

河南工业大学 Linux 基础与应用 实验报告

专业班级: 软件 1305 班 学号: 201316920311 姓名: 田劲锋 指导老师: 赵晨阳 评分: _____

实验题目: Linux 常用工具的使用 (1): Linux 文件系统的挂载与卸载工具

实验目的: (1) 理解 Ubuntu 系统的文件系统组成; (2) 掌握挂载和卸载文件系统的方法

实验内容:

- (1) 查看 Ubuntu 系统的目录结构, 并写出每个目录的作用。
- (2) 在虚拟机上添加一块未分区的硬盘, 创建文件系统, 并挂载该文件系统, 访问该分区, 创建一个目录和文件, 然后卸载该文件系统, 再次访问刚刚创建的目录和文件, 看是否能够被访问。

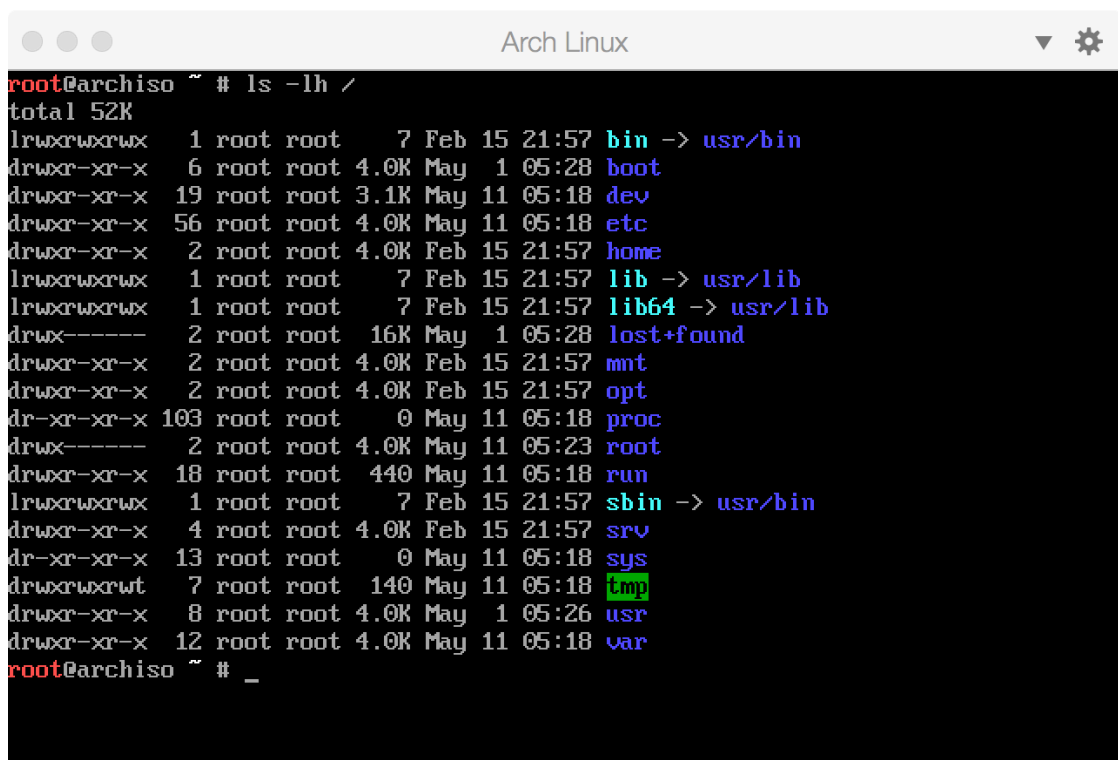
实验步骤:

这次的实验我在 Mac OS X 上的 Parallels Desktop 中建立了一个虚拟机, 并挂载上了 Arch Linux 的安装光盘镜像文件, 以进行操作。

