一、环境准备

还原快照，开启CentOS7虚拟机

昨日习题：

案例1：复制、粘贴、移动

以root用户新建/example/目录，在此目录下新建nsd.txt文件，并进一步完成下列操作

1）将“I love Study”写入到文件nsd.txt

2）将nsd.txt重命名为mylove.txt

3）将/etc/passwd、/boot、/etc/group同时拷贝到/example/目录下

4）将ifconfig命令的前两行内容，追加写入mylove.txt

5）将主机名永久配置文件，拷贝到/example/目录下

6）将DNS永久配置文件，拷贝到/example/目录下

7）将开机自动挂载配置文件，拷贝到/example/目录下

[root@A ~]# mkdir /example

[root@A ~]# echo I love Study > /example/nsd.txt

[root@A ~]# cat /example/nsd.txt

[root@A ~]# mv /example/nsd.txt /example/mylove.txt

[root@A ~]# cp -r /etc/passwd /boot /etc/group /example

[root@A ~]# ls /example

[root@A ~]# ifconfig | head -2

[root@A ~]# ifconfig | head -2 >> /example/mylove.txt

[root@A ~]# cp /etc/hostname /example

[root@A ~]# ls /example

[root@A ~]# cp /etc/resolv.conf /example

[root@A ~]# ls /example

[root@A ~]# cp /etc/fstab /example

[root@A ~]# ls /example

案例2:虚拟机上操作,（MBR分区模式）规划分区

添加一块60G的硬盘并规划分区：

划分2个10G的主分区；1个12G的主分区;1个20G的逻辑分区。

[root@A ~]# lsblk

[root@A ~]# fdisk /dev/sdd

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +10G

p 查看分区表

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +10G

p 查看分区表

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +12G

p 查看分区表

n 创建扩展分区 ----->回车---->起始回车----->结束回车 将所有空间给扩展分区

p 查看分区表

n 创建逻辑分区----->起始回车------>结束+20G

p 查看分区表

w 保存并退出

[root@A ~]# partprobe

[root@A ~]# lsblk

[root@A ~]# ls /dev/sdd[1-5]

案例3:虚拟机上操作,分区使用

1、案例2中新添加60G硬盘的第一个逻辑分区

– 格式化成xfs文件系统，实现该分区开机自动挂载，挂载点为/mnt/xfs

2、案例2中新添加60G硬盘的第一个主分区

– 完成开机自动挂载，挂载点/mnt/mypart,文件系统为ext4

[root@A ~]# mkdir /mnt/xfs

[root@A ~]# mkfs.xfs /dev/sdd5

[root@A ~]# vim /etc/fstab

/dev/sdd5 /mnt/xfs xfs defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

[root@A ~]# mkdir /mnt/mypart

[root@A ~]# mkfs.ext4 /dev/sdd1

[root@A ~]# vim /etc/fstab

/dev/sdd1 /mnt/mypart ext4 defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

案例4:虚拟机上操作,采用GPT分区模式，利用parted规划分区

添加一块20G的硬盘并规划分区：

划分2个2G的主分区；1个5G的主分区;

[root@A ~]# lsblk

[root@A ~]# parted /dev/sde

(parted) mktable gpt

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 0

结束点？ 2G

警告: The resulting partition is not properly aligned for best performance.

忽略/Ignore/放弃/Cancel? Ignore #选择忽略，给出存放分区表信息的空间

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 2G

结束点？ 4G

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 4G

结束点？ 9G

(parted) print

(parted) quit

[root@A ~]# lsblk

[root@A ~]# ls /dev/sde[1-3]

