

## 1、文件权限操作相关的命令

### 1.1 修改文件的权限的命令---> chmod

## 1.2 修改文件所属的用户和组的命令---> chown

### 1.3 修改文件所属的组的命令-->chgrp

## 2、链接文件

## 2.1 创建软链接文件--> ln

## 2.2 硬链接文件

### 3、开关机相关的命令

#### 4、磁盘操作相关的命令

## 4.1 关于VMware软件的配置

#### 4.1.1 配置VMware软件支持usb3.1接口

#### 4.1.2 让u盘被ubuntu系统识别

### 4.1.3 查看u盘是否被ubuntu系统识别

#### 4.2 查看磁盘的使用率的命令 ---> df

#### 4.3 磁盘删除分区/重新分区的命令-->fdisk

#### 4.4 对分区进行格式化--> mkfs.fat

#### 4.5 磁盘挂载(mount)和取消挂载(umount)的命令

#### 4.6 模拟一个磁盘，完成以上命令的练习

## 5、网络相关的概念及命令

## 5.1 网络相关的概念

### 5.1.1 IP地址

### 5.1.2 网关

### 5.1.3 子网掩码

#### 5.1.4 DNS服务

## 5.2 查看windows系统的IP地址->ipconfig

### 5.3 查看ubuntu系统的IP地址-》ifconfig

## 5.4 测试网络的命令->ping

## 5.5 重启ubuntu系统网络服务的命令

## ## shell编程

## 1、shell相关的概念

## 2、第一个shell脚本文件

### 3、 shell脚本文件中变量的使用

### 3.1 shell脚本文件中定义变量

### 3.2 shell中定义的变量的引用

### 3.3 shell中位置变量的使用

### 3.4 修饰变量的关键字

### 3.5 命令置换符

#### 4、作业

## 5、明天的授课内容

## 1、文件权限操作相关的命令

## 1.1 修改文件的权限的命令---> chmod

1	drwxr-xr-x	3	linux	linux	4096	3月	27	11:47	DC23032	
2									--->	文件或者目录名
3									--->	文件的时间戳
4									--->	文件的大小

```

5 | | | | | | | |----> 文件所属的组
6 | | | | | | |----> 文件所属用户
7 | | | | | |----> 硬链接数(C高级具体讲解)
8 | | | |----> 其他用户的权限
9 | | |----> 组的权限
10 | |----> 用户的权限
11 |----> 文件的类型
12
13 1. 文件类型: bsp-lcd
14     b : block块设备文件(linux驱动开发)
15     s : socked套接字文件(网络编程)
16     p : pipe管道文件(IO进程线程)
17     - : 普通文件(文本文件)
18     l : link链接文件(C高级)
19     c : char字符设备文件(linux驱动开发)
20     d : directory 目录(文件夹)
21
22 2. 文件权限:
23     r: 可读
24     w: 可写
25     x: 可执行
26     -: 没有权限
27
28 3. 用户对应的字符
29     u : 普通用户(user)
30     g : 组用户(group)
31     o : 其他用户(other)
32     a : 所有的用户(all)
33
34 4. 修改权限的字符
35     + : 给用户添加权限
36     - : 给用户删除权限
37     = : 修改用户的权限
38
39 5. 使用数字表示文件的权限
40     r: 可读(4)
41     w: 可写(2)
42     x: 可执行(1)
43     -: 没有权限(0)
44
45 6. 使用chmod命令, 修改文件的权限
46     方式1:
47         chmod 文件的权限 文件名/目录名
48             |--> "a/u/g/o" + "+/-/=" + "r/w/x"
49             |--> 从以上三个部分中, 每个部分选中一个或多个
50             | 字符组合成一个对应的字符串。
51         eg:
52         chmod a+rw hello.c ----> 给所有的用户添加可读可写的权限
53         chmod o-x hello.c ----> 删除其他用户的可执行权限
54         chmod g=rwx hello.c ----> 给组文件修改为可读, 可写, 可执行权限
55     方式2: 通过数字的方式修改文件的权限
56         chmod 0777 hello.c ----> 将所有的用户对文件的权限修改为
57             | | | | 可读, 可写, 可执行
58             | | | |----> 表示其他用户的权限: 可读(4) + 可写(2) + 可执行(1)
59             | | | |----> 表示组用户的权限: 可读(4) + 可写(2) + 可执行(1)

```

```

60      ||---> 表示普通用户的权限：可读(4) + 可写(2) + 可执行(1)
61      |----> 表示8进制数，可以省略不写
62
63      chmod 0764 hello.c
64      |||---> 表示其他用户的权限：可读(4) + 可写(0) + 可执行(0)
65      |||---> 表示组用户的权限：可读(4) + 可写(2) + 可执行(0)
66      ||---> 表示普通用户的权限：可读(4) + 可写(2) + 可执行(1)
67      |----> 表示8进制数，可以省略不写
68
69      有一堆桃子，有999，分成n堆，取任意的堆可以组合成1-999之间的所有的个数
70      1 2 4 8 16 32 64 128 256 488 .....

