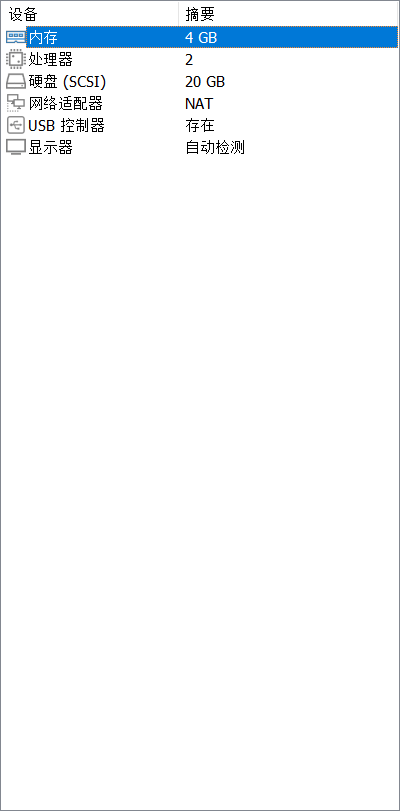
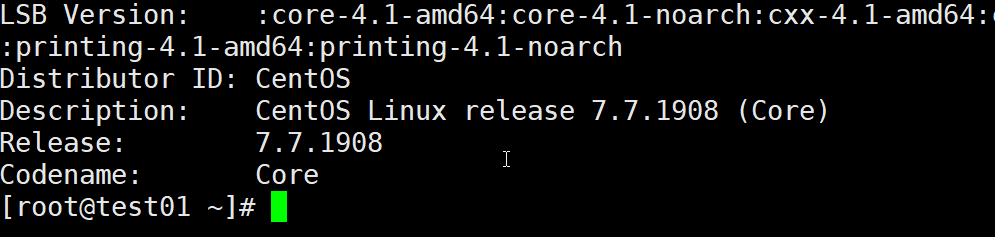
**零基础学Linux**

1. **通过U盘安装Linux，有条件的情况下尝试刻光盘安装。**

**学习和了解Linux的安装过程，发现问题寻找答案的能力。**

条件所限，通过VMware16安装虚拟机的形式安装linux，本次安装选择的是CentOS7，配置如下，通过xshell以ssh的形式连接虚拟机进行操作。





1. **在机器上挂一个空硬盘，机器重启后依然可见硬盘。（选学）**

**在一块原始硬盘上做出5块空间相同的小盘。**

**用这个5个小盘，做raid0，raid1，raid5，raid6。**

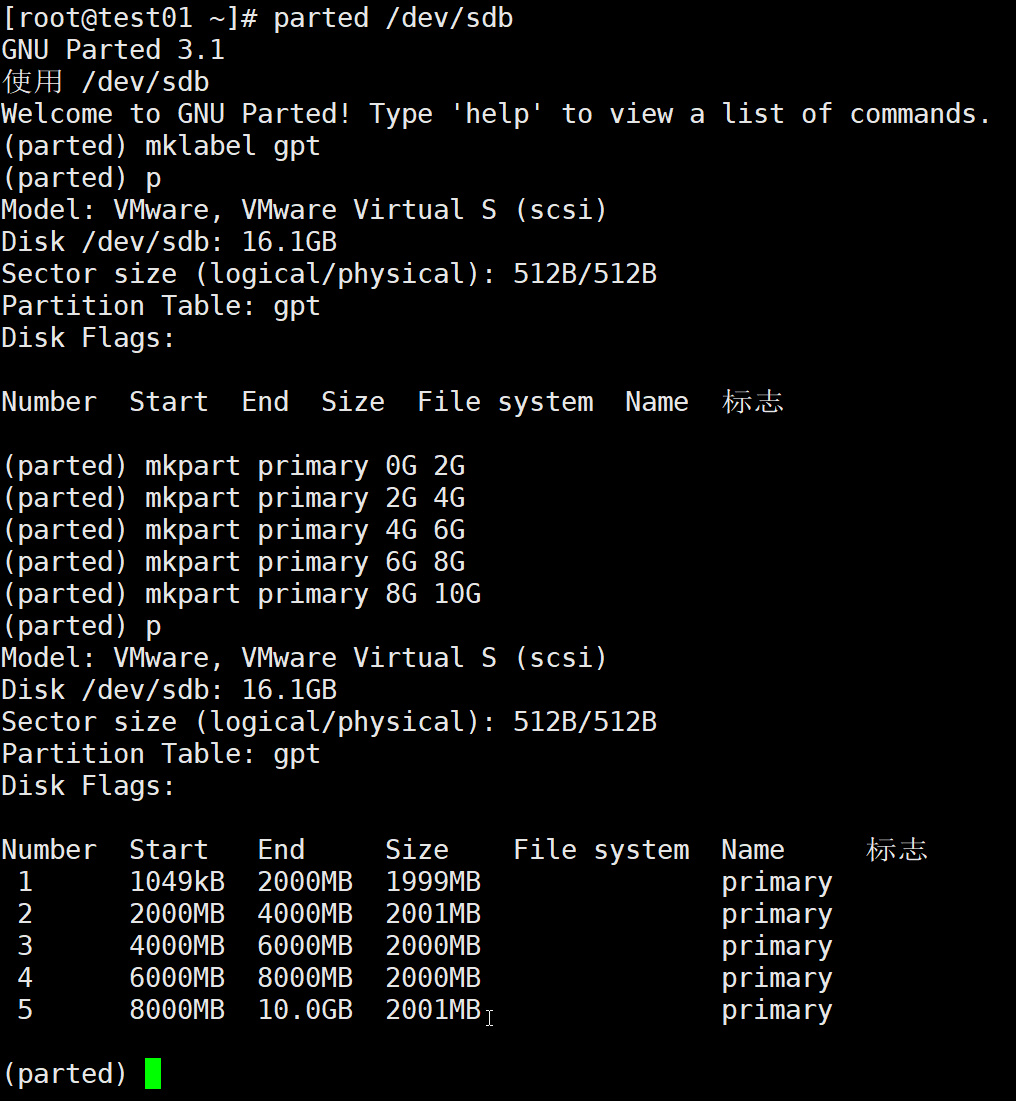
**完成模拟一块坏盘，raid恢复的过程**

**mdadm stop raid后，重新恢复raid**

**往raid里面增加新盘**

1.通过修改/etc/fstab，添加挂载情况，添加完后执行mount -a即可重启后依然可见硬盘。

2.通过虚拟机添加一块15G硬盘，通过gpt分区表进行分区，5个2G大小的分区。



通过mkfs.ext4 /分区名 对其格式化，

1）做raid0

[root@test01 ~]#mdadm --create /dev/md0 --level=0 --raid-devices=5 -a yes /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4 /dev/sdb5

mdadm: chunk size defaults to 512K

mdadm: /dev/sdb1 appears to contain an ext2fs file system

size=1951744K mtime=Sun May 8 17:27:06 2022

mdadm: /dev/sdb2 appears to contain an ext2fs file system

size=1953792K mtime=Sun May 8 17:27:10 2022

mdadm: /dev/sdb3 appears to contain an ext2fs file system

size=1952768K mtime=Sun May 8 17:27:14 2022

mdadm: /dev/sdb4 appears to contain an ext2fs file system

size=1952768K mtime=Sun May 8 17:27:17 2022

mdadm: /dev/sdb5 appears to contain an ext2fs file system

size=1953792K mtime=Thu Jan 1 08:00:00 1970

Continue creating array? y

mdadm: Defaulting to version 1.2 metadata

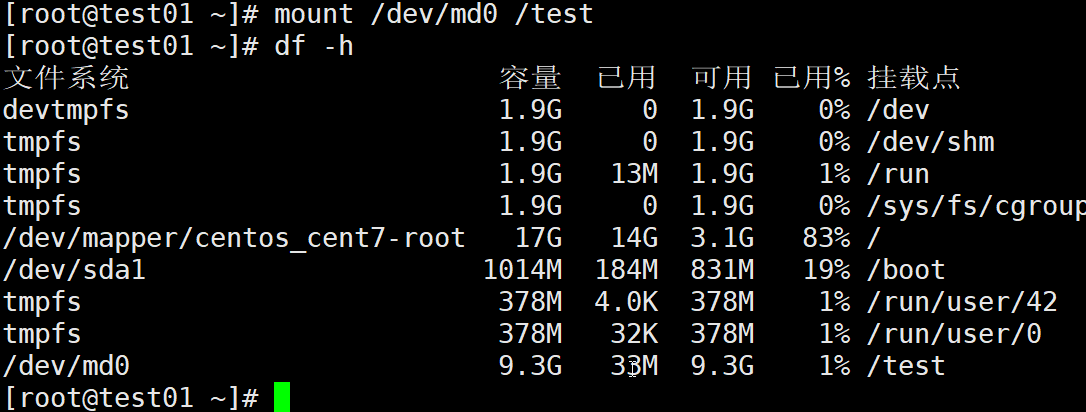
mdadm: array /dev/md0 started.

