



中山大學

SUN YAT-SEN UNIVERSITY

实 验 报 告

课程名称：_____操作系统_____

姓 名：_____孙广岩_____

学 号：_____20354242_____

专业班级：_____智能科学与技术专业 5 班_____

任课教师：_____吴贺俊_____

_____2022_____年_____11_____月_____11_____日

实验报告成绩评定表

| 评定项目 | 内 容 | 满 分 | 评 分 | 总 分 |
|-----------------------|---|--------|--------|--------|
| 实验态度 | 态度端正、遵守纪律、出勤情况 | 10 | | |
| 实验过程 | 按要求完成算法设计、代码书写、注释清晰、运行结果正确 | 30 | | |
| 实验记录 | 展示讲解清楚、任务解决良好、实验结果准确 | 20 | | |
| 报告撰写 | 报告书写规范、内容条理清楚、表达准确规范、上交及时、无抄袭，抄袭记 0 分，提供报告供抄袭者扣分。 | 40 | | |
| 评语： | | | | |
| | | | | |
| 指导老师签字：_____年 月 日 | | | | |

实验 8 I/O 系统-设备管理

一、 实验目的

1. 进一步加深对 I/O 系统设备管理方式及其实现过程的理解

二、 实验内容

1. 任务描述

- 计算机系统中使用各种各样的设备。大多数设备属于存储设备（磁盘、磁带）、传输设备（网络连接、蓝牙）和人机交互设备（屏幕、键盘、鼠标、音频输入输出）。
- 在 Linux 系统中，一切都可以看做文件，文档、目录、键盘、监视器、硬盘、可移动媒体设备、打印机、调制解调器、虚拟终端，还有进程间通信（IPC）和网络通信等输入/输出资源都是定义在文件系统空间下的字节流。
- Linux 根据设备共同特征将其划分为三大类型：字符设备；块设备；网络设备。
 - 字符设备是以字节为单位进行 I/O 传输，这种字符流的传输率通常比较低。常见字符设备有鼠标、键盘、触摸屏等；
 - 块设备是以块为单位传输的，常见块设备是磁盘；
 - 网络设备是一类比较特殊的设备，涉及到网络协议层，因此将其单独分成一类设备

2. 实验方案

编写字符设备驱动程序

- 该实验内容是编写字符设备驱动程序，要求能对该字符设备执行打开、读、写、I/O 控制和关闭五个基本操作。并通过应用程序，测试添加的字符设备的正确性。
- 该实验的操作需要管理员权限，因此需要切换到 root 权限或使用 sudo 命令。
- 具体的步骤如下：
 - (1)编写字符驱动源程序,即编写内核模块 chardev.c 文件和 Makefile 文件。使用 make 命令编译驱动模块。
 - (2)使用 insmod 命令安装驱动模块。
 - (3)创建字符设备文件，并查看创建的设备文件。
 - (4)编写一个测试程序 test.c，访问这个设备文件。用 gcc 编译这个文件，然后运行。
 - (5)使用 rmmod 卸载模块。
 - (6)使用 rm 命令删除创建的设备文件。

编写块设备驱动程序

- 这部分的实验内容是编写一个简单的块设备驱动程序，实现一套内存中的虚拟磁盘驱动器，并通过实际操作验证块设备驱动是否可以正常工作。
- 该实验的操作需要管理员权限，因此需要切换到 root 权限或使用 sudo 命令。
- 具体的步骤如下：

- (1) 编写设备驱动源程序，即编写内核模块 simp_blkdev.c 文件和 Makefile 文件。
- (2) 使用 make 命令编译驱动模块。
- (3) 使用 insmod 命令安装驱动模块。
- (4) 使用 lsblk 列出当前的块设备信息。
- (5) 格式化设备 simp_blkdev。
- (6) 创建挂载点并挂载该设备。
- (7) 查看模块使用情况，发现模块已被调用。
- (8) 对块设备驱动进行调用测试。
- (9) 取消挂载，查看模块调用结果。
- (10) 使用 rmmmod 卸载模块。