```

## 1.2 修改文件所属的用户和组的命令---> chown

```

1  -rw-rw-r-- 1 linux linux 0 4月 19 10:11 hello.c
2
3  sudo chown root:root hello.c ---> 修改hello.c文件所有的用户和组为root
4  sudo chown linux: hello.c ---> 修改hello.c文件所有的用户和组为linux
5  sudo chown root hello.c ---> 修改hello.c文件所有的用户为root
6  sudo chown :root hello.c ---> 修改hello.c文件所属的组为root

```

## 1.3 修改文件所属的组的命令--> chgrp

```

1  sudo chgrp root hello.c ---> 修改hello.c文件所属的组为root

```

# 2、链接文件

```

1  -rwxrwxr-- 1 linux linux 0 4月 19 10:11 hello.c
2      |---> 如果为普通文件，表示普通文件对应的硬链接文件的个数
3
4  drwxrwxr-x 2 linux linux 4096 4月 19 10:11 day02
5      |---> 如果为目录文件，表示此目录下的对应的子目录的个数

```

## 2.1 创建软链接文件--> ln

```

1  格式：
2      ln -s 绝对路径/被链接的文件    创建软链接文件的名字
3
4  注：
5      1. 可以对普通文件和目录创建对应的软链接文件；
6          linux@ubuntu:~/DC23032/c_advanced/day02$ ln -s
~/DC23032/c_advanced/day02/hello.c hello-s.c
7          linux@ubuntu:~/DC23032/c_advanced/day02$ ln -s
~/DC23032/c_advanced/day02/work/ work-s
8
9      2. 软链接文件对应的文件的类型为"l"；

```

```
10      lrwxrwxrwx 1 linux linux  44 4月  19 10:35 hello-s.c ->
    /home/linux/DC23032/c_advanced/day02/hello.c
11      lrwxrwxrwx 1 linux linux  42 4月  19 10:35 work-s ->
    /home/linux/DC23032/c_advanced/day02/work/
12
13      3. windows系统是不支持软链接文件的，因此不可以将软链接文件
14      存到windows的共享文件夹中"/mnt/hgfs/share".
15      如果某个压缩文件中有软链接文件，也不可以在windows中进行解压缩。
16
17      4. 在对被链接的文件创建软链接文件时，推荐被链接的文件使用绝对路径，
18          不要使用相对路径。
19
20          如果被链接的文件使用的是绝对路径，当将软链接文件拷贝到其他目录下时，
21          软链接文件和被链接的文件，链接关系依然有效。
22
23          如果被链接的文件使用的是相对路径，当将软链接文件拷贝到其他目录下时，
24          软链接文件和被链接的文件，链接关系将失效。
25
26      5. 当将被链接文件删除之后，软链接文件将失效，
27          当再次将被链接文件重新创建之后，软链接文件会重新生效
```

