

Day06课堂笔记

01-存储文件系统

1.1 什么是文件系统?

1.2 什么是格式化?

1.3 什么是分区?

1.4 挂载

1.5 总结

02-三剑客grep-sed-awk

2.1 基本概念

2.2 grep的用法

2.3 sed使用

2.4 awk使用

2.5 总结

03-补充本地YUM仓库搭建

3.1 ISO 镜像搭建本地的仓库Local

3.2 本地局域网搭建基于HTTP的仓库

2-1 Shell脚本（能看懂，能写简单的）

2-2 定时计划任务，周期任务

01-存储文件系统

1.1 什么是文件系统?

操作系统能够读取数据文件，必须有能够支持这个系统的文件格式，文件系统

-Windows的文件系统FAT，NTFS

-Linux的文件系统：ext2 ext3 ext4 xfs

-vmware文件系统：vmfs6

```
1 [kiosk@foundation0 ~]$ ls /lib/modules/`uname -r`/kernel/fs/ -l
2 total 20
3 -rw-r--r--. 1 root root 7580 Apr 15 2022 binfmt_misc.ko.xz
4 drwxr-xr-x. 2 root root 30 Jan 31 2023 cachefiles
5 drwxr-xr-x. 2 root root 24 Jan 31 2023 ceph
6 drwxr-xr-x. 2 root root 24 Jan 31 2023 cifs
7 drwxr-xr-x. 2 root root 23 Jan 31 2023 dlm
8 drwxr-xr-x. 2 root root 25 Jan 31 2023 exfat
9 drwxr-xr-x. 2 root root 24 Jan 31 2023 ext4
10 drwxr-xr-x. 2 root root 60 Jan 31 2023 fat
11 drwxr-xr-x. 2 root root 27 Jan 31 2023 fscache
12 drwxr-xr-x. 2 root root 46 Jan 31 2023 fuse
13 drwxr-xr-x. 2 root root 24 Jan 31 2023 gfs2
14 drwxr-xr-x. 2 root root 25 Jan 31 2023 isofs
15 drwxr-xr-x. 2 root root 24 Jan 31 2023 jbd2
16 drwxr-xr-x. 2 root root 25 Jan 31 2023 lockd
17 -rw-r--r--. 1 root root 5336 Apr 15 2022 mbcache.ko.xz
18 drwxr-xr-x. 2 root root 25 Jan 31 2023 netfs
19 drwxr-xr-x. 5 root root 120 Jan 31 2023 nfs
20 drwxr-xr-x. 2 root root 46 Jan 31 2023 nfs_common
21 drwxr-xr-x. 2 root root 24 Jan 31 2023 nfsd
22 drwxr-xr-x. 2 root root 4096 Jan 31 2023 nls
23 drwxr-xr-x. 2 root root 27 Jan 31 2023 overlayfs
24
```

1.2 什么是格式化?

- 格式化才能产生文件系统
- 格式化的结果是：产生文件系统、inode、block
- 格式化是必须要做的

1.3 什么是分区?

