

---

附件（四）

# 深圳大学实验报告

课程名称： 操作系统

实验项目名称： 文件系统实验

学院： 数学科学学院

专业： 信息与计算科学（数学与计算机实验班）

指导教师： 张 滇

报告人： 王曦 学号： 2021192010 班级： 数计

实验时间： 2024 年 05 月 12 日

实验报告提交时间： 2024 年 05 月 12 日

教务处制

## 一、实验目的与要求

### 1. 实验目的

- (1) 了解 Linux 文件命令行操作命令。
- (2) 了解 Linux ext3 文件系统上的软硬链接。

### 2. 实验内容

- (1) 可以使用 Linux 或其它 Unix 类操作系统。
- (2) 学习该操作系统提供的文件系统的命令行接口。
- (3) 学习文件的软硬链接的使用。

### 3. 实验环境

- (1) 硬件：桌面 PC
- (2) 软件：Linux 或其他操作系统

## 二、方法、步骤

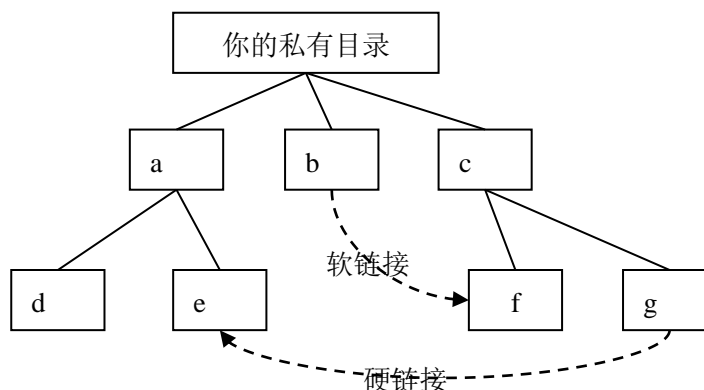
(说明程序相关的算法原理或知识内容，程序设计的思路和方法，可以用流程图表述，程序主要数据结构的设计、主要函数之间的调用关系等)

操作部分（参考）：

(1) 学习使用 Linux 文件系统提供的 ls、touch、rm、cp、mv、mkdir、df 等命令（希望尽量涵盖各种满足日常编程所需操作），记录相关命令执行结果。

(2) 学习 Linux 文件系统中关于文件硬链接和软链接的概念和相关操作命令，创建软硬链接各一个，给出实验证据表明它们分别是软硬连接。

(3) 构建以下目录子树：



## 三、实验过程及内容

(对程序代码进行说明和分析，越详细越好，代码排版要整齐，可读性要高)

### 1. Linux 的文件命令

学习使用 Linux 文件系统提供的 ls、touch、rm、cp、mv、mkdir、df 等命令（希望

尽量涵盖各种满足日常编程所需操作)，记录相关命令执行结果。

## 1.1 ls 命令

功能	列出当前目录中的文件或子目录。	
语法	ls [选项] [文件或目录]	
选项	-l	以长格式（详细信息）列出文件和目录。显示文件的权限、所有者、大小、修改日期等信息。
	-a	显示所有文件和目录，包括以.开头的隐藏文件。
	-h	以人类可读的格式显示文件和目录的大小，例如，使用 K、M、G 等单位。
	-t	按修改时间排序文件和目录，最近修改的文件先显示。
	-r	以相反的顺序显示文件和目录。
	-R	递归地列出子目录中的文件和目录。
	-S	按文件大小排序文件和目录。
	-i	显示文件和目录的 inode 号码。
	-ld	查看目录属性。
	-lh	查看目录（大小以字节、MB 显示）

当前目录如下图所示。

/home/hytidel/OSexp/coexp1						
历史 ↺ ↻ ↕ ↕ ↕						
OSexp	文件名 ^	大小	类型	修改时间	权限	用户/用户组
coexp1	os-exp-fifo		文件夹	2024/04/25 15:24		root/root
os-exp-fifo	fork-100-demo	15.7 KB	文件	2024/04/25 14:38	-rwxrwxr-x	hytidel/hytid...
exp1	fork-100-demo.c	403 B	DevCpp.c	2024/04/25 14:37	-rw-rw-r--	hytidel/hytid...
exp2	fork-demo	15.7 KB	文件	2024/04/25 10:14	-rwxrwxr-x	hytidel/hytid...
exp3	fork-demo.c	407 B	DevCpp.c	2024/04/25 10:14	-rw-rw-r--	hytidel/hytid...
Pictures	fork-exec-demo	15.8 KB	文件	2024/04/25 14:10	-rwxrwxr-x	hytidel/hytid...
	fork-exec-demo.c	591 B	DevCpp.c	2024/04/25 14:12	-rw-rw-r--	hytidel/hytid...