1 练习题：

2 1. 给windows的共享文件夹，在ubuntu的家目录下创建一个软链接文件。

3 `ln -s /mnt/hgfs/share ~/share`

4

5 好处：操作家目录下的share软链接文件，就相当于操作windows的共享文件夹。

6

7 2. 当操作的路径特别长时，可以对其创建一个软链接文件。

## 2.2 硬链接文件

1 格式：

2 `ln 被链接的文件 创建的硬链接文件的名字`

3

4 注：

5 1> 只可以对普通文件创建硬链接文件，不可以对目录创建硬链接文件。

6 `linux@ubuntu:~/day02$ ln hello.c hello-h.c`

7 `linux@ubuntu:~/day02$ ln work work-h`

8 `ln: work: 不允许将硬链接指向目录`

9

10 2> 被创建的硬链接文件的类型为“-”，创建完成硬链接文件之后，  
11 普通文件的硬链接的个数会增加。

12 `-rw-rw-r-- 2 linux linux 0 4月 19 10:46 hello.c`

13 `-rw-rw-r-- 2 linux linux 0 4月 19 10:46 hello-h.c`

14

15 3> 被链接的文件和创建的硬链接文件操作的是同一块存储空间。

16 验证操作的是同一块存储空间：修改任何一个文件，另外一个文件也会发生变化。

17

18 4> 被链接的文件和创建的硬链接文件的inode号是一致的，

19 使用`ls -li`命令可以查看文件的inode号。

20 `linux@ubuntu:~/day02$ ls -li`

21 `1835985 hello.c 1835985 hello-h.c`

22

23 5> 不管删除被链接的文件还是删除创建的硬链接文件，

- 24 | 存储空间的数据不会被删除，只是文件对应的硬链接数减1，
- 25 | 只有文件对应的硬链接数为0时，硬盘存储空间的数据才会被删除。

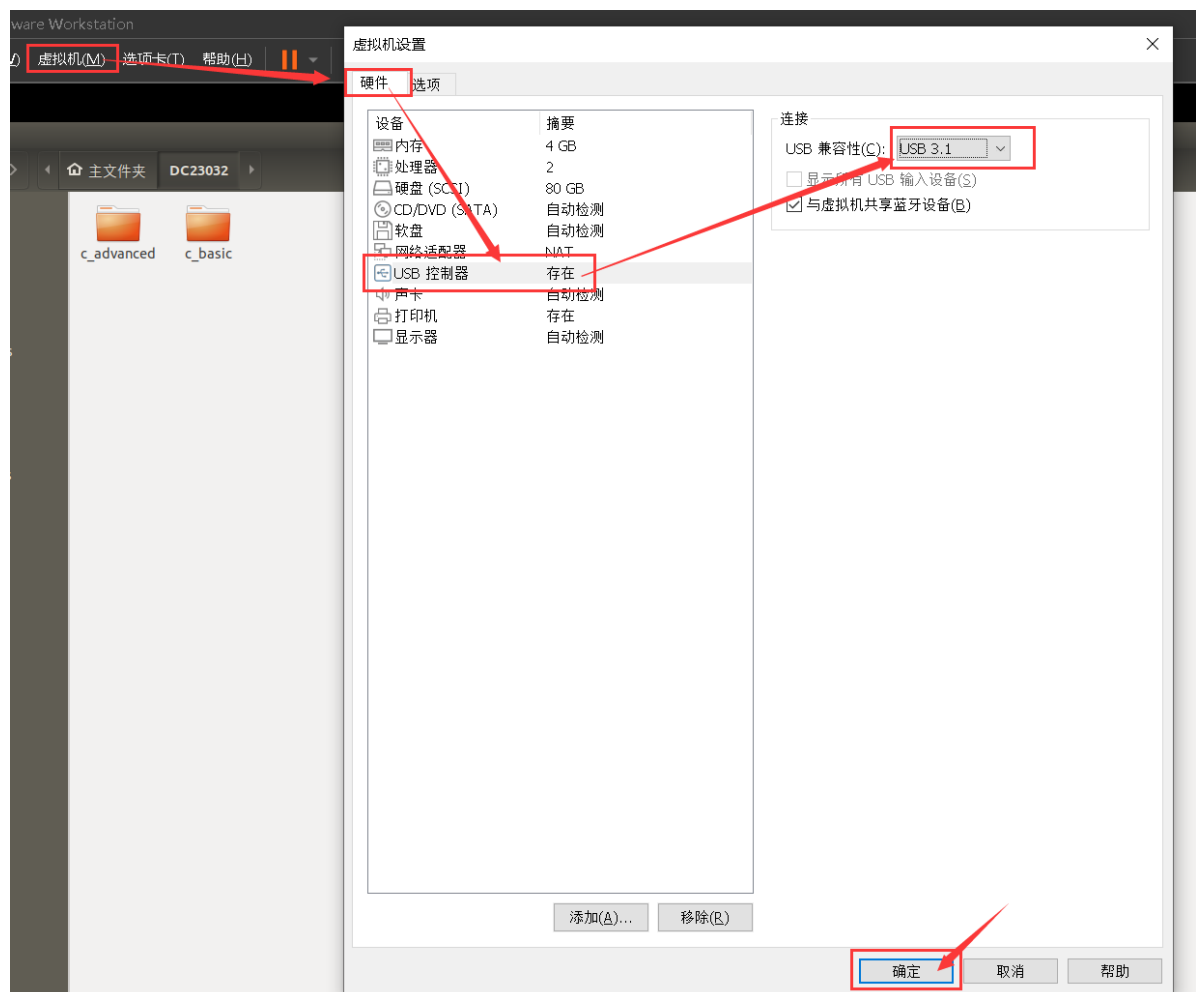
## 3、开关机相关的命令

```
1 关机命令：
2      sudo shutdown 15:10 ---> 15:10分关机
3      sudo shutdown +10 ---> 10分钟之后关机
4      sudo shutdown now ---> 立即关机
5
6 重启命令：
7      sudo reboot 15:10 ---> 15:10分重启
8      sudo reboot +10 ---> 10分钟之后重启
9      sudo reboot now ---> 立即重启
```