2） 将raid0 格式化文件系统为xfs

mkfs.xfs /dev/md0

1. 建立/test 用于挂载

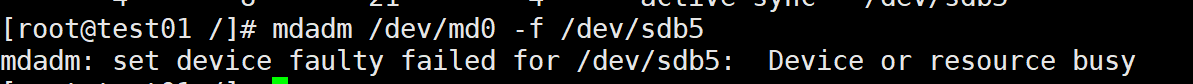
mount /dev/md0 /test



4）检查raid0详细信息

[root@test01 ~]# mdadm -D /dev/md0

5）移除一块磁盘，raid0 不提供冗余，故无法移除。



1. 卸载raid

umount /test 取消挂载

mdadm -S /dev/md0 停止服务

mdadm --misc --zero-superblock /dev/sdb1 卸载磁盘信息

mdadm --misc --zero-superblock /dev/sdb2

mdadm --misc --zero-superblock /dev/sdb3

mdadm --misc --zero-superblock /dev/sdb4

mdadm --misc --zero-superblock /dev/sdb5

rm /etc/mdadm.conf 删除配置文件

清除开机自启动配置 /etc/fstab 文件中的raid信息

1. 创建raid 1、 raid5、raid 6

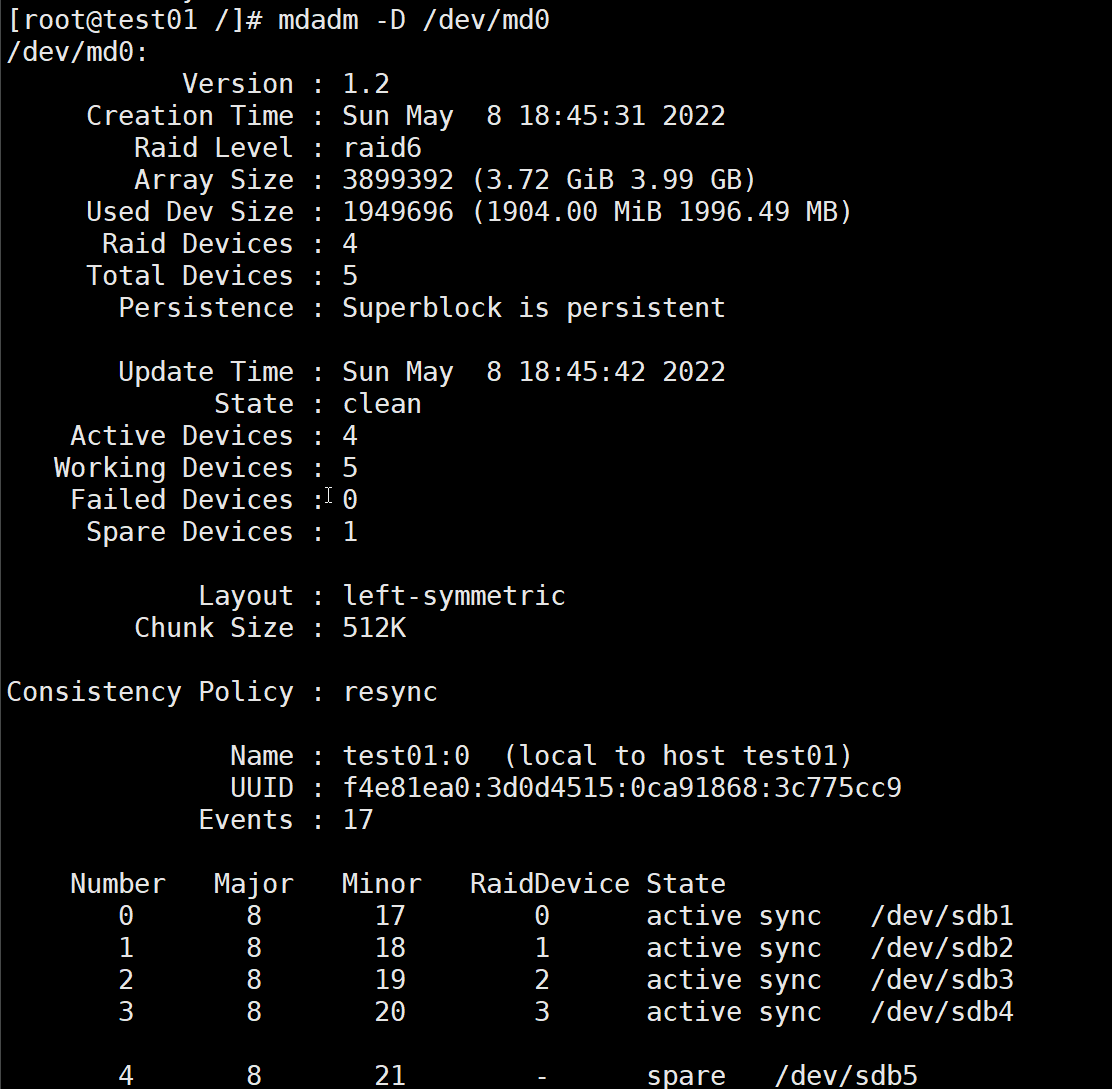
mdadm -Cv /dev/md0 -n 5 -l 1 /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4 /dev/sdb5

mdadm -Cv /dev/md0 -n 4 -l 5-x 1 /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4 /dev/sdb5