- (1) 不同的 UNIX/Linux 操作系统有着不同的根目录结构, 但基本上还是一致的。键入:

```
# ls -lh /
```

来查看根目录结构, 如图1。



```
root@archiso ~ # ls -lh /
total 52K
lrwxrwxrwx 1 root root 7 Feb 15 21:57 bin -> usr/bin
drwxr-xr-x 6 root root 4.0K May 1 05:28 boot
drwxr-xr-x 19 root root 3.1K May 11 05:18 dev
drwxr-xr-x 56 root root 4.0K May 11 05:18 etc
drwxr-xr-x 2 root root 4.0K Feb 15 21:57 home
lrwxrwxrwx 1 root root 7 Feb 15 21:57 lib -> usr/lib
lrwxrwxrwx 1 root root 7 Feb 15 21:57 lib64 -> usr/lib
drwx----- 2 root root 16K May 1 05:28 lost+found
drwxr-xr-x 2 root root 4.0K Feb 15 21:57 mnt
drwxr-xr-x 2 root root 4.0K Feb 15 21:57 opt
dr-xr-xr-x 103 root root 0 May 11 05:18 proc
drwx----- 2 root root 4.0K May 11 05:23 root
drwxr-xr-x 18 root root 440 May 11 05:18 run
lrwxrwxrwx 1 root root 7 Feb 15 21:57 sbin -> usr/bin
drwxr-xr-x 4 root root 4.0K Feb 15 21:57 srv
dr-xr-xr-x 13 root root 0 May 11 05:18 sys
drwxrwxrwt 7 root root 140 May 11 05:18 tmp
drwxr-xr-x 8 root root 4.0K May 1 05:26 usr
drwxr-xr-x 12 root root 4.0K May 11 05:18 var
root@archiso ~ # _
```

图 1: 目录结构

/bin/ 二进制可执行文件。

/boot/ 引导程序文件。

/dev/ 必要设备。

/etc/ 特定主机, 系统范围内的配置文件。

/home/ 用户的家目录。
/lib/ 必要的库文件。
/lib64/ 必要的库文件（64位平台）。
/lost+found/ 恢复文件。
/mnt/ 临时挂载的文件系统。
/opt/ 可选应用软件包。
/proc/ 虚拟文件系统，将内核与进程状态归档为文本文件。
/root/ 超级用户的家目录
/run 代替/var/run目录。自最后一次启动以来运行中的系统的信息。
/sbin/ 必要的系统二进制文件。
/srv/ 服务器站点的具体数据，由系统提供。
/sys/ 系统链接。
/tmp/ 临时文件，在系统重启时目录中文件不会被保留。
/usr/ 用于存储只读用户数据的第二层次；包含绝大多数的用户工具和应用程序。
 /usr/bin/ 非必要可执行文件 (在单用户模式中不需要)；面向所有用户。
 /usr/include/ 标准包含文件，即C语言头文件。
 /usr/lib/ /usr/bin/和/usr/sbin/中二进制文件的库。
 /usr/sbin/ 非必要的系统二进制文件。
/var/ 变量文件——在正常运行的系统中其内容不断变化的文件。

(2) 显示磁盘信息：

```
# lsblk
```

如图2，这里我们看到有一未挂载的本地磁盘sda。
在/dev/sda上建立一个EXT4格式的文件系统：

```
# mkfs.ext4 /dev/sda
```

然后如图3，将其挂载到/mnt中：

```
# mount /dev/sda /mnt
```

创建一个文件和目录：

```
# cd /mnt  
# touch file  
# mkdir dir  
# ls
```

最后卸载之：

```
# umount /mnt
```

如图4，测试文件是否还能访问：

```
# ls /mnt
```

```
Arch Linux
root@archiso ~ # lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                   8:0    0   64G  0 disk
sr0                   11:0    1   626M  0 rom  /run/archiso/bootmnt
loop0                 7:0    0   272M  1 loop /run/archiso/sfs/airootfs
loop1                 7:1    0    32G  1 loop
└─arch_airootfs       254:0    0    32G  0 dm  /
loop2                 7:2    0   256M  0 loop
└─arch_airootfs       254:0    0    32G  0 dm  /
root@archiso ~ # mkfs.ext4 /dev/sda
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 16777216 4k blocks and 4194304 inodes
Filesystem UUID: 34114abd-9bdc-45a4-bc01-e2092cb95494
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # _
```