### 1.1.1 无选项

用如下命令列出当前目录下的文件。

```
ls
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ ls
fork-100-demo    HelloWorld.cpp      os-exp-fifo        psem-named-wait-demo  shmatt-write-demo.c
fork-100-demo.c  HelloWorld-getchar  pipe-demo.c        psem-named-wait-demo.c  shmatt-write-demo.c
fork-demo        HelloWorld-getchar.cpp  pipe-demo.c        pthread-100-demo      shmget-demo.c
fork-demo.c      msgtool            psem-named-open-demo  pthread-100-demo.c    zombie-demo
fork-exec-demo   msgtool.c          psem-named-open-demo.c  pthread-demo.c        zombie-demo.c
fork-exec-demo.c  mutex-demo         psem-named-post-demo  shell-script           shmatt-read-demo
fork-twice-demo  no-mutex-demo       psem-named-post-demo.c  shmatt-read-demo.c
fork-twice-demo.c no-mutex-demo.c     psem-named-unlink-demo  shmatt-read-demo.c
HelloWorld       no-mutex-demo.c     psem-named-unlink-demo.c
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

### 1.1.2 -l 属性

用如下命令列出当前目录下的文件，并显示文件权限、所有者、大小、修改日期等信息。

```
ls -l
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ ls -l
total 420
-rwxrwxr-x 1 hytidel hytidel 16096 4月 25 14:38 fork-100-demo
-rw-rw-r-- 1 hytidel hytidel 403 4月 25 14:37 fork-100-demo.c
-rwxrwxr-x 1 hytidel hytidel 16096 4月 25 10:14 fork-demo
-rw-rw-r-- 1 hytidel hytidel 407 4月 25 10:14 fork-demo.c
-rwxrwxr-x 1 hytidel hytidel 16192 4月 25 14:10 fork-exec-demo
-rw-rw-r-- 1 hytidel hytidel 591 4月 25 14:12 fork-exec-demo.c
-rwxrwxr-x 1 hytidel hytidel 16096 4月 25 11:05 fork-twice-demo
-rw-rw-r-- 1 hytidel hytidel 429 4月 25 11:04 fork-twice-demo.c
-rwxrwxr-x 1 hytidel hytidel 17648 4月 24 20:44 HelloWorld
-rw-rw-r-- 1 hytidel hytidel 105 4月 24 20:42 HelloWorld.cpp
-rwxrwxr-x 1 hytidel hytidel 17696 4月 24 20:59 HelloWorld-getchar
-rw-rw-r-- 1 hytidel hytidel 116 4月 24 20:58 HelloWorld-getchar.cpp
-rwxrwxr-x 1 hytidel hytidel 16840 4月 25 15:43 msgtool
-rw-rw-r-- 1 hytidel hytidel 2617 4月 25 15:42 msgtool.c
```

### 1.1.3 -a 属性

用如下命令列出当前目录下的文件和目录，包括以.开头的隐藏文件。

