

# 实验八：Linux 文件操作二

姓名：汪江豪 学号：22121630 实验日期：2024.12.30

---

## 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：

- 熟悉 Linux 下的基本操作，学会使用各种 Shell 命令去操作 Linux，对 Linux 有一个感性认识
- 熟悉 Linux 下文件打包、解压的命令及 RPM 包的操作
- 熟悉 Linux 下文件系统管理命令，以及加载其他分区的方法

## 实验内容：

- 文件打包、解压(tar、gzip)
- 挂载文件系统
- 文件系统维护命令

### 一、文件打包、解包(tar、gzip)

#### 操作过程 1：

【操作要求 1】tar：掌握 tar 的参数：zcvfxt

【操作步骤】

实现以下操作：（目标目录可以选择/boot 或者其他较小的目录，包名自定义）

- ① 文件不压缩打包，格式：tar -cvf 包名.tar 目标目录
- ② 文件解包，格式：tar -xvf 包名.tar
- ③ 文件包测试，格式：tar -tf 包名.tar
- ④ 文件压缩打包，格式：tar -zcvf 包名.tar.gz 目标目录
- ⑤ 文件解压缩，格式：tar -zxvf 包名.tar.gz
- ⑥ 文件压缩包测试，格式：tar -ztf 包名.tar.gz 打包或者解包以后可以用 ls 看下结果。

#### 结果 1：

- ① 文件不压缩打包，格式：tar -cvf 包名.tar 目标目录

```
pioneer@pioneer-virtual-machine:~/os2/test8$ tar -cvf code.tar code
code/
code/mytext2.dat
code/mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar
```

- ② 文件解包，格式：tar -xvf 包名.tar

```
pioneer@pioneer-virtual-machine:~/os2/test8$ tar -xvf code.tar
code/
code/mytext2.dat
code/mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar
```

- ③ 文件包测试，格式：tar -tf 包名.tar

```
pioneer@pioneer-virtual-machine:~/os2/test8$ tar -tf code.tar
code/
code/mytext2.dat
code/mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar
```

- ④ 文件压缩打包，格式：tar -zcvf 包名.tar.gz 目标目录

```
pioneer@pioneer-virtual-machine:~/os2/test8$ tar -zcvf code.tar.gz code
code/
code/mytext2.dat
code/mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar  code.tar.gz
```

- ⑤ 文件解压缩，格式：tar -zxvf 包名.tar.gz

```
pioneer@pioneer-virtual-machine:~/os2/test8$ tar -zxvf code.tar.gz
code/
code/mytext2.dat
code/mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar  code.tar.gz
```

- ⑥ 文件压缩包测试，格式：tar -ztf 包名.tar.gz

```
pioneer@pioneer-virtual-machine:~/os2/test8$ tar -ztf code.tar.gz
code/
code/mytext2.dat
code/mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar  code.tar.gz
```

## 操作过程 2：

【操作要求 2】 gzip：掌握 gzip 的参数：-v -d

【操作步骤】

实现以下操作：

- ① 将当前目录下的每个文件压缩成.gz 文件， gzip \*
- ② 将当前目录下的每个压缩的文件解压，并列出详细信息， gzip -dv \*
- ③ 详细当前目录下的压缩文件的信息，但不进行解压， gzip -l \*
- ④ 递归的压缩目录， gzip -rv test
- ⑤ 递归的解压目录， gzip -drv test

## 结果 2：

- ① 将当前目录下的每个文件压缩成.gz 文件， gzip \*

```
pioneer@pioneer-virtual-machine:~/os2/test8$ gzip *
gzip: code is a directory -- ignored
gzip: code.tar.gz already exists; do you wish to overwrite (y or n)? y
gzip: code.tar.gz already has .gz suffix -- unchanged
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar.gz
```

- ② 将当前目录下的每个压缩的文件解压，并列出详细信息， gzip -dv \*

```
pioneer@pioneer-virtual-machine:~/os2/test8$ gzip -dv *
gzip: code is a directory -- ignored
code.tar.gz:      97.9% -- replaced with code.tar
pioneer@pioneer-virtual-machine:~/os2/test8$ ls
code  code.tar
```

