



中山大學

SUN YAT-SEN UNIVERSITY

实 验 报 告

课程名称：____操作系统____

姓 名：____方桂安____

学 号：____20354027____

专业班级：____2020 级智能科学与技术____

任课教师：____吴贺俊____

____2022____年____12____月____23____日

实验九 文件系统

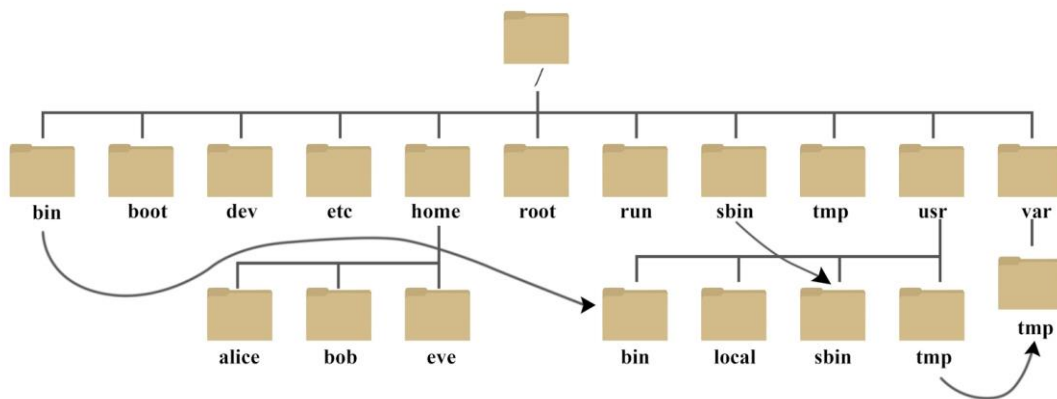
一、 实验目的

- 1、 熟悉 Linux 文件系统的文件和目录结构,掌握 Linux 文件系统的基本特征;
- 2、 模拟实现 Linux 文件系统的简单 I/O 流操作: 备份文件。

二、 实验内容

- 1、 浏览 Linux 系统根目录下的子目录,熟悉每个目录的文件和功能;
- 2、 设计程序模拟实现 Linux 文件系统的简单 I/O 流操作: 备份文件。

三、 实验原理



以下是对这些目录的解释:

- **/bin:**

bin 是 Binaries (二进制文件) 的缩写, 这个目录存放着最经常使用的命令。

- **/boot:**

这里存放的是启动 Linux 时使用的一些核心文件, 包括一些连接文件以

及镜像文件。

- **/dev :**

dev 是 Device(设备) 的缩写, 该目录下存放的是 Linux 的外部设备, 在 Linux 中访问设备的方式和访问文件的方式是相同的。

- **/etc:**

etc 是 Etcetera(等等) 的缩写, 这个目录用来存放所有的系统管理所需要的配置文件和子目录。

- **/home:**

用户的主目录, 在 Linux 中, 每个用户都有一个自己的目录, 一般该目录名是以用户的账号命名的, 如上图中的 alice、bob 和 eve。

- **/lib:**

lib 是 Library(库) 的缩写这个目录里存放着系统最基本的动态连接共享库, 其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。

- **/lost+found:**

这个目录一般情况下是空的, 当系统非法关机后, 这里就存放了一些文件。

- **/media:**

linux 系统会自动识别一些设备, 例如 U 盘、光驱等等, 当识别后, Linux 会把识别的设备挂载到这个目录下。

- **/mnt:**

系统提供该目录是为了让用户临时挂载别的文件系统的, 我们可以将光驱挂载在 /mnt/ 上, 然后进入该目录就可以查看光驱里的内容了。

- **/opt:**

opt 是 optional(可选) 的缩写,这是给主机额外安装软件所摆放的目录。

比如你安装一个 ORACLE 数据库则就可以放到这个目录下。默认是空的。

- **/proc:**

proc 是 Processes(进程) 的缩写, /proc 是一种伪文件系统(也即虚拟文件系统), 存储的是当前内核运行状态的一系列特殊文件, 这个目录是一个虚拟的目录, 它是系统内存的映射, 我们可以通过直接访问这个目录来获取系统信息。