```
ls -a
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ ls -a
.                fork-twice-demo  mutex-demo      psem-named-open-demo.c  pthread-100-demo.c  shmget-demo
..               fork-twice-demo.c  mutex-demo.c    psem-named-post-demo    pthread-demo         shmget-demo.c
.               HelloWorld    no-mutex-demo.c  psem-named-post-demo.c  pthread-demo.c      zombie-demo
fork-100-demo.c HelloWorld.cpp    no-mutex-demo.c  psem-named-unlink-demo  shell-script        zombie-demo.c
fork-demo       HelloWorld-getchar  os-exp-fifo      psem-named-unlink-demo.c  shmatt-read-demo    shmatt-read-demo.c
fork-demo.c     HelloWorld-getchar.cpp  pipe-demo        psem-named-unlink-demo.c  shmatt-read-demo.c  shmatt-read-demo.c
fork-exec-demo  msgtool           pipe-demo.c      psem-named-unlink-demo.c  shmatt-write-demo    shmatt-write-demo.c
fork-exec-demo.c msgtool.c         pipe-demo.c      psem-named-unlink-demo.c  shmatt-write-demo.c  shmatt-write-demo.c
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$
```

### 1.1.4 -r 属性

用如下命令列出当前目录下的文件和目录，包括以.开头的隐藏文件。

```
ls -r
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ ls -r
zombie-demo.c      pthread-demo.c      psem-named-post-demo  mutex-demo          fork-exec-demo.c
zombie-demo        pthread-demo        psem-named-open-demo.c msgtool.c           fork-exec-demo
shmget-demo.c      pthread-100-demo.c  psem-named-open-demo  HelloWorld-getchar.cpp fork-demo.c
shmget-demo        pthread-100-demo    pipe-demo.c           HelloWorld-getchar  fork-100-demo.c
shmatt-write-demo.c psem-named-unlink-demo.c  os-exp-fifo          HelloWorld          fork-100-demo
shmatt-write-demo.c psem-named-unlink-demo.c  no-mutex-demo.c     HelloWorld          fork-twice-demo.c
shmatt-read-demo.c psem-named-unlink-demo.c  mutex-demo.c         fork-twice-demo     fork-twice-demo
shell-script        psem-named-post-demo.c  msgtool.c           fork-twice-demo     fork-twice-demo
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$
```

### 1.1.5 -R 属性

用如下命令列出当前目录下的文件和目录，包括以.开头的隐藏文件。

```
ls -R
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ ls -R
.:
fork-100-demo      HelloWorld.cpp      os-exp-fifo        psem-named-wait-demo  shmatt-write-demo
fork-100-demo.c    HelloWorld-getchar pipe-demo           psem-named-wait-demo.c shmatt-write-demo.c
fork-demo          HelloWorld-getchar.cpp pipe-demo.c        pthread-100-demo      shmget-demo
fork-demo.c        msgtool.c          psem-named-open-demo pthread-100-demo.c    shmget-demo.c
fork-exec-demo     msgtool.c          psem-named-open-demo.c pthread-demo           zombie-demo
fork-exec-demo.c   mutex-demo         psem-named-post-demo pthread-demo.c         zombie-demo.c
fork-twice-demo    mutex-demo.c        psem-named-post-demo.c shell-script          shmatt-read-demo
fork-twice-demo.c  no-mutex-demo      psem-named-unlink-demo shmatt-read-demo.c
HelloWorld         no-mutex-demo.c    psem-named-unlink-demo.c
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

1.1.6 -S 属性

用如下命令列出当前目录下的文件和目录，包括以.开头的隐藏文件。

ls -S

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ ls -S
HelloWorld-getchar  pthread-100-demo  psem-named-open-demo  psem-named-wait-demo.c  psem-named-open-demo.c
HelloWorld          pthread-demo      psem-named-unlink-demo  pthread-100-demo.c      psem-named-unlink-demo.c
msgtool             fork-exec-demo    msgtool.c              shmget-demo.c           HelloWorld-getchar.cpp
mutex-demo          psem-named-post-demo  mutex-demo.c          fork-twice-demo.c       HelloWorld.cpp
no-mutex-demo       shmget-demo        no-mutex-demo.c        zombie-demo.c           shell-script
pipe-demo           zombie-demo        pipe-demo.c            fork-demo.c             os-exp-fifo
shmatt-read-demo    fork-100-demo     shmatt-write-demo.c    fork-100-demo.c        psem-named-post-demo.c
shmatt-write-demo   fork-demo         shmatt-read-demo.c     psem-named-post-demo.c  pthread-demo.c
psem-named-wait-demo fork-twice-demo    fork-exec-demo.c       pthread-demo.c
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