图 2: 创建文件系统

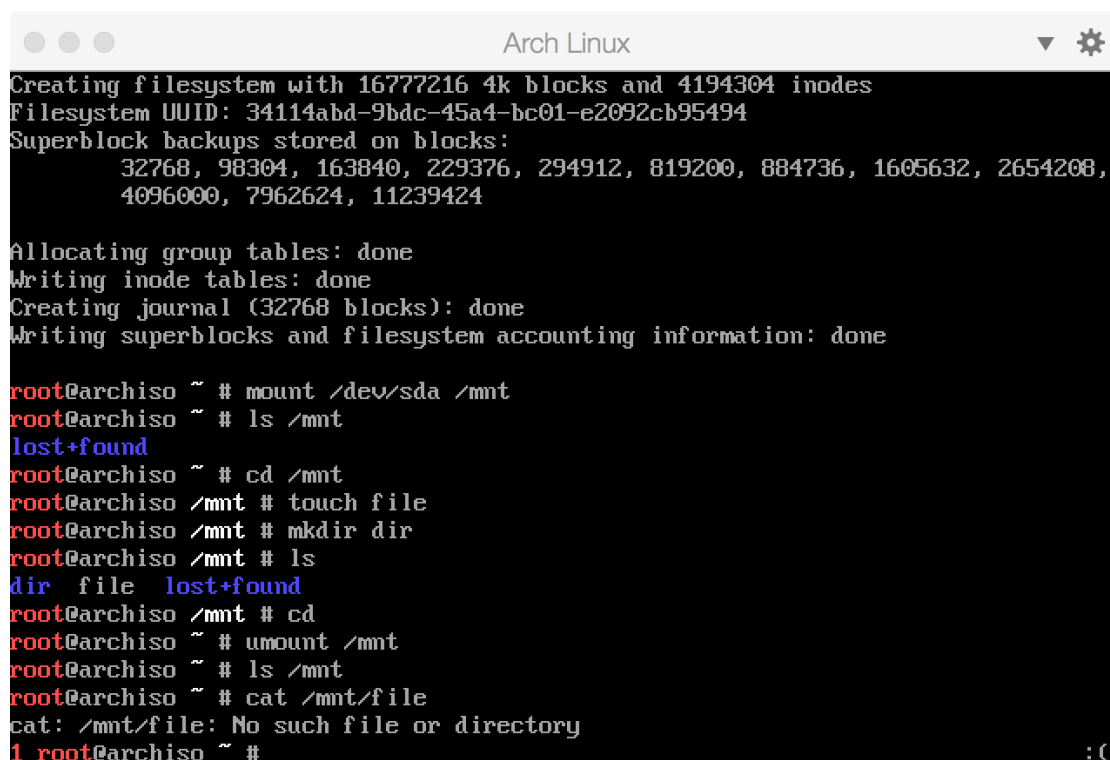
```
Arch Linux
└─arch_airootfs       254:0    0    32G  0 dm  /
loop2                 7:2    0   256M  0 loop
└─arch_airootfs       254:0    0    32G  0 dm  /
root@archiso ~ # mkfs.ext4 /dev/sda
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 16777216 4k blocks and 4194304 inodes
Filesystem UUID: 34114abd-9bdc-45a4-bc01-e2092cb95494
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mount /dev/sda /mnt
root@archiso ~ # ls /mnt
lost+found
root@archiso ~ # cd /mnt
root@archiso /mnt # touch file
root@archiso /mnt # mkdir dir
root@archiso /mnt # ls
dir file lost+found
root@archiso /mnt # _
```

图 3: 挂载文件系统

当然结论是不能访问了（笑）。

A terminal window titled "Arch Linux" with standard window controls. The terminal output shows the process of creating a filesystem on a device, mounting it, creating files and directories, and then unmounting it. The final command shows an error when trying to access a file that was not properly created or is inaccessible.

```
Creating filesystem with 16777216 4k blocks and 4194304 inodes
Filesystem UUID: 34114abd-9bdc-45a4-bc01-e2092cb95494
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mount /dev/sda /mnt
root@archiso ~ # ls /mnt
lost+found
root@archiso ~ # cd /mnt
root@archiso /mnt # touch file
root@archiso /mnt # mkdir dir
root@archiso /mnt # ls
dir file lost+found
root@archiso /mnt # cd
root@archiso ~ # umount /mnt
root@archiso ~ # ls /mnt
root@archiso ~ # cat /mnt/file
cat: /mnt/file: No such file or directory
1 root@archiso ~ # _
```

图 4: 卸载文件系统

实验体会:

Linux 的目录结构继承UNIX，是经历实践检验有效的分类方式。

关于Linux文件系统，主要在安装系统中会遇到，尤其是Arch这种以命令行形式安装的发行版，需要手动分区和挂载安装。这里的实验也是直接利用了Arch的系统安装盘自带的一些功能。