-分区的目的是什么：逻辑上安全隔离、基于应用来存放在不同的存储空间里面

-分区不是一定要做的

-主分区和扩展分区的区别？

MBR（4个主分区）

-主分区最多是4个，扩展分区最多几百个

GPT（没有限制）

-推荐一本书（大话存储）

1.4 挂载

-windows和linux文件存储的区别？

1. Windows：先分区-装系统-CDEF盘-放进去数据。C目录对应一个存储空间（目录和存储是一起的）
2. Linux：目录和存储空间是分开的

-挂载的本质：指定一个目录存放的位置，不挂载，数据会被默认放在/所挂载的存储位置

-挂载的方法（必考）：

1. 临时挂载：`mount /dve/vdb1（存储） /mnt/glab（目录）`
 - a. 重启以后挂载就消失
 - b. 问题：原来创建的文件会不会消失？不会
 - c. 应用：测试
2. 永久挂载：`/etc/fstab`，开机启动的时候就会加载，一旦有问题系统起不来（紧急模式）
 - a. `UUID=5e75a /boot xfs defaults,nfs,rw, 0 0`
 - b. 挂载点存储 - 挂载目录 - 存储空间被格式化后的文件系统格式 - options - 压缩 - 校验
 - c. 挂载存储：可以是唯一UUID、还可以是/dev/vdb1分区设备文件
3. 按需挂载（autofs）

```
1 #查看当前系统的挂载目录
2 [root@servera ~]# df -Th
3 Filesystem      Type      Size  Used Avail Use% Mounted on
4 devtmpfs        devtmpfs  844M   0  844M   0% /dev
5 tmpfs           tmpfs     888M   0  888M   0% /dev/shm
6 tmpfs           tmpfs     355M  9.6M  346M   3% /run
7 /dev/vda4       xfs       9.4G  3.1G  6.3G  33% /
8 /dev/vda3       xfs       495M  160M  335M  33% /boot
9 /dev/vdb1       ext4      974M   24K  907M   1% /mnt/qlab
10 /dev/vda2       vfat      200M   7.0M  193M   4% /boot/efi
11 tmpfs           tmpfs     178M   0  178M   0% /run/user/0
12
13
14
15 #查看当前系统中的磁盘以及分区UUID
16 [root@servera ~]# lsblk -f
17 NAME      FSTYPE FSVER LABEL UUID                               FSAVAIL FS
18 USE% MOUNTPOINTS
19 vda
20 └─vda1
21 └─vda2 vfat   FAT16      7B77-95E7                               192.8M
22    3% /boot/efi
23 └─vda3 xfs     boot     5e75a2b9-1367-4cc8-bb38-4d6abc3964b8  334.7M
24    32% /boot
25 └─vda4 xfs     root     fb535add-9799-4a27-b8bc-e8259f39a767   6.3G
26    33% /
27 vdb
28 └─vdb1 ext4    1.0      e0424b27-db45-4f13-b23b-e694b2edad0e  906.2M
29    0% /mnt/qlab
30 vdc
31 vdd
32
33 #fdisk分区的过程
34 [root@servera ~]# fdisk /dev/vdb
35
36 Welcome to fdisk (util-linux 2.37.4).
37 Changes will remain in memory only, until you decide to write them.
38 Be careful before using the write command.
```

```

36 This disk is currently in use – repartitioning is probably a bad idea.
37 It's recommended to umount all file systems, and swapoff all swap
38 partitions on this disk.
39
40
41 Command (m for help): p
42
43 Disk /dev/vdb: 5 GiB, 5368709120 bytes, 10485760 sectors
44 Units: sectors of 1 * 512 = 512 bytes
45 Sector size (logical/physical): 512 bytes / 512 bytes
46 I/O size (minimum/optimal): 512 bytes / 512 bytes
47 Disklabel type: dos
48 Disk identifier: 0x773a61d8
49
50 Device      Boot Start      End Sectors  Size Id Type
51 /dev/vdb1           2048 2099199 2097152    1G 83 Linux
52
53 Command (m for help): n
54 Partition type
55   p   primary (1 primary, 0 extended, 3 free)
56   e   extended (container for logical partitions)
57 Select (default p): p
58 Partition number (2-4, default 2):
59 First sector (2099200-10485759, default 2099200):
60 Last sector, +/-sectors or +/-size{K,M,G,T,P} (2099200-10485759, default
61 10485759): +1G
62
63 Created a new partition 2 of type 'Linux' and of size 1 GiB.
64
65 Command (m for help): p
66 Disk /dev/vdb: 5 GiB, 5368709120 bytes, 10485760 sectors
67 Units: sectors of 1 * 512 = 512 bytes
68 Sector size (logical/physical): 512 bytes / 512 bytes
69 I/O size (minimum/optimal): 512 bytes / 512 bytes
70 Disklabel type: dos
71 Disk identifier: 0x773a61d8
72
73 Device      Boot  Start      End Sectors  Size Id Type
74 /dev/vdb1           2048 2099199 2097152    1G 83 Linux
75 /dev/vdb2           2099200 4196351 2097152    1G 83 Linux
76
77 Command (m for help): w
78 The partition table has been altered.
79 Syncing disks.
80 [root@servera ~]#
81 [root@servera ~]# lsblk