三、实验记录

1. 编写字符设备驱动程序

① 编写字符驱动源程序

使用原本的 Makefile 会报错，我修改了 Makefile 之后运行成功：

```
chardev > M Makefile
1  obj-m += chardev.o
2  cc=gcc
3  KDIR := /lib/modules/$(shell uname -r)/build
4  PWD := $(shell pwd)
5  all: modules
6  modules:
7  |   $(MAKE) -C $(KDIR) M=$(PWD) modules
8  clean:
9  |   $(MAKE) -C $(KDIR) M=$(PWD) clean

haiden@haiden-virtual-machine:~/os/chardev$ make
make -C /lib/modules/4.16.10/build M=/home/haiden/os/chardev modules
make[1]: Entering directory '/usr/src/linux-4.16.10'
CC [M] /home/haiden/os/chardev/chardev.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/haiden/os/chardev/chardev.mod.o
LD [M] /home/haiden/os/chardev/chardev.ko
make[1]: Leaving directory '/usr/src/linux-4.16.10'
```

② 使用 insmod 命令安装驱动模块

```
root@haiden-virtual-machine:~/os/chardev# insmod chardev.ko
root@haiden-virtual-machine:~/os/chardev# lsmod | grep chardev
chardev                16384  0
```

③ 创建字符设备文件，并查看创建的设备文件（主设备号为 240）

```
[ 1440.439902] <1> I was assigned major number 240
[ 1440.439903] <1> the drive, create a dev file
[ 1440.439903] <1> mknod /dev/hello c 240 0.
[ 1440.439904] <1> I was assigned major number 240
[ 1440.439904] <1> the device file
[ 1440.439904] <1> Remove the file device and module when done
root@haiden-virtual-machine:~/os/chardev# mknod /dev/hello c 240 0
root@haiden-virtual-machine:~/os/chardev#
```

④ 编写一个测试程序并运行

```
root@haiden-virtual-machine:~/os/chardev# vim test.c
root@haiden-virtual-machine:~/os/chardev# gcc -o a test.c
root@haiden-virtual-machine:~/os/chardev# ./a
I already told you 0 times Hello world

root@haiden-virtual-machine:~/os/chardev# ./a
I already told you 1 times Hello world

root@haiden-virtual-machine:~/os/chardev# ./a
I already told you 2 times Hello world

root@haiden-virtual-machine:~/os/chardev# ./a
I already told you 3 times Hello world

root@haiden-virtual-machine:~/os/chardev# ./a
I already told you 4 times Hello world

root@haiden-virtual-machine:~/os/chardev#
```

⑤ 使用 rmmod 卸载模块

```
root@haiden-virtual-machine:~/os/chardev# rmmod chardev
root@haiden-virtual-machine:~/os/chardev# lsmod | grep chardev
```

2. 编写块设备驱动程序

① 编写设备驱动源程序并编译

```
root@haiden-virtual-machine:~/os/simp_blkdev# make
make -C /lib/modules/4.16.10/build M=/home/haiden/os/simp_blkdev modules
make[1]: Entering directory '/usr/src/linux-4.16.10'
  CC [M] /home/haiden/os/simp_blkdev/simp_blkdev.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC /home/haiden/os/simp_blkdev/simp_blkdev.mod.o
  LD [M] /home/haiden/os/simp_blkdev/simp_blkdev.ko
make[1]: Leaving directory '/usr/src/linux-4.16.10'
```

② 使用 insmod 命令安装驱动模块

```
root@haiden-virtual-machine:~/os/simp_blkdev# insmod simp_blkdev.ko
root@haiden-virtual-machine:~/os/simp_blkdev# lsmod | grep simp_blkdev
simp_blkdev      52445184  0
```

③ 使用 lsblk 列出当前的块设备信息

```
root@haiden-virtual-machine:~/os/simp_blkdev# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
simp_blkdev 72:0    0   50M  0 disk
sr0         11:0    1 1024M  0 rom
sda          8:0     0   50G  0 disk
├─sda2       8:2     0    1K  0 part
├─sda5       8:5     0    2G  0 part
└─sda1       8:1     0  47.9G  0 part /
```