## 4、磁盘操作相关的命令

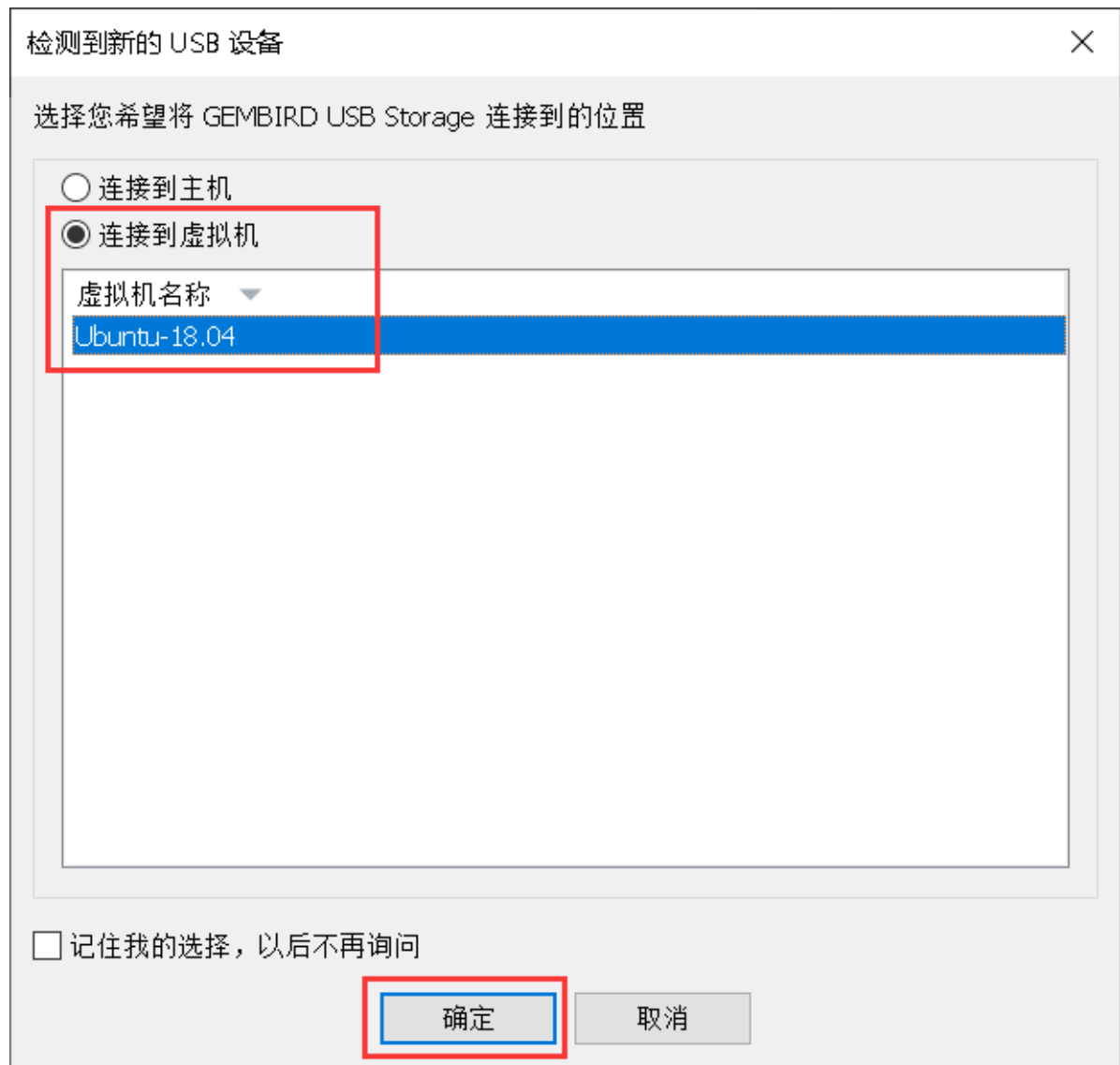
### 4.1 关于VMware软件的配置

#### 4.1.1 配置VMware软件支持usb3.1接口



## 4.1.2 让u盘被ubuntu系统识别

1> 方式1：将u盘插到电脑的USB接口，此时VMware软件会弹框，选中u盘被连接到ubuntu还是windows。



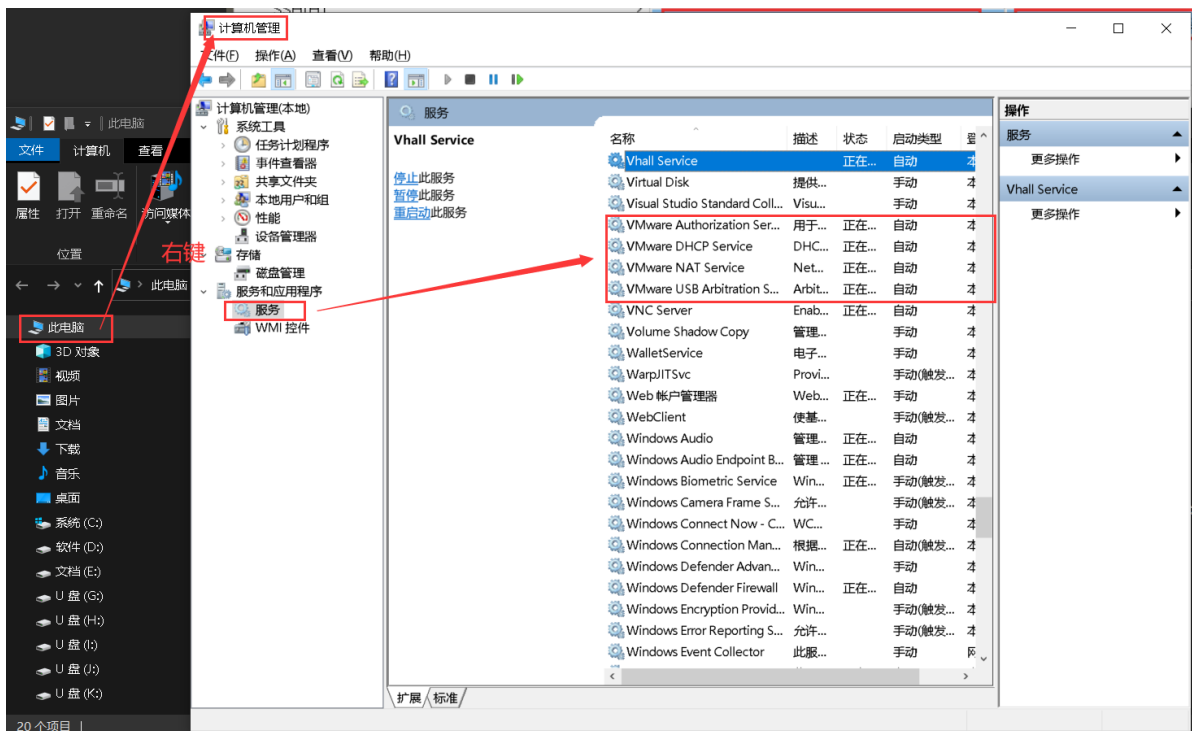
2> 方式2：通过虚拟机-->可移动设备--> 对应u盘名字的可移动设备--> 选择连接



3> 问题：部分人的电脑的VMware软件，在虚拟机->可移动设备->没有找到对应的磁盘名字，

原因：VMware软件的usb后台服务没有开启。

解决办法：ubuntu系统关机，在后台中打开usb服务。



### 4.1.3 查看u盘是否被ubuntu系统识别

```

1  将磁盘被ubuntu识别时，每个磁盘都会在/dev/目录下创建对应的设备文件。
2
3  可以使用以下命令查看磁盘是否被识别成功：
4  ls /dev/sd* -l
5
6  brw-rw---- 1 root disk 8,  0 4月  13 09:22 /dev/sda
7  brw-rw---- 1 root disk 8,  1 4月  13 09:23 /dev/sda1
8      ---> ubuntu系统对应分区，不要操作
9  brw-rw---- 1 root disk 8, 16 4月  19 11:46 /dev/sdb
10 brw-rw---- 1 root disk 8, 17 4月  19 11:46 /dev/sdb1
11 brw-rw---- 1 root disk 8, 18 4月  19 11:46 /dev/sdb2
12 brw-rw---- 1 root disk 8, 19 4月  19 11:46 /dev/sdb3
13 brw-rw---- 1 root disk 8, 20 4月  19 11:46 /dev/sdb4
14 brw-rw---- 1 root disk 8, 21 4月  19 11:46 /dev/sdb5
15      ---> 外接的u盘对应的分区，可以操作
16
17 块设备操作的特点：
18      块设备的访问是按照块进行读写操作，一块是512字节。
19      块设备文件的类型为“b”。
20
21 块设备文件的命令的方式：
22 /dev/sd[a-z][1-9]
23     |      |----> 块设备文件对应的分区变化
24     |----> 表示哪个块设备