1.2 touch 命令

功能	创建空文件或修改文件的时间戳。	
语法	touch [选项] [文件名]	
选项	-a	修改文件的访问时间。
	-c	修改文件的状态修改时间。
	-m	修改文件的数据修改时间。
	-d 时间	指定文件的修改时间。

1.2.1 无选项

用如下命令创建文件 test.txt。

touch test.txt

运行结果如下图所示。当前目录下出现文件 test.txt。

/home/hytidel/OSexp/coexp1

文件名	大小	类型	修改时间	权限	用户/用户组
shmatt-write-demo	15.9 KB	文件	2024/04/25 16:32	-rwxrwxr-x	hytidel/hytid...
shmatt-write-demo.c	787 B	DevCpp.c	2024/04/25 16:18	-rw-rw-r--	hytidel/hytid...
shmget-demo	15.8 KB	文件	2024/04/25 16:00	-rwxrwxr-x	hytidel/hytid...
shmget-demo.c	452 B	DevCpp.c	2024/04/25 15:57	-rw-rw-r--	hytidel/hytid...
test.txt	0	文本文档	2024/05/10 16:09	-rw-rw-r--	hytidel/hytid...
zombie-demo	15.8 KB	文件	2024/04/25 11:20	-rwxrwxr-x	hytidel/hytid...
zombie-demo.c	429 B	DevCpp.c	2024/04/25 11:20	-rw-rw-r--	hytidel/hytid...

用如下命令查询 test.txt 的状态修改时间。

ls -l | grep test.txt

运行结果如下图所示。test.txt 的状态修改时间为 16:09。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ ls -l | grep test.txt
-rw-rw-r-- 1 hytidel hytidel 0 5月 10 16:09 test.txt
```

1.2.2 -c 选项

用如下命令修改文件 test.txt 的状态修改时间为当前时间，并查询其状态修改时间。



```
touch test.txt -c
ls -l | grep test.txt
```

运行结果如下图所示。test.txt 的状态修改时间为 16:10。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ touch test.txt -c
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ ls -l | grep test.txt
-rw-rw-r-- 1 hytidel hytidel    0  5月 10 16:10 test.txt
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

1.2.3 -m 选项

用如下命令修改 test.txt 的数据修改时间为 2024-05-11 12:00:00。

```
touch test.txt -m 2024-05-11 12:00:00
```

1.3 rm 命令

功能	永久性地删除文件系统中指定的文件或目录。	
语法	rm [选项] 文件或目录	
选项	-f	强制删除，系统不询问。
	-i	系统在删除前会询问。
	-r	递归删除，主要用于删除目录及其下的所有文件。

1.3.1 无选项

用如下命令删除 test.txt。

```
rm test.txt
```

运行结果如下图所示。test.txt 文件被删除，删除前系统无询问。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ rm test.txt
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换)

文件

命令

/home/hytidel/OSexp/coexp1

OSexp

coexp1

os-exp-fit

exp1

exp2

exp3

Pictures

sh matt-read-demo.c

sh matt-write-demo

sh matt-write-demo.c

shmget-demo

shmget-demo.c

zombie-demo

zombie-demo.c

738 B

15.9 KB

787 B

15.8 KB

452 B

15.8 KB

429 B

DevCpp.c

文件

DevCpp.c

文件

DevCpp.c

文件

DevCpp.c

1.3.2 -r 选项

新建一个如下图所示的文件树。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ tree testDir
testDir
├── testDir2
│   └── testFile3
├── testFile
└── testFile2

1 directory, 3 files
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

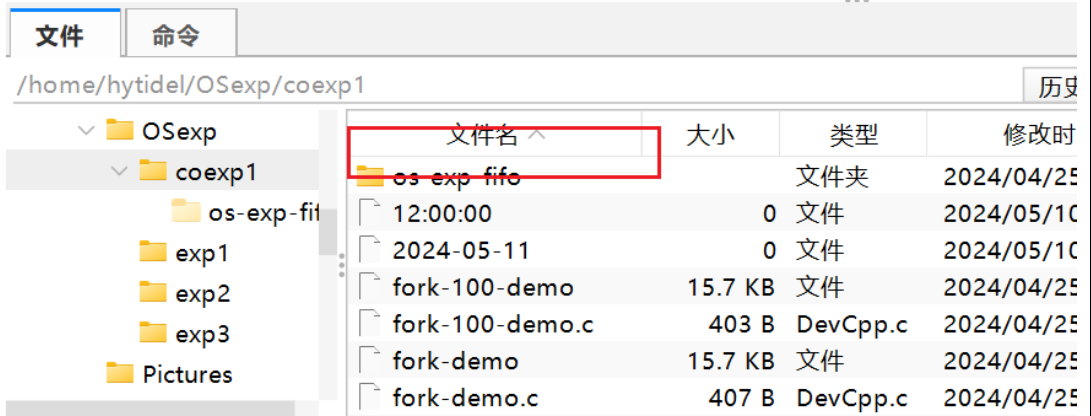
用如下命令递归地删除 testDir 目录及其下的文件和子目录。