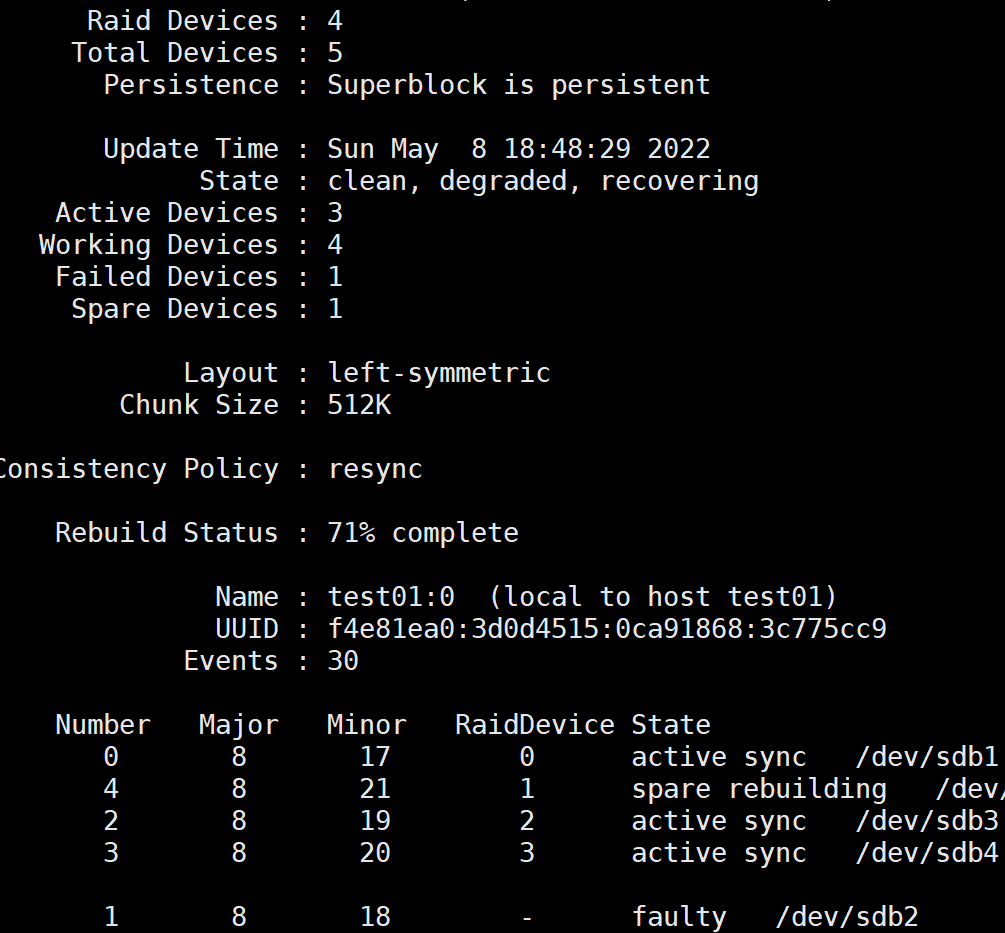
mdadm -Cv /dev/md0 -n 4 -l 6-x 1 /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4 /dev/sdb5

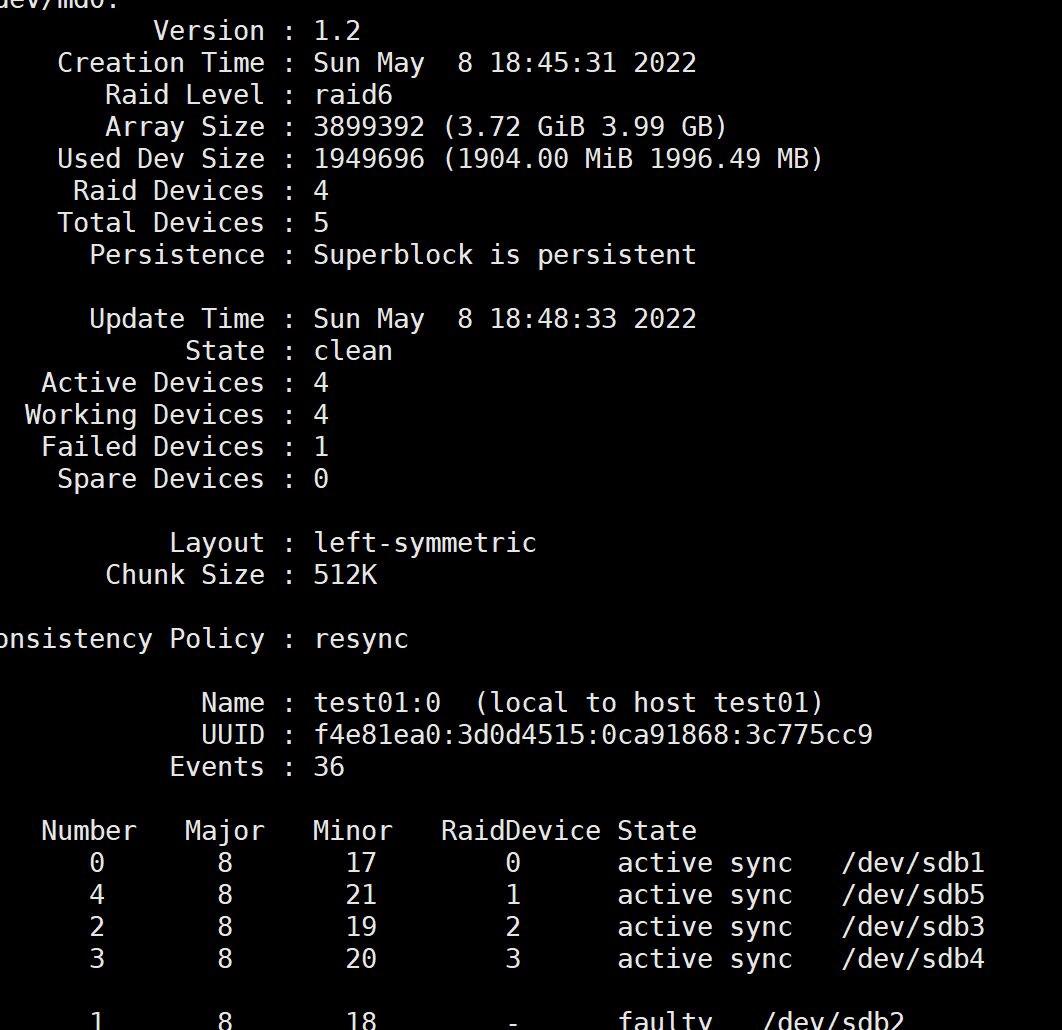
后格式化文件系统进行挂载即可使用。

1. 模拟坏盘并恢复（以raid6为例）



mdadm /dev/md0 -f /dev/sdb2



自动恢复数据。

5.重启raid （以raid0为例）

1）创建raid配置文件

echo DEVICE /dev/sdb[1-5] > /etc/mdadm.conf

1. 扫描raid信息，追加到配置文件中

mdadm -Ds >> /etc/mdadm.conf

3）取消raid的挂载

umount /test

1. 停止raid

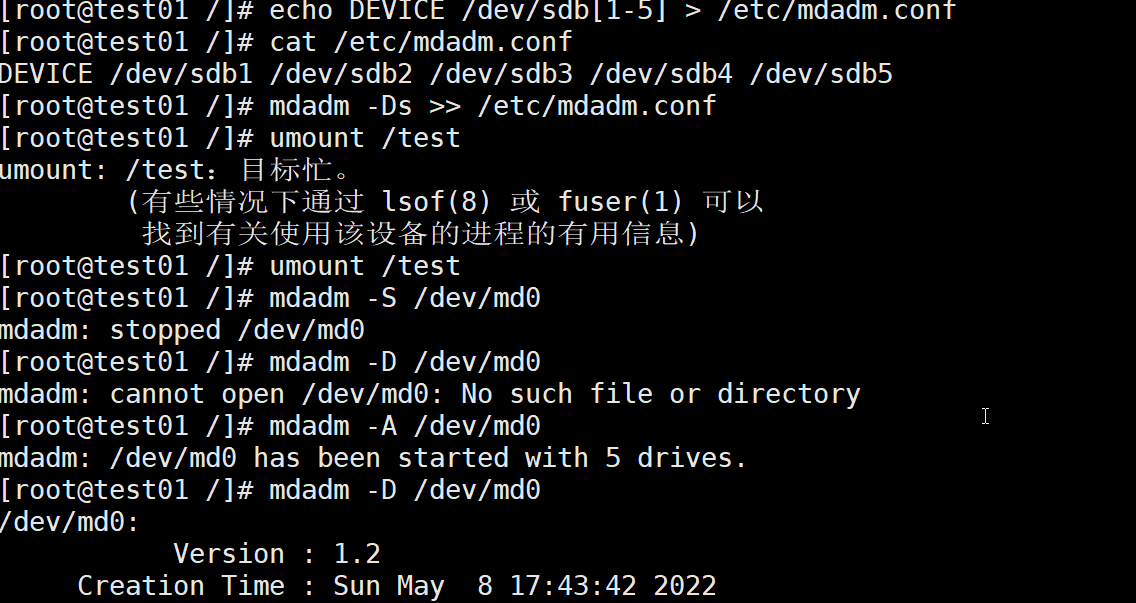
mdadm -S /dev/md0

1. 重启raid

mdadm -A /dev/md0

6）查看raid信息

mdadm -D /dev/md0



1. 添加新盘（以3中模拟损坏的sdb2为例）
2. 取消挂载 umount /test
3. 重启机器 reboot
4. 重新加入新盘 mdadm /dev/md0 -a /dev/sdb2（恢复的太快了没截到图片）