```

## 4.2 查看磁盘的使用率的命令 ---> df

```

1 linux@ubuntu:~$ df /dev/sd*
2 文件系统          1k-块      已用      可用  已用% 挂载点
3 /dev/sda1        81985944 18120144 59655216   24% /
4 /dev/sdb1          236          2      235    1% /media/linux/44B5-219B
5 /dev/sdb3         2016          2      2015    1% /media/linux/EABA-63C9
6 /dev/sdb4         59348       7456      47308   14% /media/linux/7ecd1fc8-6aa0-
   4dc4-8690-86ca59144ef5
7 /dev/sdb5        15184344       32 15184312    1% /media/linux/1CBC-E891
8

```

## 4.3 磁盘删除分区/重新分区的命令-->fdisk

```

1 命令(输入 m 获取帮助): m
2
3 帮助:
4
5 常规
6   d   删除分区
7   F   列出未分区的空闲区
8   l   列出已知分区类型
9   n   添加新分区
10  p   打印分区表
11  t   更改分区类型
12  v   检查分区表
13  i   打印某个分区的相关信息
14
15 保存并退出
16   w   将分区表写入磁盘并退出
17   q   退出而不保存更改
18
19

```

```

1 将磁盘的原有的分区删除:
2 linux@ubuntu:~$ sudo fdisk /dev/sdb    ---> 分区的命令
3
4 欢迎使用 fdisk (util-linux 2.31.1)。
5 更改将停留在内存中，直到您决定将更改写入磁盘。
6 使用写入命令前请三思。
7
8
9 命令(输入 m 获取帮助): d    ---> 删除分区
10 分区号 (1-5, 默认 5):      ---> 删除分区的编号，使用默认编号即可
11
12 分区 5 已删除。
13
14 /* 其它的几个分区的删除省略 */
15
16 命令(输入 m 获取帮助): w
17
18 分区表已调整。
19 Failed to remove partition 1 from system: 设备或资源忙
20 Failed to remove partition 3 from system: 设备或资源忙
21 Failed to remove partition 4 from system: 设备或资源忙

```



```
22 Failed to remove partition 5 from system: 设备或资源忙
23
24 The kernel still uses the old partitions. The new table will be used at
the next reboot.
25 正在同步磁盘。
26
```

```
1 对磁盘进行分区:
2
3 linux@ubuntu:~$ sudo fdisk /dev/sdb ---> 分区命令
4
5 欢迎使用 fdisk (util-linux 2.31.1)。
6 更改将停留在内存中，直到您决定将更改写入磁盘。
7 使用写入命令前请三思。
8
9
10 命令(输入 m 获取帮助): n ---> 创建分区
11 分区号 (1-128, 默认 1): ---> 分区的编号，默认为1，直接回车
12 第一个扇区 (34-30537694, 默认 2048): ---> 起始块号，默认
13 上个扇区, +sectors 或 +size{K,M,G,T,P} (2048-30537694, 默认 30537694):
14 ---> 结束块号
15 创建了一个新分区 1, 类型为“Linux filesystem”, 大小为 14.6 GiB。
16
17 命令(输入 m 获取帮助): w ---> 保存退出
18 分区表已调整。
19 将调用 ioctl() 来重新读分区表。
20 正在同步磁盘。
21
22 linux@ubuntu:~$ ls /dev/sdb* -l ---> 查看分区是否成功
23 brw-rw---- 1 root disk 8, 16 4月 19 13:36 /dev/sdb
24 brw-rw---- 1 root disk 8, 17 4月 19 13:36 /dev/sdb1
25
```

## 4.4 对分区进行格式化--> mkfs.fat

```
1 linux@ubuntu:~$ sudo mkfs.fat /dev/sdb1 ---> 格式化磁盘为fat格式
2
3
```

## 4.5 磁盘挂载(mount)和取消挂载(umount)的命令

```
1 1. 在/mnt目录下创建udisk文件夹,作用将/dev/sdb1设备文件挂载到/mnt/udisk目录下
2   sudo mkdir /mnt/udisk
3
4 2. 使用挂载的命令将/dev/sdb1挂载到/mnt/udisk目录下
5   sudo mount /dev/sdb1 /mnt/udisk
6
7   拷贝文件到/mnt/udisk目录下时,就相当于将文件拷贝到磁盘的sdb1的分区
8   cp hello.c /mnt/udisk
9   cp hello.c /dev/sdb1 ---> 不可以
10 3. 将磁盘从/mnt/udisk目录下取消挂载
11   sudo umount /mnt/udisk
```

## 4.6 模拟一个磁盘，完成以上命令的练习

```
1 1. 使用dd命令创建一个文件，可以存储数据的单个文件
2   sudo dd if=/dev/zero of=mydisk bs=1M count=128
3
4   解释：
5   dd : 拷贝的命令
6   if : 输入文件，/dev/zero:表示0设备文件
7   of : 输出文件
8   bs : 大小
9   count : 个数
10
11 2. 对mydisk文件进行格式化
12   sudo mkfs.fat mydisk
13
14 3. 将mydisk挂载到/mnt/udisk目录下
15   sudo mount mydisk /mnt/udisk
16
17 4. 向mydisk中拷贝文件
18   此时向/mnt/udisk中拷贝文件，就相当于拷贝到mydisk文件中
19   sudo cp hello.c /mnt/udisk
20
21 5. 取消挂载
22   sudo umount /mnt/udisk
```

## 5、网络相关的概念及命令

### 5.1 网络相关的概念

#### 5.1.1 IP地址

```
1 IP地址：
2   IP地址是Internet中主机的唯一标识，
3   IP地址由网络号（Network ID）和主机号（Host ID）两部分组成。
4   网络号标识的是Internet上的一个子网，而主机号标识的是子网中的某台主机。
5   IP地址根据网络号和主机号的数量而分为A、B、C三类。
```