```
rm testDir -r
```

运行结果如下图所示。testDir 目录消失。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ rm testDir -r
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换)



### 1.4 cp 命令

功能	复制文件或目录。	
语法	cp [选项] 源文件 目标文件	
选项	-p	保留文件。
	-r 或 -R	递归复制。
	-u	仅复制更新的文件。
	-i	交互式复制。

#### 1.4.1 无选项

用如下命令新建一个如下图所示的文件树。

```
mkdir testDir
mkdir testDir/testDir2
touch testDir/testFile
touch testDir/testFile2
touch testDir/testDir2/testFile3
```

用如下命令查看目录树结构。

```
tree testDir
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ tree testDir
testDir
├── testDir2
│   └── testFile3
├── testFile
└── testFile2

1 directory, 3 files
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$
```

用如下命令将文件 testDir/testFile 复制到 testDir/testDir2 目录下，并查看当前的目录树结构。

```
cp testDir/testFile testDir/testDir2/testFile4
tree testDir
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ cp testDir/testFile testDir/testDir2/testFile4
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ tree testDir
testDir
├── testDir2
│   ├── testFile3
│   └── testFile4
├── testFile
└── testFile2

1 directory, 4 files
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$
```

1.4.2 -r 选项

用如下命令递归地将目录 testDir/testDir2 复制一份到目录 testDir 下，起名 testDir3，并查看当前的目录树结构。

```
cp testDir/testDir2 testDir/testDir3 -r
tree testDir
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ tree testDir
testDir
├── testDir2
│   ├── testFile3
│   └── testFile4
├── testDir3
│   ├── testFile3
│   └── testFile4
├── testFile
└── testFile2

2 directories, 6 files
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$
```

1.5 mv 命令

功能	移动文件或目录，也可用于重命名文件或目录。	
语法	mv [选项] [源文件] [目的文件]	
选项	-i	交互式移动。
	-f	强制移动。



	-n	不覆盖已存在的目标文件。
	-u	仅当源文件比目标文件新，或者目标文件不存在时，才移动文件。
	-v	详细模式，显示命令的执行过程。

### 1.5.1 无选项

用如下命令将 testDir 目录下的 testDir2 目录移动到 testDir/testDir3 目录下，并查看当前的文件树。

```
mv testDir/testDir2 testDir/testDir3
tree testDir
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ tree testDir
testDir
├── testDir3
│   ├── testDir2
│   │   ├── testFile3
│   │   └── testFile4
│   ├── testFile3
│   └── testFile4
├── testFile
└── testFile2

2 directories, 6 files
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$
```

### 1.5.2 -v 选项

用如下命令将 testDir/testDir3 目录下的 testDir2 目录移动到 testDir 目录下，显示详细过程，并查看当前的文件树。

```
mv testDir/testDir3/testDir2 testDir -v
tree testDir
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ mv testDir/testDir3/testDir2 testDir -v
renamed 'testDir/testDir3/testDir2' -> 'testDir/testDir2'
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$ tree testDir
testDir
├── testDir2
│   ├── testFile3
│   └── testFile4
├── testDir3
│   ├── testFile3
│   └── testFile4
├── testFile
└── testFile2