- ③ 详细当前目录下的压缩文件的信息，但不进行解压， gzip -l \*

```
pioneer@pioneer-virtual-machine:~/os2/test8$ gzip -l *
gzip: code is a directory -- ignored
      compressed      uncompressed  ratio   uncompressed_name
            239                 10240  97.9%  code.tar
            239                 10240  97.9%  (totals)
```

- ④ 递归的压缩目录， gzip -rv 目录

```
pioneer@pioneer-virtual-machine:~/os2/test8$ gzip -rv code
code/code2/mytext3.txt:    0.0% -- replaced with code/code2/mytext3.txt.gz
code/mytext2.txt:          0.0% -- replaced with code/mytext2.txt.gz
code/mytext.txt:          0.0% -- replaced with code/mytext.txt.gz
```

```
pioneer@pioneer-virtual-machine:~/os2/test8$ cd code
pioneer@pioneer-virtual-machine:~/os2/test8/code$ ls
code2  mytext2.txt.gz  mytext.txt.gz
```

```
pioneer@pioneer-virtual-machine:~/os2/test8/code$ cd code2
pioneer@pioneer-virtual-machine:~/os2/test8/code/code2$ ls
mytext3.txt.gz
```

- ⑤ 递归的解压目录， gzip -drv 目录

```
pioneer@pioneer-virtual-machine:~/os2/test8$ gzip -drv code
code/code2/mytext3.txt.gz:      0.0% -- replaced with code/code2/mytext3.txt
code/mytext2.txt.gz:          0.0% -- replaced with code/mytext2.txt
code/mytext.txt.gz:          0.0% -- replaced with code/mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8$ cd code
pioneer@pioneer-virtual-machine:~/os2/test8/code$ ls
code2  mytext2.txt  mytext.txt
pioneer@pioneer-virtual-machine:~/os2/test8/code$ cd code2
pioneer@pioneer-virtual-machine:~/os2/test8/code/code2$ ls
mytext3.txt
```

思考： tar 与 gzip 有何区别、联系？

答：

tar 是用于打包多个文件和目录到一个单独的存档文件中的工具，便于传输或备份，其本身并不压缩文件。 gzip 是用于压缩文件的工具，能够减小文件的大小，通过使用特定的算法来压缩数据。

tar 和 gzip 通常结合使用，首先使用 tar 创建一个包含多文件的归档，然后使用 gzip 压缩存档文件，。这种组合用于创建. tar. gz 或. tgz 文件，是 linux 和 unix 系统中常见的

压缩存档格式。使用 tar 命令时，可以通过添加-z 选项直接调用 gzip，在打包同时进行压缩。

## 二、挂载文件系统

### 操作过程 1：

【操作要求 1】mount 命令，挂载磁盘

【操作步骤】

- ① 选择“VM”——“setting”并打开，将光标定位在 hard Disk 这一选项，然后点击下方的 Add 按钮，点击 next，执行下一个步骤，根据提示，创建一个虚拟的磁盘，并点击下一步。按照默认的点击下一步即可完成虚拟磁盘的添加。
- ② 使用“fdisk -l”的命令查看当前系统的分区（如果刚才设置 VMware—setting 的时候运行了系统，则会出现磁盘：没有识别到新的磁盘即 sdb），解决办法，重启虚拟机：shutdown -r now。如果执行第一步的时候是关闭虚拟机中的系统的，则使用“fdisk -l”命令的时候则会出现新的磁盘 sdb（不过提示未分区）
- ③ 对新建的磁盘进行分区及格式化的工作：输入 fdisk /dev/sdb，终端会提示：Command (m for help)
- ④ 根据提示输入：n，会出现提示，依次输入 p 和 1 即可。
- ⑤ 接着便会提示卷的起始地址和结束地址，都保持默认按回车的即可（只分一个区）
- ⑥ 输入“w”保存并推出，再次使用“fdisk -l”这个命令来查看会发现出现了 /dev/sdb1（说明已经完成了分区工作）
- ⑦ 对新建的分区进行格式化：输入命令 mkfs -t ext3 /dev/sdb1，格式化成 ext3 的文件系统即可
- ⑧ 手动挂载磁盘：使用 mount /dev/sdb1 /要挂载的目录（自己自定义）