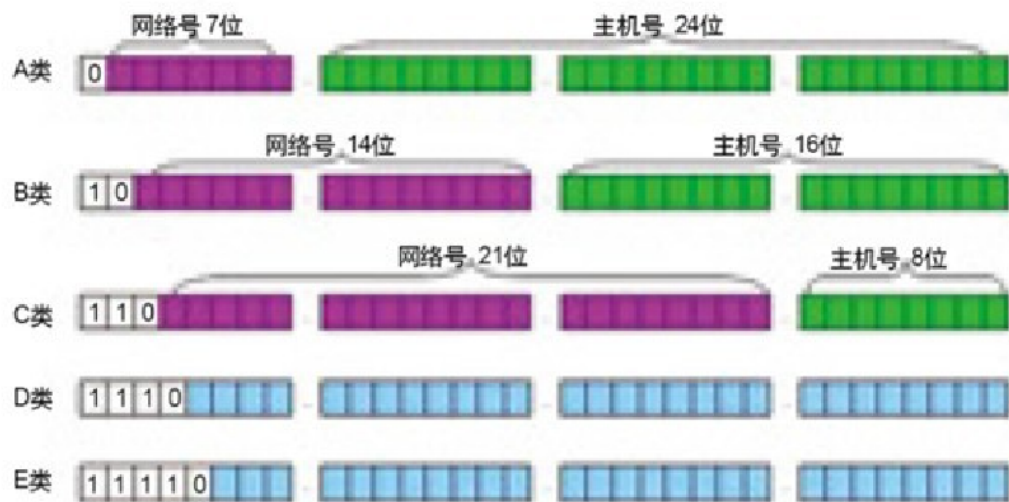


图1 IP地址的类别

- 1 IP地址为32位（IPv4）或者128位（IPv6）
- 2 每个数据包都必须携带目的IP地址和源IP地址，路由器依靠此信息为数据包选择路由。
- 3
- 4 表示形式：常用点分十进制形式，如202.38.64.10，
- 5 最后都会转换为一个32位的无符号整数。
- 6
- 7 IP地址的范围：
- 8 \*\*\*\*.\*\*\*\*.\*\*\*\*.\*\*\*\*.\*\*\*\*.\*\*\*\*.\*\*\*\*.\*\*\*\* --->这时一个二进制的数
- 9 IP地址的范围：0.0.0.0 ~ 255.255.255.255
- 10
- 11 A类网络IP地址的范围：
- 12 0000 0000.0000 0000.0000 0000.0000 0000 ~
- 13 0111 1111.1111 1111.1111 1111.1111 1111
- 14 0.0.0.0 ~ 127.255.255.255
- 15 A类网络主要用于军队，政府使用
- 16 B类网络IP地址的范围：
- 17 1000 0000.0000 0000.0000 0000.0000 0000 ~
- 18 1011 1111.1111 1111.1111 1111.1111 1111
- 19 128.0.0.0 ~ 191.255.255.255
- 20 B类网络主要用于：政府，学校，大型的企业
- 21
- 22 C类网络IP地址的范围：
- 23 1100 0000.0000 0000.0000 0000.0000 0000 ~
- 24 1101 1111.1111 1111.1111 1111.1111 1111
- 25 192.0.0.0 ~ 223.255.255.255
- 26 C类网络主要用于：个人，中小型企业
- 27
- 28 D类
- 29 224.0.0.0 - 239.255.255.255 组播地址
- 30 E类
- 31 240.0.0.0 - 255.255.255.255 保留测试
- 32
- 33 192.168.1.81：
- 34 网络号：192.168.1.0：在同一个局域网下的每台网络设备的网络号固定。
- 35 主机号：81：在同一个局域网下的每台网络设备的主机号必须唯一。

## 5.1.2 网关

- 1 每一个网段(网络号)下面 : 192.168.1.x
- 2 192.168.1.1 表示网关
- 3
- 4 每个网段(网络号)下的, 主机号为1的表示网关,

## 5.1.3 子网掩码

- 1 A类网络的子网掩码为: 255.0.0.0
- 2 B类网络的子网掩码为: 255.255.0.0
- 3 C类网络的子网掩码为: 255.255.255.0
- 4
- 5 通过子网掩码可以获取网络的网络号:
- 6 网络号 = IP地址 & 子网掩码
- 7
- 8  $192.168.1.81 \& 255.255.255.0 = 192.168.1.0$
- 9
- 10 子网掩码的另外一种写法:
- 11 在ubuntu系统中写子网掩码的写法: 24(表示24个1)

## 5.1.4 DNS服务

- 1 DNS : 域名解析器
- 2 登录某个网站一般使用的是域名, 但是真正访问网站还是通过IP地址实现的。
- 3 因此需要使用DNS域名解析器将域名解析为对应的IP地址。
- 4
- 5 免费的域名解析器:
- 6 8.8.8.8
- 7 114.114.114.114

Internet 协议版本 4 (TCP/IPv4) 属性

常规

如果网络支持此功能，则可以获取自动指派的 IP 设置。否则，你需要从网络系统管理员处获得适当的 IP 设置。

☐ 自动获得 IP 地址(O)

☒ 使用下面的 IP 地址(S):

IP 地址(I): 192 . 168 . 1 . 81

子网掩码(U): 255 . 255 . 255 . 0

默认网关(D): 192 . 168 . 1 . 1

☐ 自动获得 DNS 服务器地址(B)

☒ 使用下面的 DNS 服务器地址(E):

首选 DNS 服务器(P): 8 . 8 . 8 . 8

备用 DNS 服务器(A): 114 . 114 . 114 . 114

☐ 退出时验证设置(L)

高级(V)...