这个目录的内容不在硬盘上而是在内存里, 我们也可以直接修改里面的某些文件, 比如可以通过下面的命令来屏蔽主机的 ping 命令, 使别人无法 ping 你的机器:

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

- **/root:**

该目录为系统管理员, 也称作超级权限者的用户主目录。

- **/sbin:**

s 就是 Super User 的意思, 是 Superuser Binaries (超级用户的二进制文件) 的缩写, 这里存放的是系统管理员使用的系统管理程序。

- **/selinux:**

这个目录是 Redhat/CentOS 所特有的目录, Selinux 是一个安全机制, 类似于 windows 的防火墙, 但是这套机制比较复杂, 这个目录就是存放 selinux 相关的文件的。

- **/srv:**

该目录存放一些服务启动之后需要提取的数据。

- **/sys:**

这是 Linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 `sysfs` 。

`sysfs` 文件系统集成了下面 3 种文件系统的信息：针对进程信息的 `proc` 文件系统、针对设备的 `devfs` 文件系统以及针对伪终端的 `devpts` 文件系统。

该文件系统是内核设备树的一个直观反映。

当一个内核对象被创建的时候，对应的文件和目录也在内核对象子系统中被创建。

- **/tmp:**

`tmp` 是 `temporary` (临时) 的缩写这个目录是用来存放一些临时文件的。

- **/usr:**

`usr` 是 `unix shared resources` (共享资源) 的缩写，这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似于 windows 下的 `program files` 目录。

- **/usr/bin:**

系统用户使用的应用程序。

- **/usr/sbin:**

超级用户使用的比较高级的管理程序和系统守护程序。

- **/usr/src:**

内核源代码默认的放置目录。

- /var:

var 是 variable(变量) 的缩写, 这个目录中存放着在不断扩充着的东西, 我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。

- /run:

是一个临时文件系统, 存储系统启动以来的信息。当系统重启时, 这个目录下的文件应该被删掉或清除。如果你的系统上有 /var/run 目录, 应该让它指向 run。

四、 实验记录

1. 实施步骤

TODO 1 补充磁盘申请程序

```
p->text = new char[length];
q = new diskNode;
q->start = p->start + length;
q->maxlength = 1000;
q->useFlag = 0;
q->next = NULL;
```

TODO 2 补充读取磁盘内容代码, 并将内容输出

```
for (need = diskHead; need != NULL; need = need->next)
{
    if (need->start == p->file->start)
        break;
}
if (need)
{
    //找到磁块, 输出磁块存储内容
    cout << need->text << endl;
}
```

TODO 3 补充释放 UFD 代码

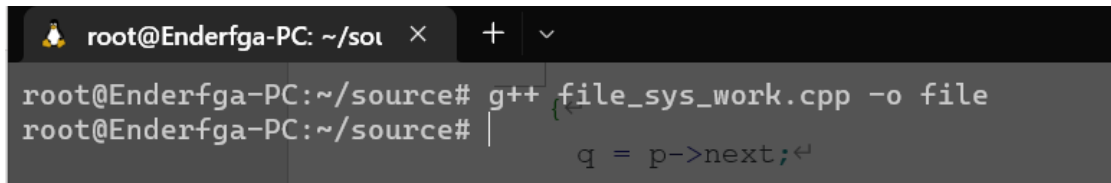
```
p = user[i].user;
while(p)
```

```

{
    q = p->next;
    delete(p);
    p = q;
}
user[i].user = NULL;