### 结果 1：

- ① 选择“VM”——“setting”并打开，将光标定位在 hard Disk 这一选项，然后点击下方的 Add 按钮，点击 next，执行下一个步骤，根据提示，创建一个虚拟的磁盘，并点击下一步。按照默认的点击下一步即可完成虚拟磁盘的添加。



- ② 使用“fdisk -l”的命令查看当前系统的分区（如果刚才设置 VMware—setting 的时候运行了系统，则会出现磁盘：没有识别到新的磁盘即 sdb），解决办法，重启虚拟机：shutdown -r now。如果执行第一步的时候是关闭虚拟机中的系统的，则使用“fdisk -l”命令的时候则会出现新的磁盘 sdb（不过提示未分区）

```
单元：扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理)：512 字节 / 512 字节
I/O 大小(最小/最佳)：512 字节 / 512 字节

Disk /dev/loop8 : 12.2 MiB, 12791808 字节, 24984 个扇区
单元：扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理)：512 字节 / 512 字节
I/O 大小(最小/最佳)：512 字节 / 512 字节

Disk /dev/loop9 : 12.32 MiB, 12922880 字节, 25240 个扇区
单元：扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理)：512 字节 / 512 字节
I/O 大小(最小/最佳)：512 字节 / 512 字节
```

- ③ 对新建的磁盘进行分区及格式化的工作：输入 fdisk /dev/sdb，终端会提示：Command (m for help)

```
欢迎使用 fdisk (util-linux 2.37.2)。
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

设备不包含可识别的分区表。
创建了一个磁盘标识符为 0x2c636582 的新 DOS 磁盘标签。

命令(输入 m 获取帮助)： ■
```

- ④ 根据提示输入：n，会出现提示，依次输入 p 和 1 即可。

```
欢迎使用 fdisk (util-linux 2.37.2)。
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

设备不包含可识别的分区表。
创建了一个磁盘标识符为 0x2c636582 的新 DOS 磁盘标签。

命令(输入 m 获取帮助)： n
分区类型
  p  主分区 (0 primary, 0 extended, 4 free)
  e  扩展分区 (逻辑分区容器)
选择 (默认 p)： p
分区号 (1-4, 默认 1)： 1
第一个扇区 (2048-4194303, 默认 2048)：
```

- ⑤ 接着便会提示卷的起始地址和结束地址，都保持默认按回车即可(只分一个区)

```
欢迎使用 fdisk (util-linux 2.37.2)。
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

设备不包含可识别的分区表。
创建了一个磁盘标识符为 0x2c636582 的新 DOS 磁盘标签。

命令(输入 m 获取帮助)： n
分区类型
  p  主分区 (0 primary, 0 extended, 4 free)
  e  扩展分区 (逻辑分区容器)
选择 (默认 p)： p
分区号 (1-4, 默认 1)： 1
第一个扇区 (2048-4194303, 默认 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, 默认 4194303)：

创建了一个新分区 1，类型为“Linux”，大小为 2 GiB。

命令(输入 m 获取帮助)： ■
```

⑥ 输入 “w” 保存并推出，再次使用 “fdisk -l” 这个命令来查看会发现出现了 /dev/sdb1(说明已经完成了分区工作)

```
Disk /dev/sdb: 2 GiB, 2147483648 字节, 4194304 个扇区
Disk model: VMware Virtual S
单元：扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理)：512 字节 / 512 字节
I/O 大小(最小/最佳)：512 字节 / 512 字节
磁盘标签类型：dos
磁盘标识符：0x2c636582

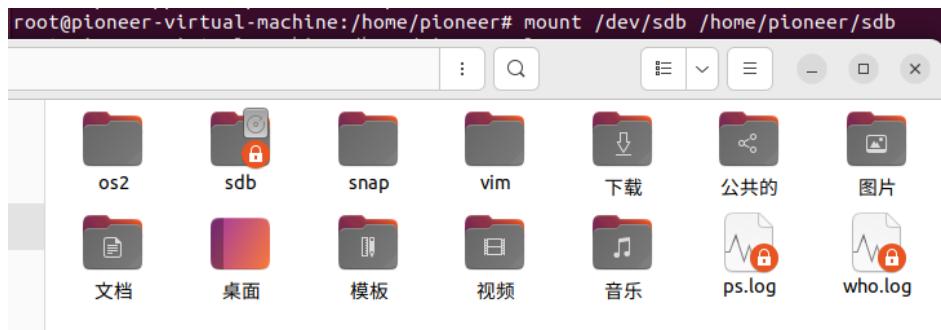
设备 启动 起点 末尾 扇区 大小 Id 类型
/dev/sdb1          2048 4194303 4192256 2G 83 Linux
```