2 directories, 6 files
hytidel@hytidel-virtual-machine:~/0Sexp/coexp1$
```

## 1.6 mkdir 命令

功能	创建目录。	
语法	mkdir [选项] 目录名	
选项	-m	设定权限模式（如 755 或 644）。这允许在创建目录的同时设定其权限。

	-p	允许创建多级目录，即如果父目录不存在，mkdir 将会创建它。
	-v	显示命令在执行过程中所创建的目录。

### 1.6.1 无选项

用如下命令在 testDir 目录下创建目录 testDir4，并查看当前的目录树。

```
mkdir testDir/testDir4
tree testDir
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ mkdir testDir/testDir4
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ tree testDir
testDir
├── testDir2
│   ├── testFile3
│   └── testFile4
├── testDir3
│   ├── testFile3
│   └── testFile4
├── testDir4
├── testFile
└── testFile2

3 directories, 6 files
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

### 1.6.2 -r 选项

用如下命令在 testDir 目录下创建目录 testDir5/testDir6，并查看当前的目录树。

```
mkdir testDir/testDir5/testDir6 -p
tree testDir
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ mkdir testDir/testDir5/testDir6 -p
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ tree testDir
testDir
├── testDir2
│   ├── testFile3
│   └── testFile4
├── testDir3
│   ├── testFile3
│   └── testFile4
├── testDir4
├── testDir5
│   └── testDir6
├── testFile
└── testFile2

5 directories, 6 files
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

## 1.7 df 命令

功能	查看文件系统的磁盘占用情况。	
语法	df [选项]	
选项	-a	包含全部的文件系统。

-h	以可读性较高的方式来显示信息。
-T	显示文件系统的类型。

### 1.7.1 无选项

用如下命令查看文件系统的磁盘占用情况。

```
df

运行结果如下图所示。

hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
tmpfs            396356      2012    394344    1% /run
/dev/sda3       19946096 11880776    7026780   63% /
tmpfs           1981764        0    1981764    0% /dev/shm
tmpfs            5120         4        5116    1% /run/lock
/dev/sda2       524252      6220    518032    2% /boot/efi
tmpfs           396352      156    396196    1% /run/user/1000
/dev/sr1        4899762 4899762        0 100% /media/hytidel/Ubuntu 22.04.4 LTS amd64
/dev/sr0        155532    155532        0 100% /media/hytidel/CDROM
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

### 1.7.2 -h 选项

用如下命令查看文件系统的磁盘占用情况，以可读性较高的方式显示。

```
df -h

运行结果如下图所示。

hytidel@hytidel-virtual-machine:~/OSexp/coexp1$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            388M   2.0M   386M   1% /run
/dev/sda3       20G    12G   6.8G   63% /
tmpfs            1.9G     0    1.9G   0% /dev/shm
tmpfs            5.0M   4.0K   5.0M   1% /run/lock
/dev/sda2       512M   6.1M   506M   2% /boot/efi
tmpfs            388M  156K   387M   1% /run/user/1000
/dev/sr1        4.7G   4.7G     0 100% /media/hytidel/Ubuntu 22.04.4 LTS amd64
/dev/sr0        152M  152M     0 100% /media/hytidel/CDROM
hytidel@hytidel-virtual-machine:~/OSexp/coexp1$
```

## 2. 文件的硬链接与软链接

学习 Linux 文件系统中关于文件硬链接和软链接的概念和相关操作命令,创建软硬链接各一个,给出实验证据表明它们分别是软硬连接。

### 2.1 硬链接与软链接

#### 2.1.1 硬链接

就好比创建了一个原文件的副本,硬链接文件与原始文件的 inode 信息块以及文本内容都是一样的,只是文件名字不同。由于硬链接文件与原始文件的 inode 是相同的,即一个 inode 节点对应两个不同的文件名,两个文件名指向同一个文件,所以不管是编辑硬链接文件还是原文件,保存之后两者显示的文本内容都是相同的,但是删除其中任何一个都不会影响另外一个的访问,因此只删除一个连接并不影响索引节点本身和它的连接,只有当最后一个连接被删除后,文件的数据块及目录的连接才会被释放。也就是说,文件真正删除的条件是与之相关的所有硬连接文件均被删除。

#### 2.1.2 软链接

指的是指向原始文件的实际链接,也称为符号链接,有点类似于 Windows 的快捷方式。在软链接中存放的不是具体的文件数据,而是所链接的原始文件的路径名,当打开软链接的时候,则会根据这个路径去找到并打开所链接的原文件。编辑软链接的文本就