3）学习常用命令cp, ls, mv,pwd, cat, crontab, cut, df, grep, head, history, ifconfig, kill, locate,ln, mkdir, netstat, nohup,ping, ps, rm, rsync, sort, tail, telnet, touch, uptime,wc 等等，了解和记录每个命令的基本功能

cp：复制文件或目录

ls ：列出指定目录下的内容（文件及目录）

mv：移动文件或目录，对文件或目录改名

pwd：显示当前目录

cat：查看文件内容

crontab：定期检查是否有要执行的工作，有就自动执行

cut：从文件的每一行剪切字节、字符、字段至标准输出

df：显示系统磁盘使用情况统计

grep：过滤查找符合条件的字符串

head：查看文件开头部分内容 -n 指定行数

history：显示历史使用过的命令

ifconfig：显示或设置网络设备

kill：杀死执行中的程序

locate：在保存文档及目录名称的数据库内查找符合条件的文档

ln：为某个文件建立链接

mkdir：创建目录

netstat：显示网络状态

nohup：在后台执行命令

ping：检测主机连接情况

ps：显示当前进程状态

rm：删除文件或目录

rsync：本地主机与远程主机数据同步

sort：将文本文件内容排序

tail：查看文件末尾部分内容 -f循环读取用于日志查看

telnet：远程登陆

touch：创建文件，或改变文件或目录的时间

uptime：查看系统运行时长与平均负载

wc：计算文件的行数、字数、字节数

4）压缩数据命令强化学习，tar、7za、zip、bz2 等

自动生成一个1GB随机纯文本文件，内容是中文或者英文。

采用以上命令压缩，比较压缩率，压缩和解压时间，完成一个表格。

生成数据

base64 /dev/urandom | head -c 1G > textrandom.txt

压缩数据

tar -xzvf tar01.tar.gz textrandom.txt

time 7za a 7z01.7z textrandom.txt

time zip tar01.zip textrandom.txt

time tar -cjvf tar01.tar.bz2 textrandom.txt

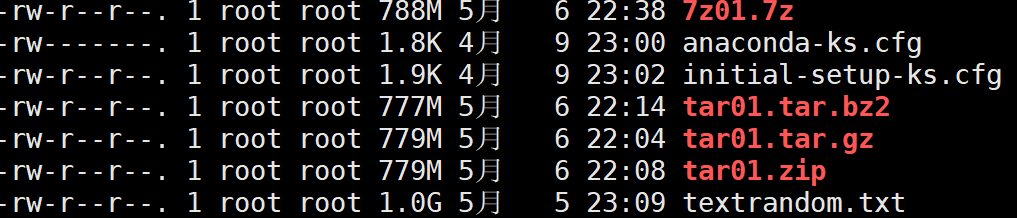
解压缩

tar -xzvf tar01.tar.gz

time 7z e 7z01.7z

time unzip tar01.zip

time bunzip2 tar01.tar.bz2



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | tar（gz） | 7za | zip | bz2 |
| 压缩时长 | 40.043s | 136.205s | 37.031s | 72.268s |
| 解压时长 | 6.259s | 43.179s | 7.075s | 49.024s |
| 压缩率 | 77.90% | 78.80% | 77.90% | 77.70% |

5）排序命令强化学习

自动生成多列文件1G，第一列是5位随机字符串，第二列是2位数值，第三列是5位数值

1）按照字符串顺序排序整个文件。

2）首先按第一列排序，第一列字符串相同的情况下，按照第二列数值排序，第二列数值相同的情况下按第三列排序。

3）考虑如何并行排序这个1G文件，加快速度（切分文件，分别压缩，然后再归并）

1).sort random.txt > t1.txt

2.)sort -k 1,1 -k 2,2n -k 3,3n random.txt >t2.txt

3.)

awk ‘{print $1} ‘ random.txt | sort >1.txt;

awk ‘{print $2} ‘ random.txt | sort -n > 2.txt;

awk ‘{print $NF} ‘ random.txt | sort -n 3.txt &&

sort 1.txt 2.txt 3.txt > randomnew.txt

1. 安装http服务

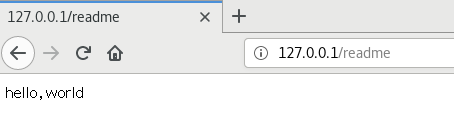
并在服务目录创建readme文件，里面写上hello world. 通过浏览器可以访问到自己开放的这个http服务，即打开 <http://127.0.0.1/readme> 看到这个服务，80是默认端口，可以不用写成 <http://127.0.0.1:80/readme>

http服务配置端口从80端口修改到8080端口，并且实现关机重启后服务自动启动

由于本机创建时自带http服务

创建readme

echo hello,world > /var/www/html/readme



修改端口

vim /etc/httpd/conf/httpd.conf

将listen 由80 改为8080即可重启http服务

systemctl restart httpd.service



设置开机启动服务

systemctl is-enabled httpd.service（查看是否开机启动）

systemctl enable httpd.service

重启即可自动开机启动

1. 安装mysql数据库服务，并进行常见的运维（导出，备份，还原）操作

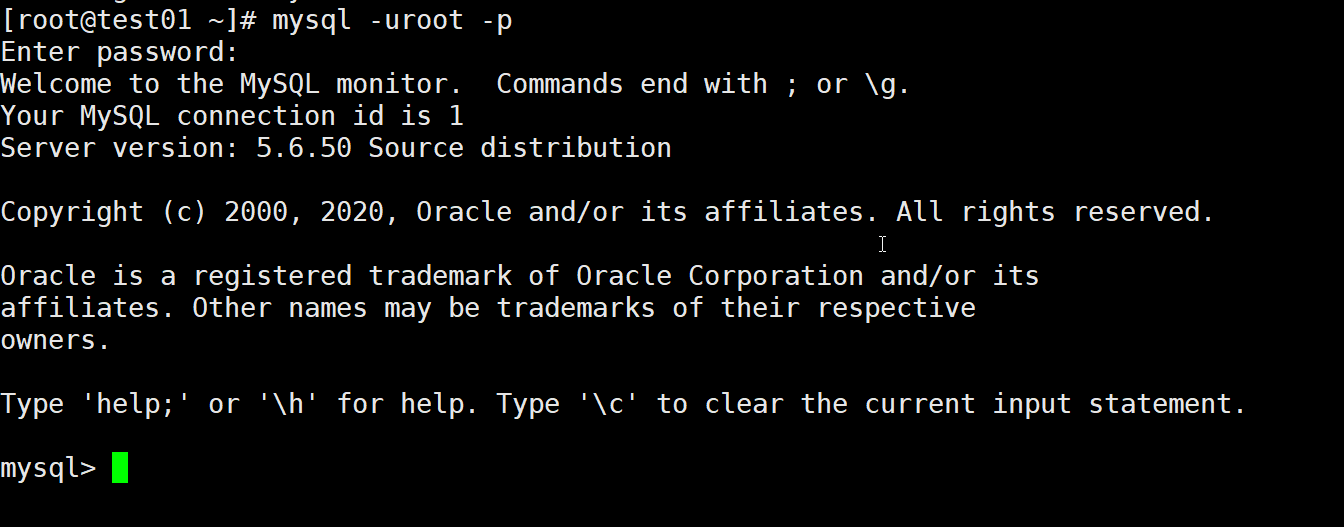
1）并实现创建一个数据库（例如叫testdb），并在数据库内创建一个表(score\_table)，表内包含id（整形），name（字符串），score（整形）这么几个字段，并且自己insert一些记录。