```

```

82 NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
83 vda 252:0 0 10G 0 disk
84 └─vda1 252:1 0 1M 0 part
85 └─vda2 252:2 0 200M 0 part /boot/efi
86 └─vda3 252:3 0 500M 0 part /boot
87 └─vda4 252:4 0 9.3G 0 part /
88 vdb 252:16 0 5G 0 disk
89 └─vdb1 252:17 0 1G 0 part /mnt/qlab
90 └─vdb2 252:18 0 1G 0 part
91 vdc 252:32 0 5G 0 disk
92 vdd 252:48 0 5G 0 disk
93 [root@servera ~]#
94
95
96 #格式化的过程
97 [root@servera ~]# mkfs.
98 mkfs.cramfs mkfs.ext3 mkfs.fat mkfs.msdos mkfs.xfs
99 mkfs.ext2 mkfs.ext4 mkfs.minix mkfs.vfat
100 [root@servera ~]# mkfs.xfs /dev/vdb2
101 meta-data=/dev/vdb2 isize=512 agcount=4, agsize=65536 blk
102 s
103 = sectsz=512 attr=2, projid32bit=1
104 = crc=1 finobt=1, sparse=1, rmapbt=
105 0
106 = reflink=1 bigtime=1 inobtcount=1
107 data = bsize=4096 blocks=262144, imaxpct=25
108 = sunit=0 swidth=0 blks
109 naming =version 2 bsize=4096 ascii-ci=0, ftype=1
110 log =internal log bsize=4096 blocks=2560, version=2
111 = sectsz=512 sunit=0 blks, lazy-count=1
112 realtime =none extsz=4096 blocks=0, rtextents=0
113 Discarding blocks...Done.
114
115 #挂载的过程
116 mount /dev/vdb2 /mnt/qlab #临时挂载
117
118 vim /etc/fstab
119 [root@servera ~]# cat /etc/fstab
120 /dev/vdb1 /mnt/qlab ext4 defaults 0 0
121 [root@servera ~]# mount -a #不报错事最好的结果
122

```

1.5 总结

分区

格式化

挂载 (*****)

02-三剑客grep-sed-awk

2.1 基本概念

-应用场景 (grep-sed-awk) : 对文件、输出文本、一切字段进行信息的提取

grep option 正则表达式 目标文本

grep: option

-i: 忽略大小写

-v: 反向查找

-o: 只显示匹配的字符

-E: 使用扩展正则表达式

-A num: 显示匹配的和后num行

-B num: 显示匹配的和前num行

-C num: 显示匹配的前后num行

正则表达式:

- 标准正则表达式

^过滤以y开头的输出

\$过滤以w结尾的输出

^\$过滤除了空行以外的其他输出

.代表任意单个字符, 包含空格

\转义字符

*重复0次或者多次

.*匹配所有字符

[abc]指定范围内的任意单个字符

[^abc]除了abc意外的任意一个, 就是!, 非

a\{n,m\}把a字符重复n到m次

\(ab\).*\1分组, 以ab为一个组, 后向引用重复1次的行

- 扩展正则表达式egrp

+重复一个或一个以上前面的字符

? 重复0个或1个前面的字符

| 或

(ab) 用户分组字符串,\1, \2, \3.....

{m,n} 次数匹配

#需求: 取出系统的IP地址

```
[root@servera ~]# ip add |grep eth0$ | egrep -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}/[0-9]{1,2}'  
172.25.250.10/24
```

grep的难点在正则表达式

2.2 grep的用法

grep的命令

Plain Text

```
1 #显示/proc/meminfo文件中以不区分大小的s开头的行  
2 [root@servera ~]# grep -i ^s /proc/meminfo  
3  
4 #显示/etc/passwd中以nologin结尾的行  
5 [root@servera ~]# grep nologin$ /etc/passwd  
6  
7 #取出默认shell为/sbin/nologin的用户列表  
8 [root@servera ~]# grep nologin$ /etc/passwd | cut -d: -f1  
9  
10 #显示/etc/inittab文件中以一个数字开头并以一个与开头数字相同的数字结尾的行  
11 [root@servera ~]# grep '^([0-9]\).*\1$' /etc/inittab  
12  
13  
14 #取出本机MAC地址  
15 [root@servera ~]# ip link | grep 'ff$' | grep -E '[0-9|a-f]{1,2}:[0-9|a-f]{1,2}:[0-9|a-f]{1,2}:[0-9|a-f]{1,2}:[0-9|a-f]{1,2}:[0-9|a-f]{1,2}' -o  
16  
17
```

2.3 sed使用

```
sed [option] 'AddressCommand' file....
```

sed:流编辑器，跟word不一样，word是可渲染的软件

流编辑器是纯文本的编辑方式

sed先把IPADD文件读入内存，操作仅仅影响在内存中的数据，默认情况是不会改变原始文件的，要改变原始文件就需要加-i，就会把内存中的文件回写到原始文件。

[option]

