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project/RanNum_app$ make all
cc -o RanNum_generate RanNum_generate.c
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project/RanNum_app$
```

- `./RanNum_generate` to run the program. Now you should see a random number is display in your program

```
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project/RanNum_app$ ./RanNum_generate

Reading from the device...
Random number is 352
```

- `dmesg` to check the random number in kernel log

```
[13000.292335] Allocated device number (242,0)
[13000.293991] Initialize RanNum driver successfully
[15604.522251] Handle open event (1)
[15604.522319] Handle read event
[15604.522321] Random number is 552
[15604.522322] Random number is [Kernel_buffer] 552
[15604.522336] Handle close event
[15752.343443] Handle open event (2)
[15752.343514] Handle read event
[15752.343516] Random number is 352
[15752.343517] Random number is [Kernel_buffer] 352
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project/RanNum_app$
```

- Make sure to remove the module after you finish running by using `sudo rmmod RanNum_drv`. Check whether it is actually removed by `lsmod | grep RanNum_drv.ko`

```
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project/RanNum_app$ sudo rmmod RanNum_drv
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project/RanNum_app$ cd ..
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$ lsmod | grep RanNum_drv.ko
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/CS333---Kernel-module-project$
```

Or using `dmesg` to check whether the module exit successfully.

```
[13000.292335] Allocated device number (242,0)
[13000.293991] Initialize RanNum driver successfully
[15604.522251] Handle open event (1)
[15604.522319] Handle read event
[15604.522321] Random number is 552
[15604.522322] Random number is [Kernel_buffer] 552
[15604.522336] Handle close event
[15752.343443] Handle open event (2)
[15752.343514] Handle read event
[15752.343516] Random number is 352
[15752.343517] Random number is [Kernel_buffer] 352
[15752.343531] Handle close event
[15848.863403] Exit RanNum driver
```