2）能够把创建的表dump出来，保存为score\_table.sql文件。

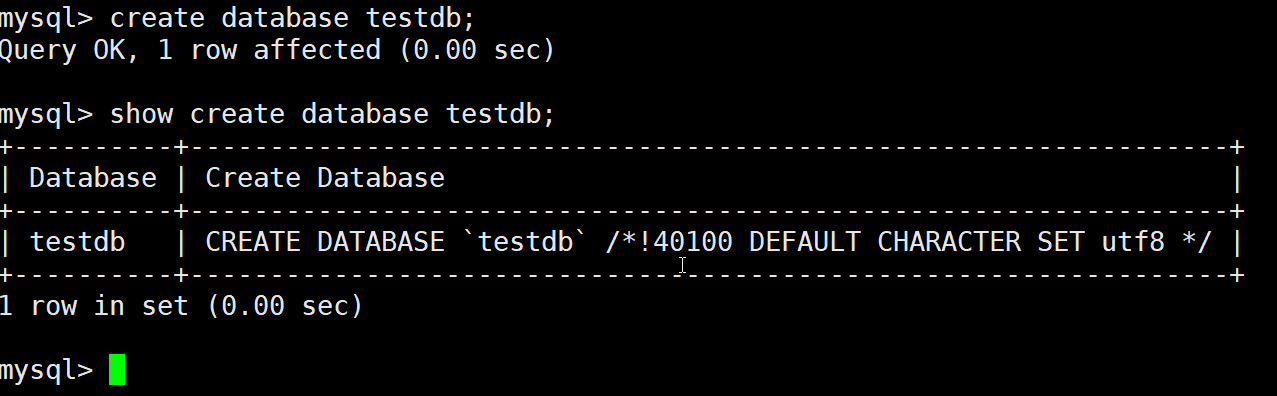
3）把数据库里的score\_table表删除了，用这个score\_table.sql能恢复出原表出来。

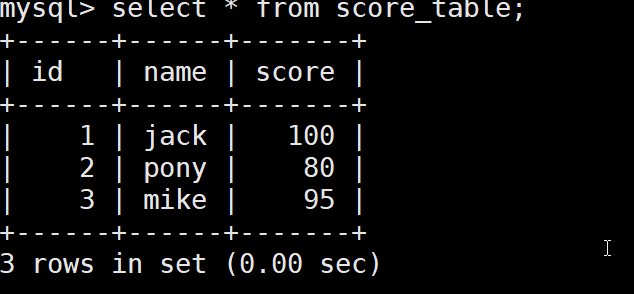
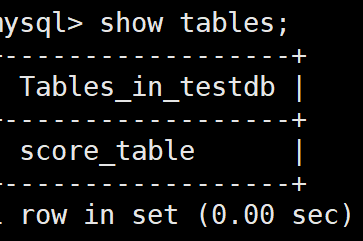
4）学习如果将score\_table表中，score（分数）在90分以上的同学记录导出来。

由于centos7以上版本通过yum安装只会装Mariadb，故本次通过阿里开源镜像下载mysql5.6.50安装包进行安装。



1. 采取系统默认参数进行创建





1. dump数据库表

mysqldump -uroot -p testdb score\_table > score\_table.sql

3.drop掉表后，恢复表

mysql> drop table score\_table;

Query OK, 0 rows affected (0.00 sec)

mysql> show tables;

Empty set (0.00 sec)

mysql> source /root/score\_table.sql

Query OK, 0 rows affected (0.00 sec)

恢复数据表

Query OK, 0 rows affected (0.00 sec)

mysql> show tables;

+------------------+

| Tables\_in\_testdb |

+------------------+

| score\_table |

+------------------+

1 row in set (0.00 sec)

4.导出查询结果

mysql> show variables like '%secure%';

+------------------+-------+

| Variable\_name | Value |

+------------------+-------+

| secure\_auth | ON |

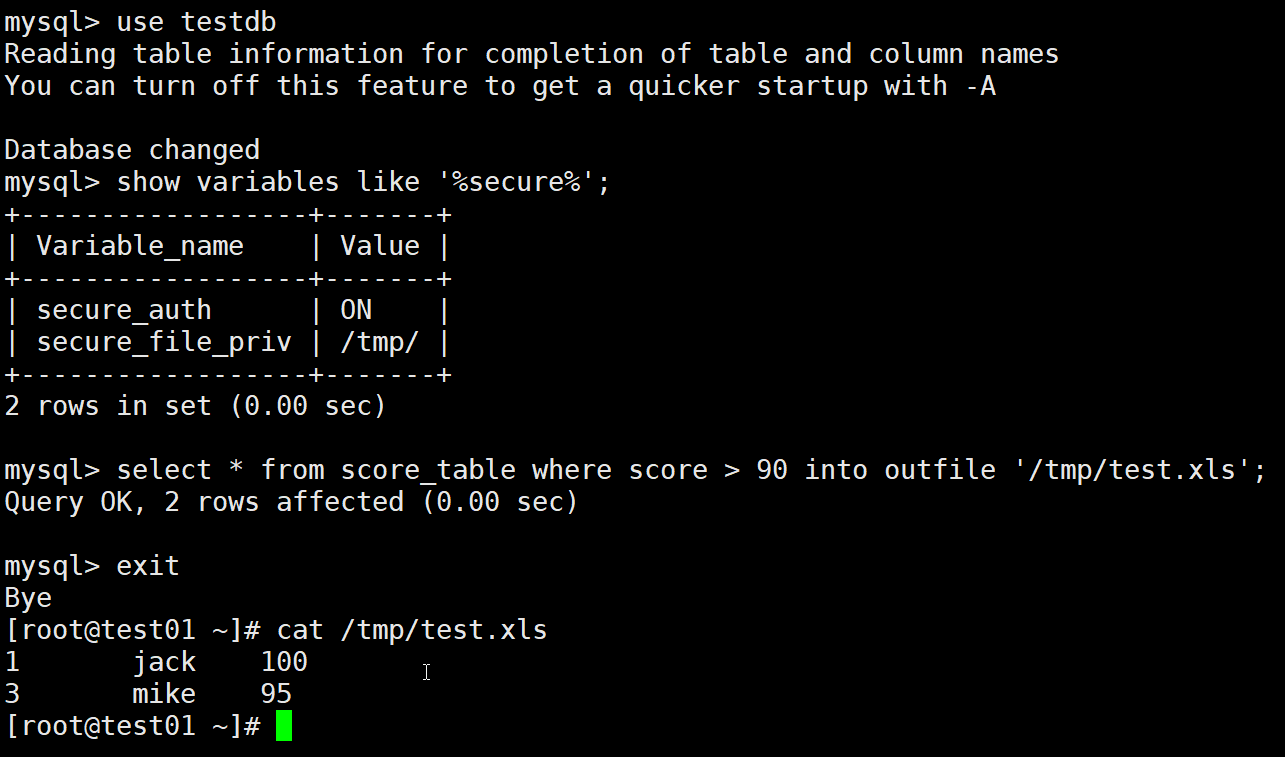
| secure\_file\_priv | NULL |

+------------------+-------+

2 rows in set (0.00 sec)

没有指定导出文件位置

修改my.cnf,添加secure\_file\_priv位置并重启mysql。导出查询结果。



下面的学习需要有网络环境，网络环境可以这样得到

1. 创建两个linux虚拟机，互相可以通信

2）没有装虚拟机的同学，我们可以给一个远程机器，这样和自己实验的机器形成一个网络环境

1. 学习iptables服务**（选学）**

安装iptables服务，并完成iptables几个简单功能

1. 假定之前安装的http服务开在8080端口，在iptables里面增加一个配置，使得原来访问readme能看到hello world，配置后不可打开（相当于封了8080端口）
2. 把httpd服务配置在80端口上启动服务。netstat -anp | grep httpd 可以看到这个服务在80端口上服务。 然后做一个nat转发，将8080端口的请求转发到80端口上，这样打开<http://127.0.0.1:80/readme> 和 <http://127.0.0.1:8080/readme> 看到的文件是一样的。

