《计算机操作系统》实验报告

实验题目：内存分配和设备管理实验

姓名：严昕宇　　学号：20121802　　实验日期：2023.01.06

实验环境：

实验设备：Lenovo Thinkbook16+ 2022
操作系统：Ubuntu 22.04.1 LTS 64 位

实验目的：

了解 Linux 管理设备的基本方式

实验准备：

复习设备管理基本原理

实验内容：

1. 用 ls-l 命令观察设备文件的组织方式
2. 参照例程 12 编程，显示设备文件的设备号信息

操作过程 1：

用 ls -l 命令观察设备文件的组织方式

$ ls -l /dev

$ ls /dev | wc > data.out　　(将设备文件名通过管道送到 wc 命令计算设备文件名的行数，结果重定向传送到文件 data.out 中，计算设备文件个数。)

$ cat data.out

结果 1：

$ ls -l /dev

```
yanxinyu@ThinkBook16-2022:~$ ls -l /dev
总用量 0
crw-r--r--  1 root     root     10, 235 12月 22 13:40 autofs
drwxr-xr-x  2 root     root         460 12月 22 13:40 block
drwxr-xr-x  2 root     root         100 12月 22 13:40 bsg
crw-------  1 root     root     10, 234 12月 22 13:40 btrfs-control
drwxr-xr-x  3 root     root          60 12月 22 13:40 bus
lrwxrwxrwx  1 root     root           3 12月 22 13:40 cdrom -> sr0
drwxr-xr-x  2 root     root        3820 12月 22 15:47 char
crw--w----  1 root     tty        5,   1 12月 22 13:40 console
lrwxrwxrwx  1 root     root          11 12月 22 13:40 core -> /proc/kcore
drwxr-xr-x  4 root     root          80 12月 22 13:40 cpu
crw-------  1 root     root     10, 124 12月 22 13:40 cpu_dma_latency
crw-------  1 root     root     10, 203 12月 22 13:40 cuse
drwxr-xr-x  7 root     root         140 12月 22 13:40 disk
drwxr-xr-x  2 root     root          60 12月 22 13:40 dma_heap
crw-rw----+ 1 root     audio    14,   9 12月 22 13:40 dmmidi
drwxr-xr-x  3 root     root         100 12月 22 13:40 dri
crw-------  1 root     root     10, 126 12月 22 13:40 ecryptfs
```

```
crw-rw----  1 root     tty       7, 128 12月 22 13:40 vcsa
crw-rw----  1 root     tty       7, 129 12月 22 13:40 vcsa1
crw-rw----  1 root     tty       7, 130 12月 22 13:40 vcsa2
crw-rw----  1 root     tty       7, 131 12月 22 13:40 vcsa3
crw-rw----  1 root     tty       7, 132 12月 22 13:40 vcsa4
crw-rw----  1 root     tty       7, 133 12月 22 13:40 vcsa5
crw-rw----  1 root     tty       7, 134 12月 22 13:40 vcsa6
crw-rw----  1 root     tty       7,  64 12月 22 13:40 vcsu
crw-rw----  1 root     tty       7,  65 12月 22 13:40 vcsu1
crw-rw----  1 root     tty       7,  66 12月 22 13:40 vcsu2
crw-rw----  1 root     tty       7,  67 12月 22 13:40 vcsu3
crw-rw----  1 root     tty       7,  68 12月 22 13:40 vcsu4
crw-rw----  1 root     tty       7,  69 12月 22 13:40 vcsu5
crw-rw----  1 root     tty       7,  70 12月 22 13:40 vcsu6
drwxr-xr-x  2 root     root          60 12月 22 13:40 vfio
crw-------  1 root     root     10, 127 12月 22 13:40 vga_arbiter
crw-------  1 root     root     10, 137 12月 22 13:40 vhci
crw-rw----  1 root     kvm      10, 238 12月 22 13:40 vhost-net
crw-rw----  1 root     kvm      10, 241 12月 22 13:40 vhost-vsock
crw-------  1 root     root     10, 123 12月 22 13:40 vmci
crw-rw-rw-  1 root     root     10, 122 12月 22 13:40 vsock
crw-rw-rw-  1 root     root      1,   5 12月 22 13:40 zero
crw-------  1 root     root     10, 249 12月 22 13:40 zfs
yanxinyu@ThinkBook16-2022:~$
```

$ ls /dev | wc > data.out

```
yanxinyu@ThinkBook16-2022:~$ ls /dev | wc > data.out
yanxinyu@ThinkBook16-2022:~$
```

$ cat data.out

```
yanxinyu@ThinkBook16-2022:~$ cat data.out
    210     210    1278
yanxinyu@ThinkBook16-2022:~$
```

思考：Linux 管理设备的方法与管理文件的方式有何异同?为什么用管文件的方式来管设备?有什么好处?

