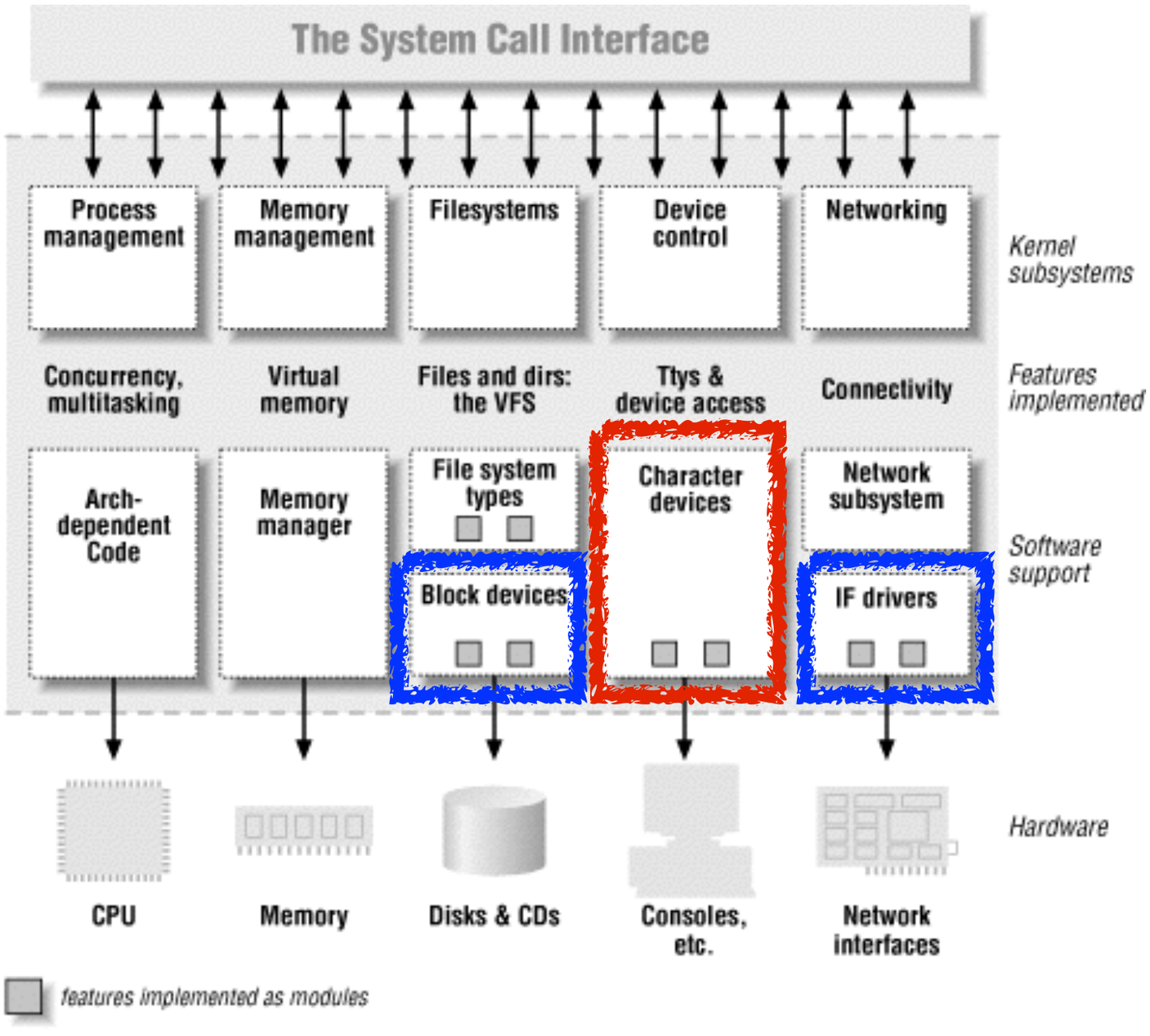
Char Device Driver
(I/O Control)

Speaker: Richard Chen
Team: Firmware

Foxlink Image Technology

Motivation



- Fundamental Device Driver
- Understanding **Register** and **Release** Device Driver Flow

Foxlink Image Technology

Device Driver

- Register Char Device Driver
- Release Char Device Driver

Device Driver

- **Register Char Device Driver**
- Release Char Device Driver

Register Char Device Driver

- Add Platform Device
- Add Platform Driver
- Register Char Device
- Load Module Steps
- App Test Result