3）如果有两台linux（两个IP，可以是内网IP，只要能互通就行），可以在一台机器（甲）上启动httpd服务（80端口），然后把对另一台机器(乙)的iptables配置成80端口转发到甲的80端口上。这样是访问乙的80端口，和打开甲的80端口看到的内容一样的。

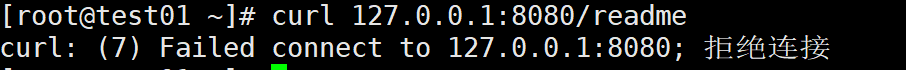
4）如果有三台linux，可以做另外一个实验，甲，乙，丙。甲乙上开httpd服务，都启动在80端口。丙上用iptables把 80端口的请求50% nat 到甲，50% nat 到 乙上，实现负载分流。当然也可以在2台linux上做实验，其中在甲机器上启动两个httpd实例，一个启动在80端口，一个启动在8080端口。然后乙机器50%的流量打给甲的80端口，50%的流量打给乙的7070端口。

1.添加规则拒绝8080端口访问

通过iptables -F 清空当前所有防火墙规则，以避免后添加的规则在执行链后得不到执行。添加规则 拒绝8080 端口访问

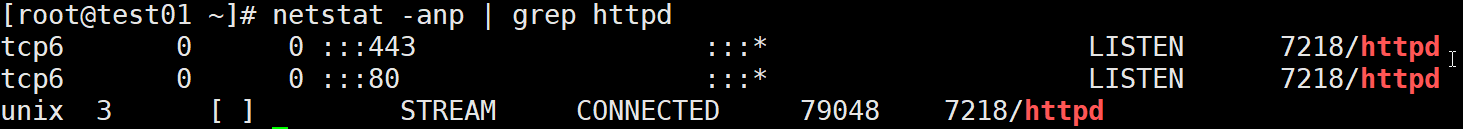


通过curl命令访问，结果如下



2.NAT转发

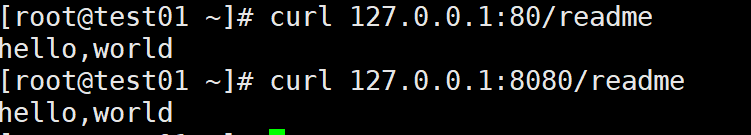
根据题目6中，可将httpd在80端口启动服务。通过netstat查看httpd服务运行端口80.



添加规则



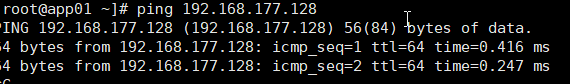
由于是在本地访问127，故数据包直接由内核发送到output链，不经过input



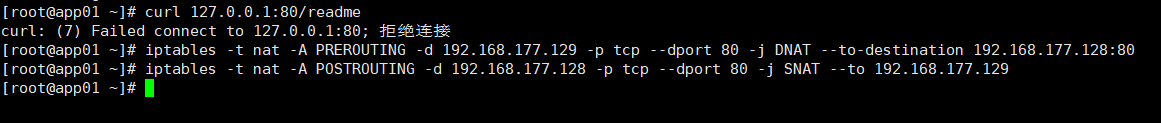
3.两台机器，将乙对自己80端口的访问转发到甲的80端口从而看到甲的内容。

机器乙，对甲进行ping命令，保证能够通信

echo 1 > /proc/sys/net/ipv4/ip\_forward 开启NAT功能



在机器乙中设置iptable规则



4.三台机器实现负载均衡

在128服务器上添加规则

iptables -A PREROUTING -t nat -p tcp -d 192.168.177.128 --dport 80 -m statistic --mode random --probability 0.5 -j DNAT --to-destination 192.168.177.130:80

iptables -A PREROUTING -t nat -p tcp -d 192.168.177.128 --dport 80 -m statistic --mode random --probability 1 -j DNAT --to-destination 192.168.177.129:80

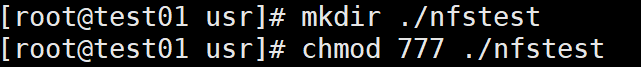
1. 学习NFS服务**（选学）**

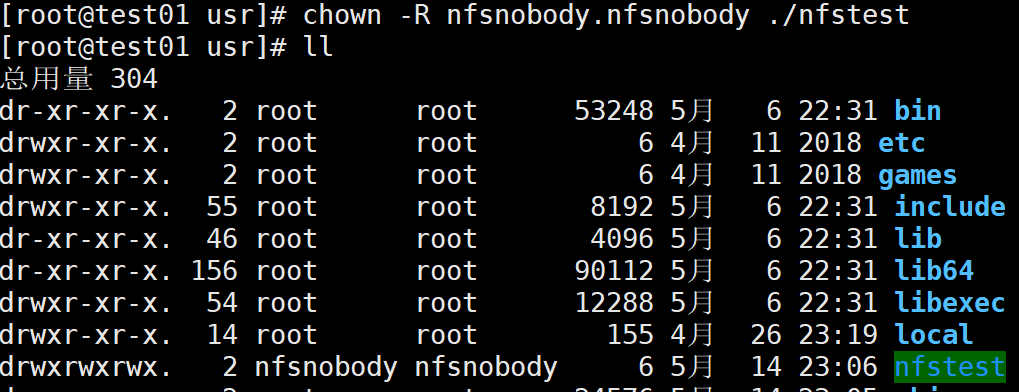
两个Linux机器（甲和乙），网络可以互通，将甲的硬盘远程mount到乙的机器上。并测试下这种远程硬盘的读写速度和本地硬盘读写速度的差异，找一个度量硬盘读写速度的工具，并对比这种差异，比如dd命令创建一个文件，在本地硬盘创建一个1G文件的时间和在远程硬盘（但mount在本地）上创建的时间的差距。

1.安装NFS工具包、rpcbind包，将test01主机设为服务器端



2.创建共享目录并赋权，更改属主和属组以便安全





3.写入配置文件

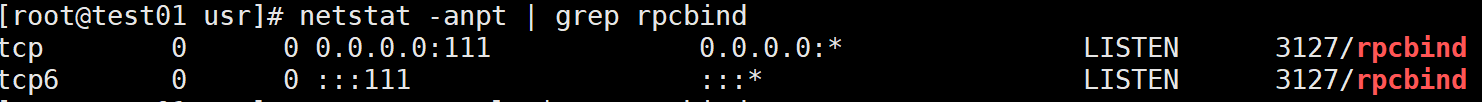


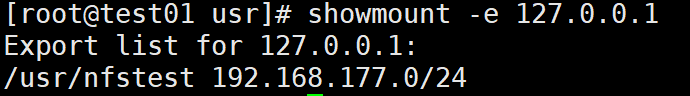
对177网段下子网掩码为255.255.255.0的ip 授权接入，赋予rw读写权利，sync同步写到内存和硬盘中，all\_squash 客户机访问本地机赋予时映射成匿名用户nfsnobody,insecure在CentOs7中允许客户端从大于1024的tcp/ip端口连接服务器。