答:

Linux 管理设备的方法与管理文件的方式有何异同?

- Linux 系统将所有的硬件设备都当作文件来处理，当使用光驱等硬件设备时，就必须将其挂载到系统中，只有这样 Linux 才能识别。也就是所谓的 Linux 系统“一切皆文件”，所有文件都放置在以根目录为树根的树形目录结构中。在 Linux 看来，任何硬件设备也都是文件。Linux 把所有外部设备统一当作成文件来看待，只要安装它们的驱动程序，任何用户都可以像使用文件一样，操纵、使用这些设备，而不必知道它们的具体存在形式。
- 但是，它们各有自己的一套文件系统(文件目录结构)。而且，当在 Linux 系统中使用这些硬件设备时，只有将 Linux 本身的文件目录与硬件设备的文件目录合二为一，硬件设备才能为我们所用。合二为一的过程称为“挂载”。如果不挂载，通过 Linux 系统中的图形界面系统可以查看找到硬件设备，但命令行方式无法找到。
- 而且，管理设备还需要设备控制器、设备与控制器之间的接口、缓冲管理等。

为什么用管文件的方式来管设备？有什么好处？

- 开发者仅需要使用一套 API 和开发工具即可调取 Linux 系统中绝大部分的资源，即大部分操作可以统一接口。例如，Linux 中几乎所有读(读文件、读系统状态、读 socket、读 pipe)的操作都可以用 read 函数来进行；几乎所有更改(更改文件、更改系统参数、写 socket、写 pipe)的操作都可以用 write 函数来进行。
- 所以，从本质上说“一切皆文件”，对于开发者来说更有益，代码的移植也更为方便。

思考：查阅资料了解 struct stat 结构体中包含的设备信息

答： stat 结构体用来表示相关文件状态信息。以 x86 为例，stat 结构体的定义如下：

```
struct stat {
    unsigned long  st_dev;
    unsigned long  st_ino;
    unsigned short st_mode;
    unsigned short st_nlink;
    unsigned short st_uid;
    unsigned short st_gid;
    unsigned long  st_rdev;
    unsigned long  st_size;
    unsigned long  st_blksize;
    unsigned long  st_blocks;
    unsigned long  st_atime;
    unsigned long  st_atime_nsec;
    unsigned long  st_mtime;
    unsigned long  st_mtime_nsec;
    unsigned long  st_ctime;
    unsigned long  st_ctime_nsec;
    unsigned long  __unused4;
    unsigned long  __unused5;
};
```

表 1 stat 结构体分析

变量	含义	变量	含义
st_dev	保存文件的设备	st_size	以字节为单位的文件容量
st_ino	与该文件关联的 inode	st_blksize	包含该文件的磁盘块的大小
st_mode	文件权限和文件类型信息	st_blocks	该文件所占的磁盘块数
st_nlink	该文件上硬连接的个数	st_atime	文件上一次被访问的时间
st_uid	文件属主的 UID 号	st_ctime	文件的权限、属主、组或内容上一次被修改的时间
st_gid	文件属主的 GID 号	st_mtime	文件的内容上一次被修改的时间
st_rdev	设备文件的设备号		

操作过程 2：

参照例程 12 编程，显示设备文件的设备号信息

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/sysmacros.h>


int main(int argc, char *argv[]) {
    int i;
    struct stat buf;
    for (i = 1; i < argc; i++) {
        printf("%s: ", argv[i]);
        if (stat(argv[i], &buf) < 0) {
            perror("stat error\n");
            continue;
        }
        //st_dev 为文件系统的设备号，st_rdev 为实际设备的设备号（字符特殊文件和块特殊文件才有）
        //设备号分为主、次设备号，可以分别使用宏 major 与 minor 获取
        printf("dev = %d/%d ", major(buf.st_dev), minor(buf.st_dev));

        if (S_ISCHR(buf.st_mode) || S_ISBLK(buf.st_mode)) {
            printf(" (%s) rdev = %d/%d",(S_ISCHR(buf.st_mode)) ? "character" : "block" ,major(buf.st_rdev),minor(buf.st_rdev));
        }
        printf("\n");
    }
    exit(0);
}
```

结果 2：

先查看硬盘设备的设备号，再执行程序,可以发现输入内容确实是硬盘的设备号。