④ 格式化设备 simp_blkdev

```
root@haiden-virtual-machine:~/os/simp_blkdev# mkfs.ext3 /dev/simp_blkdev
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 51200 1k blocks and 12824 inodes
Filesystem UUID: 4c62edeb-253f-4eb9-b22d-af4855a27870
Superblock backups stored on blocks:
    8193, 24577, 40961

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```


- ⑤ 创建挂载点并挂载该设备

```
root@haiden-virtual-machine:/mnt# mount /dev/simp_blkdev /mnt/temp1/
root@haiden-virtual-machine:/mnt# mount | grep simp_blkdev
/dev/simp_blkdev on /mnt/temp1 type ext3 (rw,relatime,data=ordered)
```

- ⑥ 查看模块使用情况，发现模块已被调用

```
root@haiden-virtual-machine:/mnt# lsmod | grep simp_blkdev
simp_blkdev          52445184  1
```

- ⑦ 对块设备驱动进行调用测试

```
root@haiden-virtual-machine:/mnt# cp /etc/init.d/* /mnt/temp1/
```

```
root@haiden-virtual-machine:/mnt# ls /mnt/temp1/
acpid          irqbalance    README
alsa-utils     kerneloops    reboot
anacron        keyboard-setup  redis-server
apparmor       killprocs     resolvconf
appport        kmod          rsync
avahi-daemon   lightdm       rsyslog
binfmt-support lost+found     saned
bluetooth     mountall-bootclean.sh  sendsigs
bootmisc.sh   mountall.sh   single
brltty        mountdevsubfs.sh  skeleton
checkfs.sh    mountkernfs.sh  speech-dispatcher
checkroot-bootclean.sh  mountnfs-bootclean.sh  thermald
checkroot.sh  mountnfs.sh    udev
console-setup networking     ufw
cron          network-manager  umountfs
cups          ondemand        umountnfs.sh
cups-browsed  open-vm-tools   umountroot
dbus          plymouth        unattended-upgrades
dns-clean    plymouth-log    urandom
grub-common  pppd-dns        uuid
halt         procps          whoopsie
hddtemp      rc              x11-common
```

```
root@haiden-virtual-machine:/mnt# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            956M   0 956M   0% /dev
tmpfs           198M  6.3M 191M   4% /run
/dev/sda1       48G   27G 19G   59% /
tmpfs           986M  252K 985M   1% /dev/shm
tmpfs           5.0M   4.0K 5.0M   1% /run/lock
tmpfs           986M   0 986M   0% /sys/fs/cgroup
tmpfs           198M   72K 197M   1% /run/user/1000
/dev/simp_blkdev 45M   1.1M 41M   3% /mnt/temp1
root@haiden-virtual-machine:/mnt# rm -rf /mnt/temp1/*
root@haiden-virtual-machine:/mnt# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            956M   0 956M   0% /dev
tmpfs           198M  6.3M 191M   4% /run
/dev/sda1       48G   27G 19G   59% /
tmpfs           986M  252K 985M   1% /dev/shm
tmpfs           5.0M   4.0K 5.0M   1% /run/lock
tmpfs           986M   0 986M   0% /sys/fs/cgroup
tmpfs           198M   72K 197M   1% /run/user/1000
/dev/simp_blkdev 45M   817K 42M   2% /mnt/temp1
root@haiden-virtual-machine:/mnt#
```

- ⑧ 取消挂载，查看模块调用结果

```
root@haiden-virtual-machine:/mnt# umount /mnt/temp1/
root@haiden-virtual-machine:/mnt# lsmod | grep simp_blkdev
simp_blkdev          52445184  0
```

- ⑨ 使用 rmmod 卸载模块

```
root@haiden-virtual-machine:/mnt# rmmod simp_blkdev
root@haiden-virtual-machine:/mnt# lsmod | grep simp_blkdev
```