4.开启NFS\RPCBIND服务并查看是否启动成功，检查nfs挂载情况

systemctl start rpcbind.service 开启rpcbind

systemctl start nfs 开启nfs

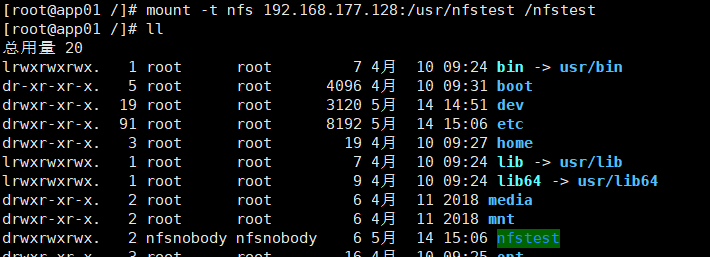




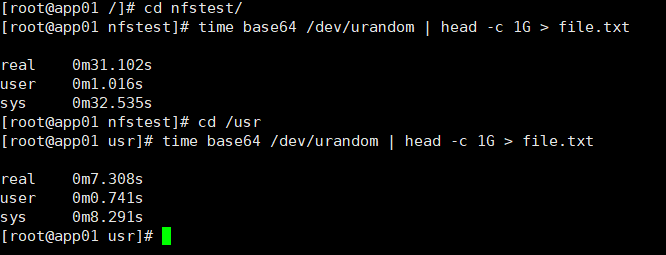
5.部署客户端

客户端安装服务端同样的包，并启动服务，将服务端的目录挂载到本地/nfstest目录下





6.测试时间差距



在远程硬盘上创建文件时间31.102s 在本地创建文件7.308s，时间差距较大。服务端配置文件改为async可改善情况，但数据安全无法得到有效保障。

1. 学习awk命令，简单的字符串处理。awk中需要了解的内容包括。

* BEGIN、END的含义
* NR和FNR 处理两个文件
* 数组的理解（常用于计数器）

学习并理解这个命令：

假定有两个文件，分别是account和cdr。例如张三的卡号是000001，消费了2次，一次10元，一次20元。现在需要用awk命令得到张三和李四消费了多少元。

#cat account

张三|000001

李四|000002

#cat cdr

000001|10

000001|20

000002|30

000002|15

理解下面命令：

awk -F \| 'NR==FNR{a[$2]=$1;next}{sum[a[$1]]+=$2;}END{ for (var in sum) { print var"\t"sum[var];} }' account cdr

* awk的内置函数rand，gsub，substr，index，length，substr
* awk 里面用system函数执行一个脚本，system也是awk里面一个内置函数，但是比较重要，这使得awk可以启动其他任何脚本命令的能力

awk命令非常强大，可以很方便的做很多统计性工作，开发和运维都需要大量使用。

基础语法

awk [-F ERE] [-v assignment] ... program（pattern{action}）[argument ...]

BEGIN动作指令仅在读取任何数据记录之前执行一次

END动作指令仅在读取完所有数据记录后执行一次

FNR 当前输入文档的当前行号 （两个文档各自有不同的行号）

NR 输入流的当前行号 （即将两个文档合并作为一个文档看待）

数组

array\_name[index]=value

awk中的index是以字符串作为数组索引的

用于计数器时，字符串和空字符串在参与运算时会被当做数字0

理解命令

最终结果是输出张三李四各自消费的金额总数

首先 -F 指定分割符为| 通过 \进行转义避免识别为管道符

执行的文本为account 、cdr

在内部指令中，首先进行pattern部分的条件判断即NR==FNR，

当NR==FNR时，执行{a[$2]=$1;next}，即 a[000001]=张三，next，a[000002]=李四

当读取到第3行时，NR！==FNR，执行{sum[a[$1]]+=$2;}，

即 sum[a[$1]] =sum[a[$1]]+$2,即sum[a[000001]]=sum[a[000001]]+10，

sum[a[000001]]=sum[a[000001]]+20，最终sum[a[000001]]值为30，

同理sum[a[000002]]值为45，行读取完毕。

执行END{ for (var in sum) { print var"\t"sum[var];} }

循环读取sum数组，按索引读取顺序，到var变量中

打印var变量，间隔1个制表符，sum[var]的值

即 李四 45

张三 30

内置函数

rand（） 返回0到1之间的随机数

gsub（r,s,[,t]）将字符串t中所有与正则表达式r中匹配字符串全部替换成s，若未指定t则以$0作为t

sub（r，s，[，t]）与gsub类似，不同在于仅替换第一个与r中匹配的字符串。

index（字符串1，字符串2）返回字符串2在字符串1中的位置

length（[s]）统计字符串s的长度，若未指定则统计$0的长度

substr（s，i，[，n]）对字符串s进行截取，从第i位开始截取n个。若未指定n，则一直到字符串s的末尾。

执行脚本

system（“linux命令”）

1. Linux系统配置的学习

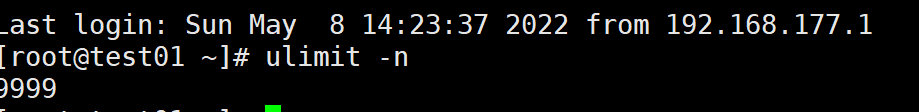
之前做iptables的nat，大家应该接触过需要在/etc/sysctl.conf 下配置：net.ipv4.ip\_forward=1 才可能激活nat的功能。我们需要熟悉和了解一系列linux下的配置。以下是最常见的一些配置，大家上网找资料学习下，并掌握。

* 配置linux文件最大打开数（ulimit -n 可以看到当前系统默认值）

系统默认1024个，通过在/etc/security/limits.conf增加

\* soft nofile 9999 、\* hard nofile 9999

后重启即可修改成功



* 学习交换区的概念

1. cat /proc/sys/vm/swappiness 了解下这里配置的含义
2. 内存不够的时候，不能增加内存需要增加自定义交换区，如何配置？

用swapon，swapoff增加和取消自定义交换区

1. 用free命令考察交换区使用情况

1）swappiness，取值范围是0-100，值越高内核会越积极的使用swap。值=0时，通过清理缓存来回收内存。值为100时，从缓存回收内存与swap交换的优先级一样。

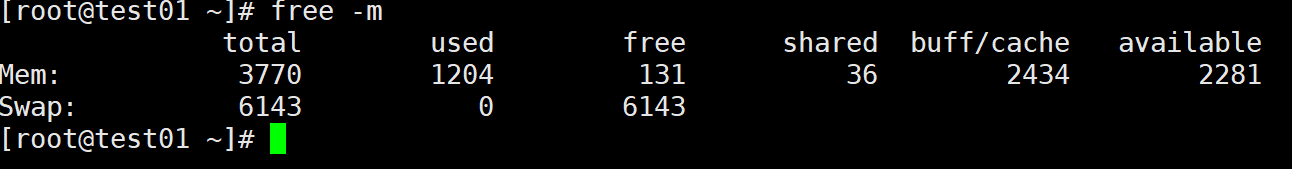
2）3）因为我的虚拟机按4G内存设立，故也建立4Gswap

dd if=/dev/zero of=/var/swapfile bs=1M count=4096

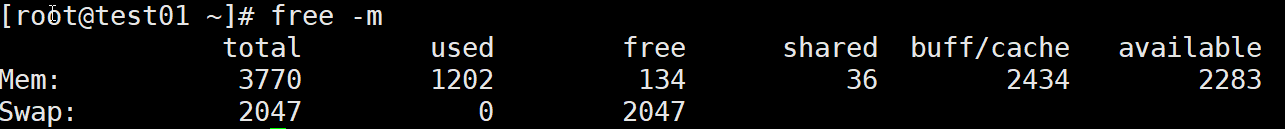
mkswap /var/swapfile

swapon /var/swapfile

free -m 发现建系统时已经分配了2Gswap



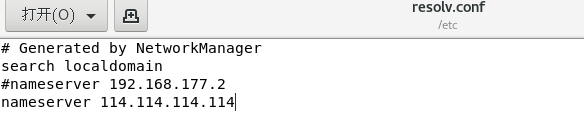
swapoff /var/swapfile



* DNS的配置

在/etc/resolv.conf中配置dns。将本机的DNS解析使用114.114.114.114，如何配置？配置好以后，怎么整明自己解析的baidu.com 是通过114.114.114.114解析出来的，了解下dig命令。看看dig baidu.com返回什么？