```

2. 实验记录

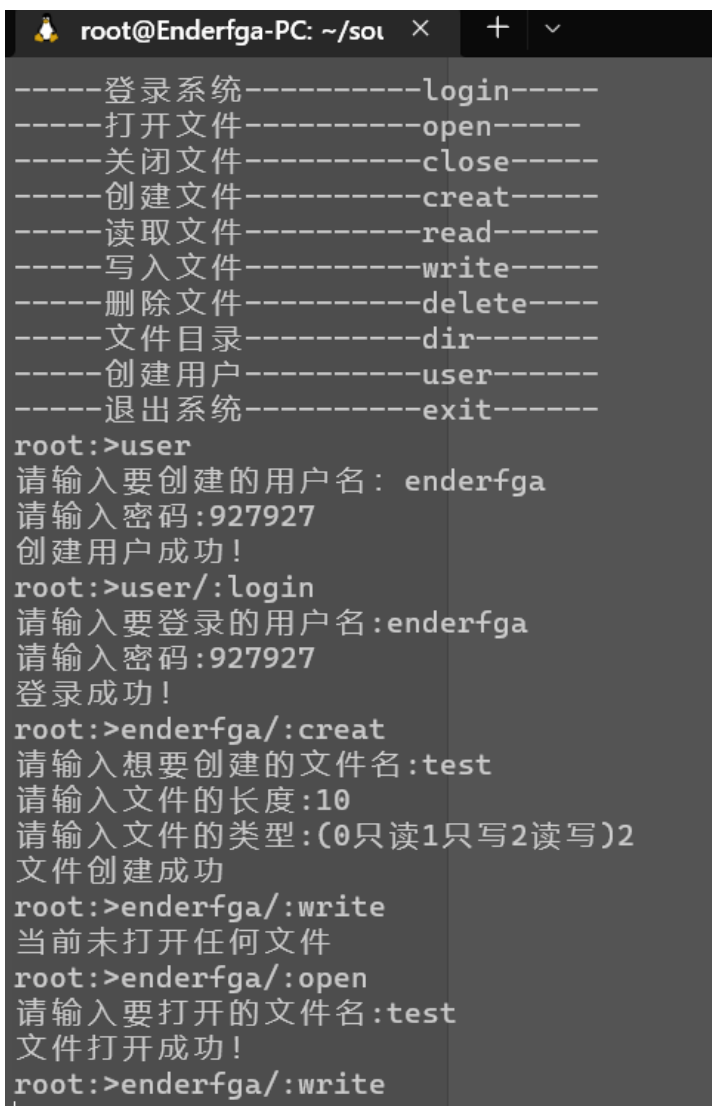


```

root@Enderfga-PC: ~/source # g++ file_sys_work.cpp -o file
root@Enderfga-PC: ~/source #

```

3. 实验结果



```

root@Enderfga-PC: ~/source #
-----登录系统-----login-----
-----打开文件-----open-----
-----关闭文件-----close-----
-----创建文件-----creat-----
-----读取文件-----read-----
-----写入文件-----write-----
-----删除文件-----delete-----
-----文件目录-----dir-----
-----创建用户-----user-----
-----退出系统-----exit-----
root:>user
请输入要创建的用户名:enderfga
请输入密码:927927
创建用户成功!
root:>user/:login
请输入要登录的用户名:enderfga
请输入密码:927927
登录成功!
root:>enderfga/:creat
请输入想要创建的文件名:test
请输入文件的长度:10
请输入文件的类型:(0只读1只写2读写)2
文件创建成功
root:>enderfga/:write
当前未打开任何文件
root:>enderfga/:open
请输入要打开的文件名:test
文件打开成功!
root:>enderfga/:write

```


五、 总结与讨论

在 Linux 系统中，有几个目录是比较重要的，需要注意不要误删除或者随意更改内部文件。

/etc: 上边也提到了，这个是系统中的配置文件，如果更改了该目录下的某个文件可能会导致系统不能启动。

/bin, /sbin, /usr/bin, /usr/sbin: 这是系统预设的执行文件的放置目录，比如 `ls` 就是在 `/bin/ls` 目录下的。

值得提出的是 `/bin`、`/usr/bin` 是给系统用户使用的指令（除 `root` 外的通用用户），而 `/sbin`、`/usr/sbin` 则是给 `root` 使用的指令。

/var: 这是一个非常重要的目录，系统上跑了很多程序，那么每个程序都会有相应的日志产生，而这些日志就被记录到这个目录下，具体在 `/var/log` 目录下，另外 `mail` 的预设放置也是在这里。