Foxlink Image Technology

Register Char Device Driver

- **Add Platform Device**
- Add Platform Driver
- Register Char Device
- Load Module Steps
- App Test Result

Foxlink Image Technology

Add Platform Device

(3)

```
//=======Platform_Device_Structure=====================/
struct platform_device chardevice1 =
{
        .name= "char_dev",
        .id=-1,
};
static struct platform_device __initdata *chardevice[]=
{
&chardevice1,
};
//=======Initial Function========//
static int __init ioctltest_init(void)
{
        printk("Begin Driver Init Richard\n");
        printk("Start Platform Device Add! Richard\n");
//======Add Devices==============//
        platform_add_devices(chardevice,1);
//===============================================
        printk("Finished to add device Richard\n");
        printk("Start to register driver Richard\n");
//======Platform_Register_Device_Driver==========
        platform_driver_register(&chrdev_driver);
//===============================================
        printk("Driver Register Successful!\n");
        return 0;
}
```

(2)

(1)

- **struct platform_device** still can set another parameters.
- **Such_as:** .resource (for ioremap)

Foxlink Image Technology

Register Char Device Driver

- Add Platform Device
- **Add Platform Driver**
- Register Char Device
- Load Module Steps
- App Test Result

Foxlink Image Technology

Add Platform Driver

```
//=======Platform_Driver_Structure=====================//
static struct platform_driver chrdev_driver=
{
        .driver ={
         .name  = "char_dev",
         .owner = THIS_MODULE,
        },
        .probe  = char_probe,
};
//=======Initial Function========//
static int __init ioctltest_init(void)
{
        printk("Begin Driver Init_Richard\n");
        printk("Start Platform Device Add!_Richard\n");
//======Add Devices==============//
        platform_add_devices(chardevice,1);
//================================================
        printk("Finished to add device_Richard\n");
        printk("Start to register driver_Richard\n");
//======Platform_Register_Device_Driver==========
        platform_driver_register(&chrdev_driver);
//
        printk("Driver Register Successful!\n");
        return 0;
}
```

(2)

(1)

- **struct platform_driver** still can set another parameters.
- **Such_as:**
  .remove (for release)

Foxlink Image Technology

Register Char Device Driver

- Add Platform Device
- Add Platform Driver
- **Register Char Device**
- Load Module Steps
- App Test Result

Foxlink Image Technology

Register Char Device

```
//=======Platform_Driver_Structure=====================//
static struct platform_driver chrdev_driver=
{
        .driver ={
         .name  = "char_dev",
         .owner = THIS_MODULE,
        },
        .probe  = char_probe,
};
//======Char_Probe_Function============================//
static int char_probe(struct platform_device *pdev)
{
        int ret =0;
        printk("Start Char Probe\n");
        ret = register_chrdev(MAJORNUM,DEVICE_NAME,&fops);
                {
                        printk("Register Error!\n");
                }
                gpdev=pdev;
        return 0 ;
}
//=====File_Operations_Structure=======================//
static struct file_operations fops =
{
        .owner  = THIS_MODULE,
        .ioctl  = char_ioctl,
};
```

(1)

(2)

(3)

- Probe() Function:
  - Initial Function
  - Included Operation Functions
- **struct file_operations** still can set another parameters.
- **<u>Such_as:</u>**
  - .read (for read)

Foxlink Image Technology

Register Char Device Driver

- Add Platform Device
- Add Platform Driver
- Register Char Device
- **Load Module Steps**
- App Test Result

Foxlink Image Technology

Load Module Steps

- Make to build .ko (make)
- Insert Module (insmod char_dev.ko)
- List Module (lsmod l head)
- Check Device (cat /proc/devices)
- Make Node
  - (mknod /dev/char_device c 60 0)
- Debug (dmesg l tail)

Foxlink Image Technology

Register Char Device Driver

- Add Platform Device
- Add Platform Driver
- Register Char Device
- Load Module Steps
- **App Test Result**

Foxlink Image Technology

App Test Result

```
root@ubuntu: ~/Drivers
File  Edit  View  Terminal  Help
fd0u360          ram0                snd         tty32    tty63
root@ubuntu:~/Drivers# ls
a.out       chardev.ko     chardev.mod.o  ioctl_test.c  modules.order   opcode.h
chardev.c   chardev.mod.c  chardev.o      Makefile      Module.symvers
root@ubuntu:~/Drivers# chmod 777 /dev/char_device
root@ubuntu:~/Drivers# ls
a.out       chardev.ko     chardev.mod.o  ioctl_test.c  modules.order   opcode.h
chardev.c   chardev.mod                   Makefile      Module.symvers
root@ubuntu:~/Drivers# gcc ioctl_test.c
root@ubuntu:~/Drivers# ./a.out
IO Control Test
Please input command!
1)IOCTL_WRITE 2)EXIT
IOCTL_WRITE
Done. Wait 1 seconds...
IO Control Test
Please input command!
1)IOCTL_WRITE 2)EXIT
```