确定 取消

## 5.2 查看windows系统的IP地址->ipconfig

- 1 | win + r键--> 输入cmd-->弹出终端窗口，输入ipconfig查看IP地址。
- 2 |

```
C:\Users\7504B>ipconfig

Windows IP 配置

以太网适配器 以太网:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

以太网适配器 以太网 3:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址. . . . . : fe80::cccc:81e7:66f1:bf10%21
    IPv4 地址 . . . . . : 192.168.1.81
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 192.168.1.1
```

## 5.3 查看ubuntu系统的IP地址-》ifconfig

```
linux@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.74.133 netmask 255.255.255.0 broadcast 192.168.74.255
    inet6 fe80::898f:867f:fd36:3fc6 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:a0:a6:f0 txqueuelen 1000 (以太网)
    RX packets 400181 bytes 490771953 (490.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 41735 bytes 3236387 (3.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## 5.4 测试网络的命令->ping

```
linux@ubuntu:~$ ping www.baidu.com
PING www.a.shifen.com (39.156.66.14) 56(84) bytes of data:
64 bytes from 39.156.66.14 (39.156.66.14): icmp_seq=1 ttl=128 time=7.26 ms
64 bytes from 39.156.66.14 (39.156.66.14): icmp_seq=2 ttl=128 time=6.42 ms
^C
--- www.a.shifen.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 6.424/6.846/7.268/0.422 ms
linux@ubuntu:~$
```

## 5.5 重启ubuntu系统网络服务的命令

- 1     sudo service NetworkManager stop   ---> 停止网络服务
- 2     sudo service NetworkManager start   ---> 启动网络服务
- 3     sudo service NetworkManager restart ---> 重启网络服务
- 4

# ## shell编程

## 1、shell相关的概念

```
1 1. 编程语言的分类
2   编译型的语言--> 需要使用编译器: gcc/g++ ---> C/C++
3   解释型的语言--> 需要使用解析器: bash/python3.8 ---> shell脚本/python
4
5
6 2. shell脚本使用的解析器
7   bash      ---> ubuntu系统中使用的就是bash解析器
8   sh        ---> 开发板中启动linux系统之后, 使用的sh解析器
9   ksh
10  csh
11
12   在终端输入shell命令之后, 由bash解析器解释shell命令,
13   最终将命令的执行结果在终端进行回显。
```

## 2、第一个shell脚本文件

```
1 1. 创建一个01first.sh文件
2   touch 01first.sh
3
4   脚本文件的后缀一般为.sh
5
6 2. 打开脚本文件, 添加以下内容, 输出hello world
7   #!/bin/bash
8
9   # 1. 在shell脚本文件中, 单行注释为 #
10  # 2. shell脚本文件的第一行添加: #!/bin/bash 表示使用bash解析器
11
12  # 3. echo : 输出的命令
13  echo "hello world"
14
15 3. 执行脚本文件
16   方式1: ./脚本文件名
17   要求: 脚本文件必须具有可执行的权限, 因此使用chmod添加可执行的权限
18   chmod 777 脚本文件名
19   chmod a+x 脚本文件名
20
21   特点: 脚本文件必须具有可执行的权限,
22   执行脚本文件会重新打开一个新的终端, 然后执行脚本文件中的命令,
23   最终将脚本文件的执行结果显示到当前的终端之上。
24
25   方式2: bash ./脚本文件名
26
27   特点: 脚本文件可以不具有可执行的权限
28   执行脚本文件会重新打开一个新的终端, 然后执行脚本文件中的命令,
29   最终将脚本文件的执行结果显示到当前的终端之上。
30
31   方式3: source ./脚本文件名
32
33   特点: 脚本文件可以不具有可执行的权限
34   执行脚本文件是在当前终端执行脚本文件中的命令,
35   不需要重新打开一个新的终端
```

1 练习题: 编写脚本文件, 实现以下功能:

```

2  #!/bin/bash
3  # 1.在自己的用户主目录下新建两个子目录subdir1 subdir2
4  cd ~
5  mkdir subdir1 subdir2
6
7  # 2.将/etc/passwd文件拷贝到subdir1 ,将/etc/group 拷贝到subdir2
8  cp /etc/passwd  subdir1
9  cp /etc/group   subdir2
10
11 # 3.将subdir2重命名为subdir
12 mv subdir2  subdir
13
14 # 4.对subdir1进行打包并且压缩成xz格式
15 tar -Jvcf subdir1.tar.xz  subdir1
16
17 # 5.将打包后的xz格式文件拷贝到subdir目录下
18 cp subdir1.tar.xz  subdir
19
20 # 6.解压subdir目录下的压缩格式文件
21 cd subdir
22 tar -vxf subdir1.tar.xz
23 cd ../
24 # 7.查看subdir目录下所有的文件
25 ls subdir
26

```

## 3、 shell脚本文件中变量的使用

### 3.1 shell脚本文件中定义变量

```

1  1. 定义变量的格式
2      变量名=初始值
3      变量名='初始值'
4      变量名="初始值"
5
6  2. 注
7      1> 在shell中定义的变量没有数据类型，变量的值都属于字符串
8          比如：num=12345678      ---> 字符串
9
10     2> 变量名的书写规则和C语言一致。
11
12     3> 在shell中定义变量时，等号两边不允许有空格
13         name = 'zhangsan'      # 错误
14
15     4> 以上三种定义变量的格式，一般情况下可以通用，没有区别。
16         str1=hello world
17         str1='hello world'
18         str1="hello world"  # 一样的没有区别
19
20     5> 如果变量的初始值之间有空格时，需要使用''或者双引号括起来
21         str1=hello world      # error
22         str1='hello world'    # ok
23         str1="hello world"    # ok
24
25     6> 如果使用某个变量的值对另外一个变量赋值时，使用""括起来

```