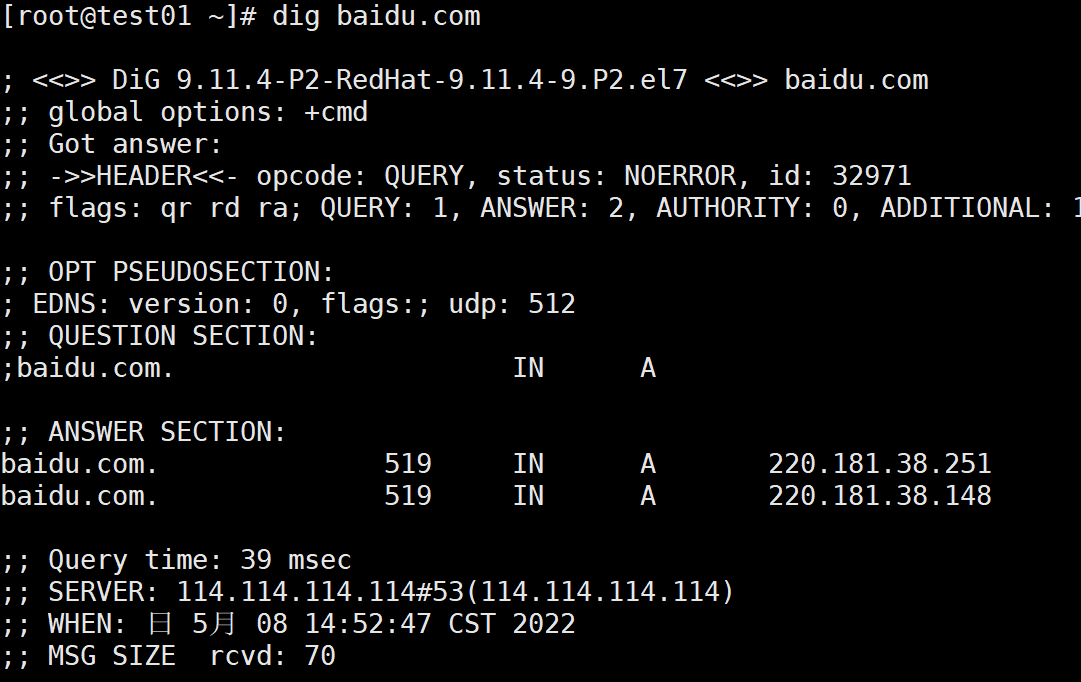
/etc/resolv.conf 中配置dns



解析顺序

先会找/etc/hosts文件，如果/etc/hosts文件没有对应，会找DNS缓存中是否有对应关系，最后才会去找DNS(/etc/resolv.conf)服务器，发起请求获得返回IP地址。

dig命令



第一部分显示 dig 命令的版本和输入的参数。

第二部分显示服务返回的一些技术详情，比较重要的是 status。如果 status 的值为 NOERROR 则说明本次查询成功结束。

第三部分中的 "QUESTION SECTION" 显示我们要查询的域名。

第四部分的 "ANSWER SECTION" 是查询到的结果。

第五部分则是本次查询的一些统计信息，比如用了多长时间，查询了哪个 DNS 服务器，在什么时间进行的查询

* Yum源的配置

通常机器默认yum源都很慢，往往在国外，尝试配置网易yum源，或者阿里云yum源。对比下载安装的速度差异。

1.备份原yum源

mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup

2.通过wget 下载阿里镜像源

wget -O /etc/yum.repos.d/CentOS-Base.repo https://mirrors.aliyun.com/repo/Centos-7.repo

3.运行 yum makecache 生成缓存

4.非阿里云ECS用户修改配置

sed -i -e '/mirrors.cloud.aliyuncs.com/d' -e '/mirrors.aliyuncs.com/d' /etc/yum.repos.d/CentOS-Base.repo

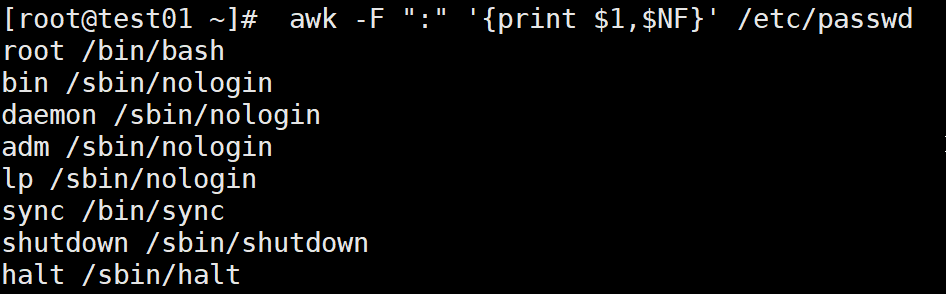
* cat /etc/passwd

查看当前机器的所有用户，有时候判断机器是否被黑了，或者出现了不明登录用户就要用这个命令。了解nologin的含义。几个系统默认账号的含义。

nologin是禁止用户登录，一般用于给mysql用户等降权，保证安全性。

通过awk命令 对用户进行查看

超级用户root，程序用户：bin daemon等在/bin /sbin 目录下的账户



* cat ~/.ssh/known\_hosts

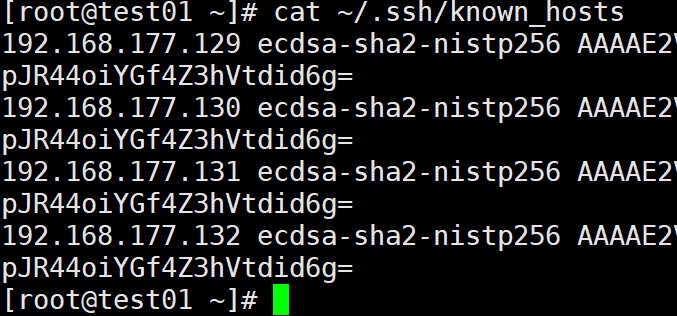
这个里面存放了什么？ /root/.ssh/id\_rsa.pub 这里面存放的是什么？ 如果本机（比如叫机器A）上没有，可以通过什么方式创建。有了id\_rsa.pub后，怎么把这个东西存放到其他机器上（比如叫机器B），使得本机（机器A）可以ssh 到那台机器B而不需要密码。

~/.ssh/known\_hosts存放了访问过的计算机的公钥

/root/.ssh/id\_rsa.pub存放本地机器的公钥。

ssh-keygen 生成公钥，ssh-copy-id -i ~/.ssh/id\_rsa.pub root@远程服务端地址

输入服务端密码即可



1. 学会使用github

* 每个人在<https://github.com/>上注册账号，上传一些自己的代码或者文档。学会checkin checkout,clone,fetch,pull 等命令。
* 理解分支

什么是主分支，一个人的项目和多个人合作的项目会有很大区别，一个人的项目搞搞主分支就行了，但是多人合作的项目会有很大区别，特别是同一个部分可能有两个人一起开发。

* 理解Tag

一个大的项目可能有多人合作开发，肯定会在一个合适的时候，大家都开发结束了，这个时候需要有项目管理人员做一个Tag（往往一个Tag 对应一个版本号），然后测试人员就取这个tag下的代码，其他人可以继续在主分支提交代码。测试人员测试OK后，运维人员就需要从这个tag上取代码下来，到线上环境去跑。

因此如果不做开发，分支用到的机会很少，但是Tag用到的机会几乎是百分之百的。

下载gitbash，通过绑定github的ssh进行链接。

建立一个新仓库，对仓库进行连接。

<https://github.com/HanchuQin/pennystudy>

在本地仓库，执行git init ; git add . git commit -m "20220508第一次笔记作业"

已准备好代码数据，执行

git remote add origin <https://github.com/HanchuQin/pennystudy.git> 添加远程仓库

提交代码到远程仓库的master 分支 git push -u origin master

命令学习

git clone + 远端仓库地址 下载他人仓库

git checkout 分支名 切换到分支

git checkout 文件名 撤销修改

git fetch 相当于是从远程获取最新版本到本地，不会自动merge

git pull =git fetch + merge合并

git tag 标签名 git push 分支名 标签名