(App Name)

(I/O Command)

```
root@ubuntu: ~/Drivers
root@ubuntu:~/Drivers# dmesg | tail
[  314.461628] usb 1-2: USB disconnect, address 3
[ 8354.503288] e1000: eth0 NIC Link is Down
[ 8383.384536] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[12092.690293] Begin Driver Init_Richard
[12092.690297] Start Platform Device Add!_Richard
[12092.690330] Finished to add device_Richard
[12092.690332] Start to register driver_Richard
[12092.690811] Start Char_Probe
[12320.779143] IO Control
root@ubuntu:~/Drivers#
```

(Test Successful)

Foxlink Image Technology

Device Driver

- Register Char Device Driver
- **Release Char Device Driver**

Release Char Device Driver

- Unregister Platform Device
- Unregister Platform Driver
- Unregister Char Device
- Unload Module Steps

Foxlink Image Technology

Release Char Device Driver

- **Unregister Platform Device**
- Unregister Platform Driver
- Unregister Char Device
- Unload Module Steps

Foxlink Image Technology

Unregister Platform Device

```
//=======Release Driver Function=====//
static void ioctltest_exit(void)
{
        printk("Start Release Driver\n");
(1)     platform_device_unregister(gpdev);
        platform_driver_unregister(&chrdev_driver);
        unregister_chrdev(MAJORNUM,DEVICE_NAME);
        printk("Test finished,exit! Richard\n");
}
//=======Load and unload Module======//
module_init(ioctltest_init);    //==Load====Module======//
module_exit(ioctltest_exit);    //==Release=Module=======//

//=======Char_Probe_Function============================//
static int char_probe(struct platform_device *pdev)
{
        int ret =0;
        printk("Start Char_Probe\n");
        ret = register_chrdev(MAJORNUM,DEVICE_NAME,&fops);
        if (ret<0)
                {
                        printk("Register Error!\n");
(2)             }
                gpdev=pdev;
        return 0 ;
}
```

Foxlink Image Technology

Release Char Device Driver

- Unregister Platform Device
- **Unregister Platform Driver**
- Unregister Char Device
- Unload Module Steps

Foxlink Image Technology

Unregister Platform Driver

```
//=======Release Driver Function=====//
static void ioctltest_exit(void)
{
        printk("Start Release Driver\n");
        platform_device_unregister(gpdev);
        platform_driver_unregister(&chrdev_driver);
        unregister_chrdev(MAJORNUM,DEVICE_NAME);
        printk("Test finished,exit! _Richard\n");
}
//=======Load and unload Module======//
module_init(ioctltest_init);    //==Load====Module======//
module_exit(ioctltest_exit);    //==Release=Module=======//
//=======Platform Driver Structure=====================//
static struct platform_driver chrdev_driver=
{
        .driver ={
         .name  = "char_dev",
         .owner = THIS_MODULE,
        },
        .probe  = char_probe,
};
```

(1)

(2)

Foxlink Image Technology

Release Char Device Driver

- Unregister Platform Device
- Unregister Platform Driver
- **Unregister Char Device**
- Unload Module Steps

Foxlink Image Technology

Unregister Platform Device

```
//=======Release Driver Function=====//
static void ioctltest_exit(void)
{
        printk("Start Release Driver\n");
(I)     platform_device_unregister(gpdev);
        platform_driver_unregister(&chrdev_driver);
        unregister_chrdev(MAJORNUM,DEVICE_NAME);
        printk("Test Finished,Exit!_Richard\n");
}
//=======Load and unload Module======//
module_init(ioctltest_init);    //==Load====Module======//
module_exit(ioctltest_exit);    //==Release=Module=======//
```

Foxlink Image Technology

Release Char Device Driver

- Unregister Platform Device
- Unregister Platform Driver
- Unregister Char Device
- **Unload Module Steps**

Foxlink Image Technology

Unload Module Steps

- Remove Node (rm /dev/char_device)
- Release Module (rmmod char_dev)
- List Module (lsmod l head)
- Check Device (cat /proc/devices)
- Make Clean (Optional)
  - (make clean)
- Debug (dmesg l tail)

Foxlink Image Technology