

# 实验七：内存分配和设备管理实验

姓名：汪江豪 学号：22121630 实验日期：2025.01.06

---

## 实验环境：

实验设备：Lenovo Legion R7000P 2021H

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：

- 用 ls-l 命令观察设备文件的组织方式

```
$ls -l /dev  
$ls /dev | wc > data.out (将设备文件通过管道送到 wc 命令计算设备文件名的行数，结果重定向到文件 data.out 中，计算设备文件个数。)  
$cat data.out (显示结果)
```

- 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 (将设备文件通过管道送到 wc 命令计算设备文件名的行数，结果重定向到文件 data.out 中，计算设备文件个数。)  
$cat data.out (显示结果)
```

## 结果 1：

```
$ls -l /dev
```

```
pioneer@pioneer-virtual-machine:~$ ls -l /dev  
total 0  
crw-r--r-- 1 root root 10, 235 1月 6 13:18 autofs  
drwxr-xr-x 2 root root 460 1月 6 13:18 block  
drwxr-xr-x 2 root root 100 1月 6 13:18 bsg  
crw----- 1 root root 10, 234 1月 6 13:18 btrfs-control  
drwxr-xr-x 3 root root 60 1月 6 13:18 bus  
lrwxrwxrwx 1 root root 3 1月 6 13:18 cdrom -> sr0  
drwxr-xr-x 2 root root 3840 1月 6 13:21 char  
crw--w---- 1 root tty 5, 1 1月 6 13:18 console  
lrwxrwxrwx 1 root root 11 1月 6 13:18 core -> /proc/kcore  
drwxr-xr-x 6 root root 120 1月 6 13:18 cpu  
crw----- 1 root root 10, 123 1月 6 13:18 cpu_dma_latency
```

```

pioneer@pioneer-virtual-machine:~$ ls /dev | wc > data.out
pioneer@pioneer-virtual-machine:~$ ls
data.out ps.log snap vim 下载 图片 桌面 视频
os2 sdb st.sh who.log 公共的 文档 模板 音乐
pioneer@pioneer-virtual-machine:~$ cat data.out
211 211 1292

```

**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:

相同点: 统一接口。设备和文件都通过文件系统路径来访问, 可以使用相同命令进行操作。权限控制。设备文件像普通文件一样拥有权限设置, 可以控制用户和组对设备的访问。标准操作。可以像操作文件一样对设备进行打开、读取、写入和关闭操作。

不同点: 本质区别, 文件是存储在磁盘上的数据集合, 而设备文件是硬件设备的抽象表示。设备驱动, 设备文件通常与特定的设备驱动程序关联, 这些驱动程序负责与硬件通信。特殊操作, 某些设备文件支持特殊的文件操作, 允许程序向设备发送控制命令。

为什么使用管理文件的方式来管理设备: 简化用户界面, 通过将设备抽象为文件, 用户和程序可以使用统一的文件操作接口与设备交互, 无需学习额外的命令或 API。易于脚本化, 设备管理类似于文件操作, 可以容易地通过脚本来自动化设备管理任务。提高了可移植性, 且这种模块化设计, 可以动态地加载和卸载。

好处是将硬件设备抽象为文件, 提供了统一的接口访问不同的设备, 简化了编程模型。应用了文件系统的权限模型来控制对设备的访问, 增强了系统的安全性。且便于扩展。

## 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;
        }
    }
}

```

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

## 结果 2：

```
pioneer@pioneer-virtual-machine:~/os2/test7$ ls -l /dev/vcs\ncrw-rw---- 1 root tty 7, 0 1月 6 13:18 /dev/vcs\npioneer@pioneer-virtual-machine:~/os2/test7$ ./test7 /dev/vcs\n/dev/vcs: dev = 0/5 (character) rdev = 7/0\n\npioneer@pioneer-virtual-machine:~/os2/test7$ ls -l /dev/sda2\nbrw-rw---- 1 root disk 8, 2 1月 6 13:18 /dev/sda2\npioneer@pioneer-virtual-machine:~/os2/test7$ ./test7 /dev/sda2\n/dev/sda2: dev = 0/5 (block) rdev = 8/2
```

程序正确显示了设备号

## 讨论：

Linux 管理设备的特点：

答：

首先是设备文件系统。Linux 将设备表示为文件，统一通过文件系统接口进行访问，这些设备文件通常位于/dev 目录下，便于统一管理和操作。

设备分为字符设备和块设备，字符设备一次读写一个字符，如串口，终端；块设备一次读写一块数据，如硬盘、U 盘。

Linux 通过 udev 守护进程，Linux 能够动态地添加和移除设备节点，管理设备的命名和权限。

Linux 支持模块化管理设备，能够对设备的驱动程序动态加载和卸载，提高了系统的灵活性和效率。

Linux 还为设备文件添加了权限管理功能，可通过 chmod, chown 等命令设置读写权限，确保设备的安全访问。

## 心得体会：

本次实验，我深刻体会到了 Linux 操作系统与硬件设备之间紧密的交互关系，每个设备都被抽象为一个文件，极大地简化了硬件的访问和管理。同时，还与普通文件一样，为设备文件添加了权限管理措施，提高了系统的安全性。本次 Linux 设备管理实验，不仅增强了我对于 Linux 操作系统底层对于设备的管理机制，也提升了我在实际环境中管理硬件设备的能力。这是一次理论与实践结合的学习体验。