是在编辑软链接所链接的原文件的文本，如果所链接的原文件被删除了，这个软链接就找不到当初所链接的原文件，就成了无效的链接，这时再次打开软链接就会提示将创建一个新的原文件。删除软链接不会影响到所链接的原文件。

## 2.2 创建硬软链接

功能	创建文件链接。	
语法	ln [选项] 源文件 链接文件	
选项	无	创建硬链接。
	-s	创建软链接。

## 2.3 硬软链接的检验

见 3. 构建目录树。

## 2.4 硬软链接的对比

### 2.4.1 硬链接

(1) 同一原文件创建的多个硬链接 inode 节点和 block 块都与原文件一样，可以看做同一个文件，只是文件名不同。

(2) 编辑其中的任一文件，其它文件都会随之改变。

(3) 删除其中任一的文件，都会不影响到其他文件，除非把相关联的所有文件都删除，那么这个文件才会被删除。

(4) 硬链接不能跨分区使用。

(5) 硬链接不能针对目录使用。

### 2.4.2 软链接

(1) 每个软链接都拥有自己的 inode 节点和 block 块。

(2) 软链接不保存实际的数据，只保存原始文件的路径。

(3) 通过软链接打开或者编辑文件其实操作的就是所链接的原文件。

(4) 不管是通过编辑原文件还是通过其软链接去编辑文件，该原文件下的其他软链接打开的文件都随之改变。

(5) 软连接的文件类型标志为：l，软链接的权限都为 rwxrwxrwx。

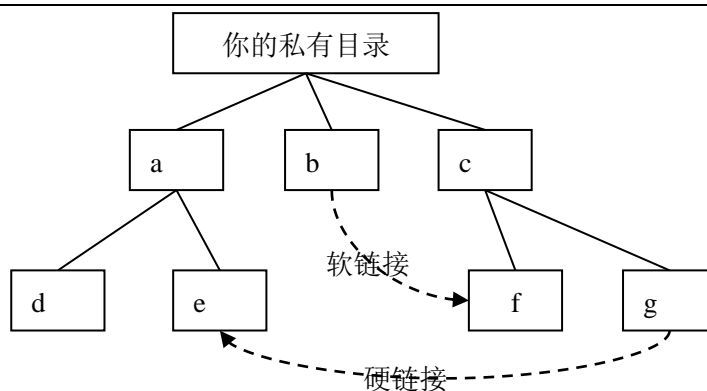
(6) 删除原文件，其对应的软链接都不能使用，将会称为无效链接。

(7) 删除其中任一软链接文件，原文件以及该原文件下的其他软链接文件依然可用。

## 3. 构建目录树

构建以下目录子树：





### 3.1 建立目录树

用如下命令先建立无链接的部分，即目录 a、c 和文件 d、g，并查看目录树。

```
mkdir a
mkdir c
touch a/d
touch c/g
tree
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ mkdir a
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ mkdir c
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ touch a/d
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ touch c/g
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ tree
.
├── a
│   └── d
└── c
    └── g

2 directories, 2 files
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$
```

再创建 c/g 到 a/e 的硬链接，和 c/f 到 b 的软链接，并查看目录树。

```
ln c/g a/e
ln c/f b -s
tree
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$ ln c/g a/e
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$ ln c/f b -s
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$ tree
.
├── a
│   ├── d
│   └── e
├── b -> c/f
└── c
    └── g

2 directories, 4 files
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$
```



## 3.2 硬软链接的检验

### 3.2.1 检验硬链接

用如下命令将文本“text in e”写入文件 a/e，并查看文件 c/g 的内容。

```
echo "text in e" > a/e
cat c/g
```

运行结果如下图所示，a/e 中的内容同步到了 c/g 中。

```
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ echo "text in e" > a/e
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ cat c/g
text in e
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$
```