```
yanxinyu@ThinkBook16-2022:~$ df -h
文件系统        容量  已用  可用 已用% 挂载点
tmpfs           389M  2.1M  387M    1% /run
/dev/sda3        20G   12G  6.7G   64% /
tmpfs           1.9G     0  1.9G    0% /dev/shm
tmpfs           5.0M  4.0K  5.0M    1% /run/lock
/dev/sda2       512M  5.3M  507M    2% /boot/efi
tmpfs           389M  2.4M  387M    1% /run/user/1000
yanxinyu@ThinkBook16-2022:~$ ls -l /dev/sda3
brw-rw---- 1 root disk 8, 3 12月 22 13:40 /dev/sda3
yanxinyu@ThinkBook16-2022:~$ ./prog7.o /dev/sda3
/dev/sda3:dev = 0/5  (block) rdev = 8/3
```

再查看其他的设备文件，此时后面为字符设备的设备号，且提示为 character。

```
yanxinyu@ThinkBook16-2022:~$ ./prog7.o /dev/ttyS0
/dev/ttyS0:dev = 0/5  (character) rdev = 4/64
yanxinyu@ThinkBook16-2022:~$ ./prog7.o /dev/zero
/dev/zero:dev = 0/5  (character) rdev = 1/5
```

讨论

Linux 管理设备的特点

答：

- Linux 设备管理的主要任务是控制设备完成输入输出操作，所以又称输入输出(I/O)子系统。它的任务是把各种设备硬件的复杂物理特性的细节屏蔽起来，提供一个对各种不同设备使用统一方式进行操作的接口。
- Linux 可以说是 Unix 操作系统的一个克隆，Linux 沿用了类似于 Unix 的系统架构。Unix 系统将设备看作是文件系统的节点。设备以特殊文件节点的方式呈现在目录中，该目录通常包含设备文件系统的节点入口。用文件系统节点来表示设备的目的是使得应用程序能以设备无关的方式访问各种设备。应用程序仍然可以通过 I/O 控制操作进行特定于设备的操作。设备由主设备号和次社保号进行标识。主设备号用来作为驱动数组的索引下标，而次设备号将相似的物理设备归组。
- Unix 有两种类型的设备，即字符设备和块设备。字符设备驱动管理没有缓冲并需要顺序访问的设备，块设备驱动管理那些可随机访问的设备，数据以块的方式被访问。另外，块设备驱动还用到缓冲区。块设备必须以文件系统节点的方式挂载之后才能被访问。
- Linux 保留了 Unix 的很多架构设计，区别在于，Unix 系统中，每个块设备需要创建一个对应的字符设备，而在 Linux 中，虚拟文件系统(VFS)接口使得字符设备和块设备的区分变得模糊。
- Linux 还引入了第三种设备，叫网络设备。访问网络设备驱动的方式和访问字符设备、块设备的方式不同，它使用了不同于文件系统 I/O 接口的一个接口集。比如 socket 接口，就是用来访问网络设备的。

实验体会

通过本次实验，我进一步理解陈老师在理论课上所提及的，“Linux 哲学核心思想是一切皆文件”背后的含义。

“一切皆文件”，指的是 Linux/Unix 对所有文件(目录、字符设备、块设备、套接字、打印机、进程、线程、管道等)操作，读写都可用 fopen()/fclose()/fwrite()/fread()等函数进行处理。屏蔽了硬件的区别，所有设备都抽象成文件，提供统一的接口给用户。通过实验，我发现虽然其类型各不相同，但是对其提供的却是同一套操作界面。更进一步，对文件的操作也可以跨文件系统执行。这使得开发者仅需要使用一套 API 和开发工具，即可调取 Linux 系统中绝大部分的资源。