⑦ 对新建的分区进行格式化：输入命令 mkfs -t ext3 /dev/sdb1, 格式化成 ext3 的文件系统即可

```
root@pioneer-virtual-machine:/home/pioneer# mkfs -t ext3 /dev/sdb
mke2fs 1.46.5 (30-Dec-2021)
在 /dev/sdb 中发现一个 dos 分区表
无论如何也要继续？ (y,N) y
创建含有 524288 个块 (每块 4k) 和 131072 个 inode 的文件系统
文件系统 UUID : 21486b58-a2d3-42d1-999f-7e25bae5fa92
超级块的备份存储于下列块：
32768, 98304, 163840, 229376, 294912

正在分配组表： 完成
正在写入 inode表： 完成
创建日志（16384 个块）： 完成
写入超级块和文件系统账户统计信息： 已完成
```

⑧ 手动挂载磁盘：使用 mount /dev/sdb1 /要挂载的目录(自己自定义)



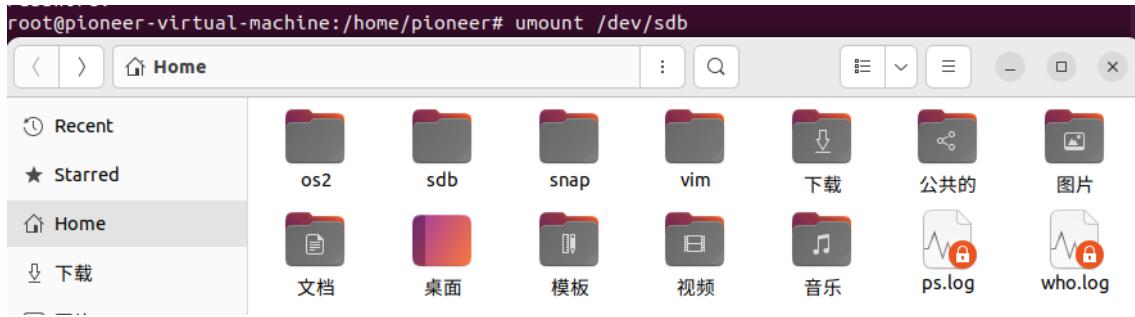
## 操作过程 2：

【操作要求 2】 umount 命令，卸载文件系统

【操作步骤】

- ① # umount -v /dev/sda1                  通过设备名卸载
- ② # umount -v /mnt/mymount/            通过挂载点卸载

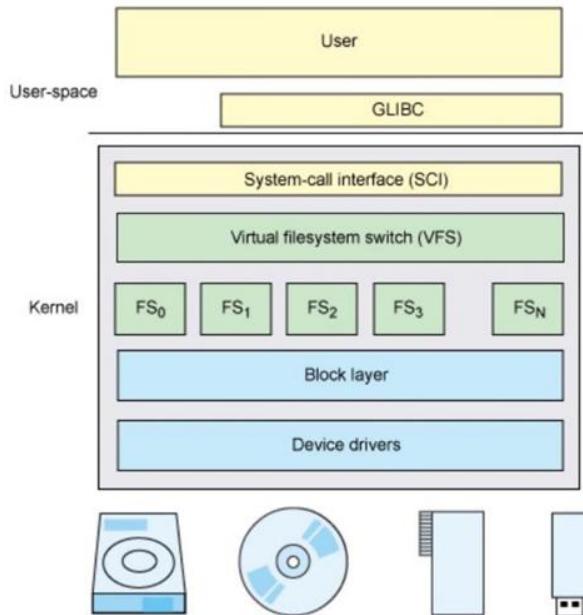
## 结果 2：



**思考:** Linux 是如何应对多文件系统?

**答:**

Linux 使用虚拟文件系统(VFS)来应对多文件系统。VFS 作为文件系统与用户空间之间的抽象层。提供了一组标准的文件操作 API，使得不同的文件系统可以在 Linux 内核中以统一的方式被访问。VFS 只存在于内存中，不存在于外存空间，VFS 在系统启动时建立，在系统关闭时消亡。其在 Linux 系统中的架构层次如图：