-n:只显示符合条件的行

-i: 修改源文件

-r: 表示使用扩展的正则表达式

Address

1、 StartLine,EndLine

比如1,100

\$: 最后一行

2、 /RegExp/

/^root/

3、 /pattern1/,/pattern2/

第一次被pattern1匹配到的行开始，至第一次被pattern2匹配到的行结束，这中间的所有行

4、 LineNumber

指定的行

5、 StartLine, +N

从startLine开始，向后的N行

Command:

d: 删除符合条件的行;

p: 显示符合条件的行;

a \string: 在指定的行后面追加新行，内容为string

\n: 可以用于换行

i \string: 在指定的行前面添加新行, 内容为string

r FILE: 将指定的文件的内容添加至符合条件的行处

w FILE: 将地址指定的范围内的行另存至指定的文件中;

s/pattern/string/修饰符: 查找并替换, 默认只替换每行中第一次被模式匹配到的字符串
加修饰符

g: 全局替换

i: 忽略字符大小写

s///: s###, s@@@

\(\), \1, \2

l..e: like-->liker

love-->lover

like-->Like

love-->Love

&: 引用模式匹配整个串

```
1 sed练习题:
2 #删除/etc/rsyslog.conf文件中行首空白的行
3 [root@servera ~]# sed '/^\s/d' /etc/rsyslog.conf
4
5 #替换/etc/rsyslog.conf文件中" fwdRule1"一行中的数字为5
6 [root@servera ~]# grep Rule1 /etc/rsyslog.conf
7 #queue.filename="fwdRule1"          # unique name prefix for spool files
8
9 [root@servera ~]# sed '/Rule1/p' /etc/rsyslog.conf -n
10 #queue.filename="fwdRule1"          # unique name prefix for spool files
11
12 [root@servera ~]# sed -n 's/fwdRule1/fwdRule5/p' /etc/rsyslog.conf
13
14 #删除/etc/rsyslog.conf文件中的空白行
15 [root@servera ~]# sed '/^\s*$/d' /etc/rsyslog.conf
16
17 #删除/etc/rsyslog.conf文件中开头的#号
18 [root@servera ~]# sed 's/^#//' /etc/rsyslog.conf
19
20 #删除一个#开头后面有一个空格得行
21 删除#空格
22 [root@servera ~]# sed 's/^#\s//' /etc/rsyslog.conf
23 删除#空格的这一行
24 [root@servera ~]# sed '/^#\s/d' /etc/rsyslog.conf
25 #删除一个或者多个空白开头后见加一个#得所有行
26 [root@servera ~]# sed -r 's/\s+##/' /etc/rsyslog.conf
27
28 #取出一个文件路劲的目录名
29 [root@servera ~]# egrep -o "[a-z|A-Z|0-9|/]+" /etc/rsyslog.conf
30
31 #通过sed来取出ip地址
32
33 [root@servera ~]# sed -n '10p' IPADD | sed 's/inet//' | sed 's/brd.*//' |
34 sed -r 's/\s*//'
35 172.25.250.10/24
36
37 [root@servera ~]# sed -n '10p' IPADD | sed 's/inet//' | sed 's/brd.*//' |
38 sed 's/\s*//'
```

2.4 awk使用

awk是一种轻量级的编程语言，非长强大

awk 轻量级编程语言

FILENAME	awk浏览的文件名
FNR	浏览文件的记录数
FS	设置输入域分隔符，等价于命令行 -F选项
NF	浏览记录的域的个数
NR	已读的记录数
OFS	输出域分隔符
ORS	输出记录分隔符
RS	控制记录分隔符

\$0 变量是指整条记录。\$1表示当前行的第一个域,\$2表示当前行的第二个域,.....以此类推。

\$NF是number finally,表示最后一列的信息，跟变量NF是有区别的，变量NF统计的是每行列的总数



Plain Text |

```
1 搜索/etc/passwd有root关键字的所有行
2  awk '/root/' /etc/passwd
3  指定分隔符，并且打印最后一个字段
4  awk -F ":" '{print $NF}' /etc/passwd
5
6  获取第12到31行的第一列的信息
7  awk -F ":" '{if(NR<31 && NR>12) print $1}' /etc/passwd
```

2.5 总结

-相同需求可以综合使用三剑客的命令

把stat /etc/hosts 输出的0644取出来

```
stat -c %a /etc/hosts
```

```
stat /etc/hosts | awk -F "[(/]" 'NR==4 {print $2}'
```

```
stat /etc/hosts | sed -rn '4s#^.*\((.*)/-.*$#\1#p'
```

如何过滤出已知当前目录下glab中的所有一级目录(提示:不包含glab目录下面目录的子目录及隐藏目录, 即只能是一级目录)?

```
tree -L 1 -d
```

```
ls -lF glab/ | sed -n '/^d/p' | awk -F " " '{print $NF}'
```