用如下命令将文件 c/g 移到当前目录下，并向文件 a/e 写入文本“test in e 2”。查看文件树，并查看文件 g 的内容。

```
mv c/g g
tree
echo "text in e 2" > a/e
cat g
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ mv c/g g
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ tree
.
├── a
│   ├── d
│   └── e
├── b
├── c
│   └── f -> b
└── g

2 directories, 5 files
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ echo "text in e 2" > a/e
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ cat c/g
cat: c/g: No such file or directory
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ cat g
text in e 2
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$
```

这表明：硬链接记录的是源文件的唯一标识符，源文件移动位置时，硬链接也能找到位置。

### 3.2.2 检验软链接

用如下命令将文本“text in b”写入文件 b，并查看文件 c/f 的内容。

```
echo "text in b" > b
cat c/f
```

运行结果如下图所示，文件 b 的内容同步到文件 c/f 中。

```
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ echo "text in b" > b
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ cat c/f
text in b
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$
```

用如下命令将文件 b 移到 a 目录下，查看文件树，并将文本“text in b 2”写入文件 a/b，并查看文件 c/f 的内容。

```
mv b a/b
echo "text in b 2" > a\b
cat c/f
```

运行结果如下图所示。

```
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$ mv b a/b
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$ echo "text in b 2" > a\b
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$ cat c/f
text in b
hytidel@hytidel-virtual-machine:~/0Sexp/exp4.2$
```

这表明：软链接记录源文件的路径。源文件移动位置后，无法通过软链接找到。

## 四、实验结论

（提供运行结果，对结果进行探讨、分析、评价，并提出结论性意见和改进想法）

### 1. Linux 的文件命令

命令	功能
ls	列出当前目录中的文件或子目录。
touch	创建空文件或修改文件的时间戳。
rm	永久性地删除文件系统中指定的文件或目录。
cp	复制文件或目录。
mv	移动文件或目录，也可用于重命名文件或目录。
mkdir	创建目录。
df	查看文件系统的磁盘占用情况。

### 2. 文件的硬链接与软链接

#### 2.4.1 硬链接

（1）同一原文件创建的多个硬链接 inode 节点和 block 块都与原文件一样，可以看做同一个文件，只是文件名不同。

（2）编辑其中的任一文件，其它文件都会随之改变。

（3）删除其中任一的文件，都会不影响到其他文件，除非把相关联的所有文件都删除，那么这个文件才会被删除。

（4）硬链接不能跨分区使用。

（5）硬链接不能针对目录使用。

#### 2.4.2 软链接

（1）每个软链接都拥有自己的 inode 节点和 block 块。

（2）软链接不保存实际的数据，只保存原始文件的路径。

（3）通过软链接打开或者编辑文件其实操作的就是所链接的原文件。

(4) 不管是通过编辑原文件还是通过其软链接去编辑文件，该原文件下的其他软链接打开的文件都随之改变。

(5) 软连接的文件类型标志为：l，软链接的权限都为 rwxrwxrwx。

(6) 删除原文件，其对应的软链接都不能使用，将会称为无效链接。

(7) 删除其中任一的软链接文件，原文件以及该原文件下的其他软链接文件依然可用。

### 3. 构建目录树

#### 3.1 目录树

建立了如下图所示的目录树。

```
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ touch b
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$ tree
.
├── a
│   ├── d
│   └── e
├── b
├── c
└── f -> b
    └── g

2 directories, 5 files
hytidel@hytidel-virtual-machine:~/0Sexp/exp4$
```

#### 3.2 检验硬软链接

通过实验检验了两个链接一个是硬链接，一个是软链接。

(1) 硬链接记录的是源文件的唯一标识符，源文件移动位置时，硬链接也能找到位置。

(2) 软链接记录源文件的路径。源文件移动位置后，无法通过软链接找到。

## 五、实验体会

(根据自己情况填写)

本次《文件系统实验》通过对 Linux 文件命令行操作命令的学习和实践，让我对 Linux 操作系统中的文件系统有了更深入的了解。以下是我的实验心得：

通过本次实验，我对 Linux 文件系统中的文件命令行操作命令有了更深入的了解，掌握了常用命令的使用方法。同时，我也学会了如何创建硬链接和软链接，并且通过实验验证了它们的区别。这些知识和技能对我今后的学习和工作都将有很大的帮助。

---

注：“指导教师批阅意见”栏请单独放置一页

---

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：