Linux 通过挂载管理，允许将多个文件系统挂载到单一目录树中。每个挂载点对应一个文件系统。Mount 命令用于挂载文件系统，umount 命令用于卸载文件系统。挂载点可以是任何已经存在的目录，挂载后，该目录下的原有内容将被隐藏，直到文件系统被卸载。

Linux 内核支持多种文件系统类型，如 EXT4、XFS、Btrfs、NFS、SMB 等。

### 三、文件系统维护命令

#### 操作过程 1：

【操作要求 1】 du 命令，功能：用于查看磁盘使用情况

【操作步骤】 格式： du 目录名

#### 结果 1：

```
pioneer@pioneer-virtual-machine:~/os2/test8$ du
8      ./code/code2
20     ./code
36     .
```

## 操作过程 2:

【操作要求 2】 df 命令, 功能: 用于查看磁盘剩余情况

【操作步骤】 格式: df [option]

## 结果 2:

```
pioneer@pioneer-virtual-machine:~/os2/test8$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
tmpfs            396152     1964   394188   1% /run
/dev/sda3        19946096 12358340   6549216  66% /
tmpfs            1980748      0   1980748   0% /dev/shm
tmpfs             5120       4     5116   1% /run/lock
/dev/sda2        524252     6228   518024   2% /boot/efi
tmpfs            396148     104   396044   1% /run/user/1000
/dev/sr0          4899762   4899762      0 100% /media/pioneer/Ubuntu 22.04.4 LTS amd64
```

## 操作过程 3:

【操作要求 3】 fsck 命令, 功能: 扫描文件系统内容检查内部一致性的工具

【操作步骤】 格式: fsck 文件系统, 如: fsck / (查根文件系统) (选做)

## 结果 3:

系统提示: 不应该用 fsck 检查已挂载的磁盘, 此行为很可能会对磁盘 造成永久性的伤害。因此, 使用刚卸载的/dev/sdb 进行实验, 结果如下:

```
root@pioneer-virtual-machine:/home/pioneer/os2/test8# fsck /dev/sdb
fsck, 来自 util-linux 2.37.2
e2fsck 1.46.5 (30-Dec-2021)
/dev/sdb : 没有问题, 11/131072 文件, 25405/524288 块
```

## 操作过程 4:

【操作要求 4】 free 命令, 功能: 查看系统的物理内存和虚拟内存的使用情况

【操作步骤】 格式: free

## 结果 4:

```
pioneer@pioneer-virtual-machine:~/os2/test8$ free
              total        used        free      shared  buff/cache   available
Mem:      3961496     994656     1925896        34372    1040944     2698088
Swap:     2191356         0     2191356
```

思考: free 命令显示内容分别代表什么?

答:

total 总内存量。系统可用的总物理内存量和总交换量。

Used 已使用的内存量。是目前已分配的内存量, 包括应用程序使用的内存和内核适使用的内存。

Free 未使用的内存量。没有分配给任何进程的内存量。

Shared 共享内存量。被多个进程共享的内存量。

Buff/cache 用于缓冲和缓存的内存量。包括文件系统缓存和磁盘 I/O 缓冲区。

Available 可用内存量。估计在没有发生交换的情况下，可以立即分配给应用程序的内存量，该值是 free 和 buff/cache 的总和。

## 实验体会：

本次实验，我通过动手实践，了解了更多的 linux 系统的一些基本操作，包括但不限于如何打包文件、解包文件、压缩和解压文件。了解了如何挂载文件系统，了解文件系统的维护命令。

操作系统涉及的知识面是极为广泛的，尤其是当代最新的 操作系统，拥有许多先进特性，我还需要不断学习，来进一步提高我的知识。