```

25     str1=hello
26     str2=${str1}          # ok
27     str3=${str1}world    # ok
28     str4=hello${str1}    # ok
29
30     str5=${str1} world   # error
31     str6=hello ${str1}  # error
32
33     str5='${str1} world' # ok   单引号中的变量不会被解析
34     str6="hello ${str1}" # ok   双引号中的变量会被解析

```

## 3.2 shell中定义的变量的引用

```

1  ${变量名}或者$变量名
2
3  对于初学者来说推荐使用${变量名}
4
5  如果不加{}可能会出现以下问题:
6      str1=hello          # ok
7      str2=${str1}world   # ok
8      str3=hello${str1}   # ok
9      str4=$str1world     # 有问题: 将str1world整体当成变量名
10     str3=hello$str1      # ok

```

```

1  #!/bin/bash
2
3  echo "--- 1. 定义变量的三种格式 ---"
4  name=zhangsan
5  sex='man'
6  age="18"
7  echo "name = ${name}"
8  echo sex = $sex
9  echo "age = $age"
10
11 echo "--- 2. 定义变量时, 等号两边不允许有空格---"
12 # str = helloworld    # error
13
14
15 echo "--- 3. 定义变量时, 初始值有空格, 需要使用' '或者\"\"括起来"
16 # str1=hello world    # 错误
17 str2='hello world'
18 str3="hello world"
19 echo "str1 = $str1"
20 echo "str2 = $str2"
21 echo "str3 = $str3"
22
23 echo "--- 4. 使用某个变量给另外一个变量赋值时, 使用\"\"括起来"
24 firstname=zhang
25 lastname=san
26 name=$firstname$lastname # ok
27 echo "name = $name"
28
29
30 #name=$firstname $lastname # error

```

```

31 name='$firstname$lastname' # ok
32 echo "name = $name 单引号中的变量名不解析"
33 name="$firstname$lastname" # ok
34 echo "name = $name 双引号中的变量名解析"
35

```

## 3.3 shell中位置变量的使用

```

1  执行脚本文件时，可以给脚本文件传递参数，在脚本文件中通过位置变量
2  获取执行脚本文件传递的参数。
3  $0      --> 脚本文件的名字
4  $1      --> 执行脚本文件时传递的第1个参数
5  $2      --> 执行脚本文件时传递的第2个参数
6  $3      --> 执行脚本文件时传递的第3个参数
7  .....
8  $9      --> 执行脚本文件时传递的第9个参数
9  $10     --> 执行脚本文件时传递的第10个参数和字符串0结合
10 ${10}   --> 执行脚本文件时传递的第10个参数
11 ${11}   --> 执行脚本文件时传递的第11个参数
12 ${n}    --> 执行脚本文件时传递的第n个参数,n是一个大于等于1的整数
13
14 $*      --> 执行脚本文件时传递的所有的参数，不包括脚本文件名
15 $@      --> 执行脚本文件时传递的所有的参数，不包括脚本文件名
16 $#      --> 执行脚本文件时传递的所有的参数个数之和，不包括脚本文件名
17

```

```

1
2 echo '$0 = '$0
3 echo '$1 = '$1
4 echo '$2 = '$2
5 echo '$3 = '$3
6 echo '$4 = '$4
7 echo '$5 = '$5
8 echo '$6 = '$6
9 echo '$7 = '$7
10 echo '$8 = '$8
11 echo '$9 = '$9
12 echo '$10 = '$10
13 echo '${10} = '${10}
14 echo '$* = '$*
15 echo '$@ = '$@
16 echo '$# = '$#
17
18 -----
19 bash 04variable.sh aa bb cc dd ee 11 22 33 44 55

```

## 3.4 修饰变量的关键字

```
1 1. readonly : 修饰的变量只读
2 2. unset : 取消变量的定义
3 3. local : 定义局部变量
4 在shell中默认所有的变量都是全局变量，即使在函数中定义的变量
5 默认也是全局变量，只有在函数中定义的变量，
6 使用local修饰之后才能变成局部变量。
```

```
1
2 #!/bin/bash
3
4 echo "---1. 定义只读变量 ---"
5 readonly name="zhangsan"
6 echo "name = $name"
7 name="lisi"      # 修改只读变量
8
9 echo "--- 2. 取消变量的定义 ---"
10 sex='man'
11 echo "sex = $sex"
12 unset sex
13 echo "sex = $sex" # 空值
14
15 echo "--- 3. 定义局部变量 ---"
16
17 function set_init_age()
18 {
19     global_age=18      # 全局变量
20     local local_age=20 # 局部变量
21 }
22
23 set_init_age          # shell中函数的调用
24
25 echo "global_age = $global_age"
26
27 echo "local_age = $local_age"
28
```

## 3.5 命令置换符

```
1 将shell命令的执行的结果赋值给某个变量，需要使用命令置换符。
2
3 变量名=`shell命令`
4     反向单引号： ~键上边的那个反向单引号
5
6 或者
7
8 变量名=$(shell命令)
```

```
1  #!/bin/bash
2
3
4  currentDir=$(pwd)
5
6  HostName=`hostname`
7
8  echo "currentDir = $currentDir"
9  echo "HostName = $HostName"
10
```

## 4、作业

---

```
1 第4套C语言的试卷
2
3 配置vscode软件:
4 文档: 01.vscode配置.md
5 链接: http://note.youdao.com/noteshare?id=081e0752dec5f61c498f97e71be325b6&sub=4D4C2BD0BB3E4A0FB7C1C6A5B507EF7B
```

## 5、明天的授课内容

---

```
1 1. shell中的数组
2 2. shell中的算数运算
3 3. 分支语句
4 4. 循环语句
```