03-补充本地YUM仓库搭建

3.1 ISO 镜像搭建本地的仓库Local

```
1 上传ISO镜像到ntp服务器 (winscp)
2 [root@ntp ~]# ls /iso/
3 RHEL7OSP-6.0-2015-02-23.2-x86_64.iso
4 rhel-server-7.1-x86_64-dvd.iso
5
6 挂载系统安装光盘, 建立临时本地 YUM 仓库, 安装 Apache
7 [root@ntp ~]# mkdir /dvd
8 [root@ntp ~]# mount /iso/rhel-server-7.1-x86_64-dvd.iso /dvd/
9 建立本地 YUM 仓库
10 [root@ntp ~]# vim/etc/yum.repos.d/dvd.repo
11 [DVD]
12 name=dvd
13 baseurl=file:///dvd
14 enabled=1
15 gpgcheck=0
16 测试本地仓库可用性
17 [root@ntp ~]# yum makecache ; yum repolist all
18
```

3.2 本地局域网搭建基于HTTP的仓库

```
1 安装 Apache
2 [root@ntp ~]# yum -y install httpd
3 启动 httpd 服务
4 [root@ntp ~]# systemctl start httpd.service
5 查看 httpd 服务状态
6 [root@ntp ~]# systemctl status httpd.service
7 设置 httpd 服务开机自启动
8 [root@ntp ~]# systemctl enable httpd.service
9
10
11 [root@ntp ~]# mkdir /var/www/html/{dvd,openstack}
12
13 设置开机自动挂载
14 [root@ntp ~]# cat /etc/fstab
15 /iso/rhel-server-7.1-x86_64-dvd.iso /var/www/html/dvd iso9660 defaults,loop,ro 0 0
16 /iso/RHEL7OSP-6.0-2015-02-23.2-x86_64.iso /var/www/html/openstack iso9660
    defaults,loop,ro 0 0
17
18 挂载 ISO
19 [root@ntp ~]# mount -a
20
21 测试挂载结果
22 [root@ntp ~]# df
23 Filesystem                1K-blocks    Used Available Use% Mounted on
24 /dev/mapper/rhel-root      18307072  5259852   13047220  29% /
25 devtmpfs                   932780         0     932780    0% /dev
26 tmpfs                      942368         0     942368    0% /dev/shm
27 tmpfs                      942368    16928     925440    2% /run
28 tmpfs                      942368         0     942368    0% /sys/fs/cgroup
29 /dev/sda1                  508588    121808     386780   24% /boot
30 /dev/loop1                 526940    526940         0 100% /var/www/html/opens
    tack
31 /dev/loop0                 3798292  3798292         0 100% /var/www/html/dvd
32
33 配置基于 http 的网络 YUM 源仓库
34 [root@ntp ~]# cat /etc/yum.repos.d/dvd.repo
35 [dvd]
36 name=dvd
37 baseurl=http://172.16.89.17/dvd/
38 enabled=1
39 gpgcheck=0
40 [root@ntp ~]#
41 [root@ntp ~]# cat /etc/yum.repos.d/openstack.repo
```

```

42 [RH7-RHOS-6.0]
43 name=RH7-RHOS-6.0
44 baseurl=http://172.16.89.17/openstack/RH7-RHOS-6.0
45 enabled=1
46 gpgcheck=0
47
48 [RH7-RHOS-6.0-Installer]
49 name=RH7-RHOS-6.0-Installer
50 baseurl=http://172.16.89.17/openstack/RH7-RHOS-6.0-Installer
51 enabled=1
52 gpgcheck=0
53
54 [RHEL7-Errata]
55 name=RHEL7-Errata
56 baseurl=http://172.16.89.17/openstack/RHEL7-Errata
57 enabled=1
58 gpgcheck=0
59
60 [RHEL-7-RHSC-1.2]
61 name=RHEL-7-RHSC-1.2
62 baseurl=http://172.16.89.17/openstack/RHEL-7-RHSC-1.2
63 enabled=1
64 gpgcheck=0
65
66
67 清空本地 YUM 缓存数据
68 [root@ntp ~]# yum clean all
69
70 重建 YUM 缓存并测试网络 YUM 源仓库可用性
71 [root@ntp ~]# yum repolist all
72 Loaded plugins: product-id, subscription-manager
73 This system is not registered to Red Hat Subscription Management. You can
74 use subscription-manager to register.
75 Repodata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast
76 repo id                repo name
77      status
78 !RH7-RHOS-6.0          RH7-RHOS-6.0
79      enabled:    393
80 !RH7-RHOS-6.0-Installer RH7-RHOS-6.0-Installer
81      enabled:    194
82 !RHEL-7-RHSC-1.2      RHEL-7-RHSC-1.2
83      enabled: 1,766
84 !RHEL7-Errata         RHEL7-Errata
85      enabled:     11
86 !dvd                  dvd
87      enabled: 4,371
88 repolist: 6,735

```

2-1 Shell脚本（能看懂，能写简单的）

2-2 定时计划任务，周期任务