案例5:虚拟机上操作,交换分区使用

1、案例4中新添加20G硬盘的第一个主分区

– 格式化成ext4文件系统，实现该分区开机挂载到/stu01

2、案例4中新添加20G硬盘的第二个主分区

– 格式化成xfs文件系统，实现该分区开机挂载到/stu02

[root@A ~]# mkdir /stu01

[root@A ~]# mkfs.ext4 /dev/sde1

[root@A ~]# vim /etc/fstab

/dev/sde1 /stu01 ext4 defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

[root@A ~]# mkdir /stu02

[root@A ~]# mkfs.xfs /dev/sde2

[root@A ~]# vim /etc/fstab

/dev/sde2 /stu02 xfs defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

案例6:虚拟机上操作,交换分区使用

1、案例4中新添加20G硬盘的第三个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

[root@A ~]# swapon -s

[root@A ~]# mkswap /dev/sde3

[root@A ~]# blkid /dev/sde3

[root@A ~]# vim /etc/fstab

/dev/sde3 swap swap defaults 0 0

[root@A ~]# swapon -a

[root@A ~]# swapon -s

二、交换空间

思路：利用文件所占用的硬盘空间，充当交换空间

1.建立一个大文件

–dd  if=源设备   of=目标设备   bs=块大小  count=块数

]# ls /dev/zero #永远产生无意义数据

/dev/zero

]# dd if=/dev/zero of=/opt/swap.txt bs=1M count=2048

]# du -sh /opt/swap.txt

2.0G /opt/swap.txt

2.利用文件制作交换空间

[root@localhost ~]# mkswap /opt/swap.txt

[root@localhost ~]# swapon /opt/swap.txt

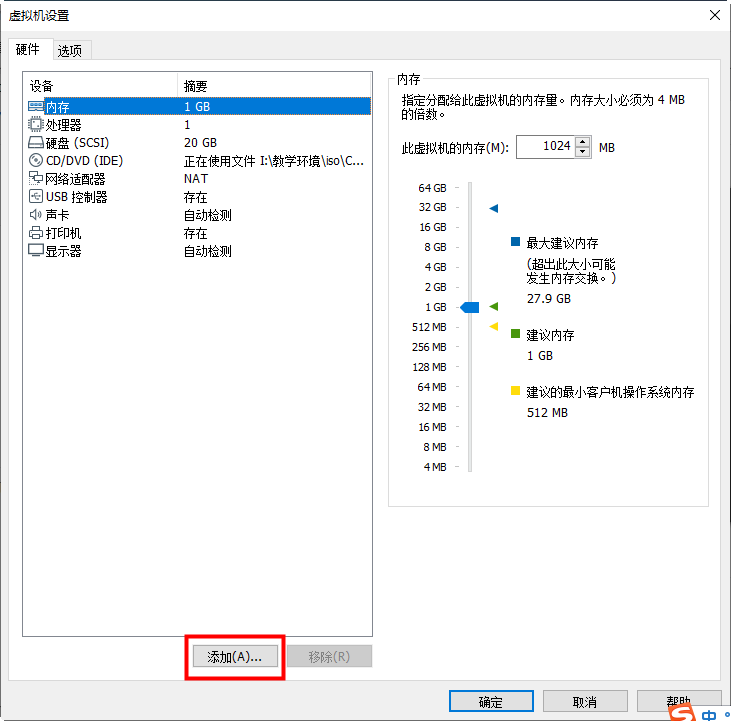
swapon: /opt/swap.txt：不安全的权限 0644，建议使用 0600。

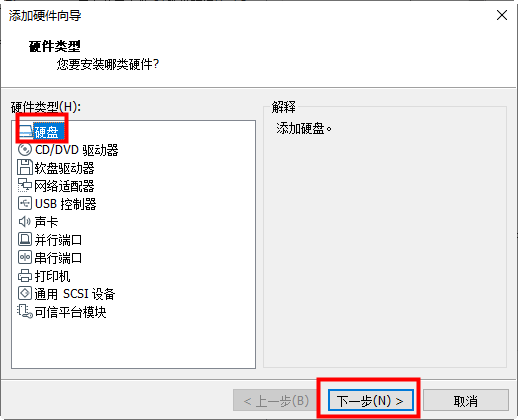
[root@localhost ~]# swapon

三、综合分区，添加一块新80G的硬盘

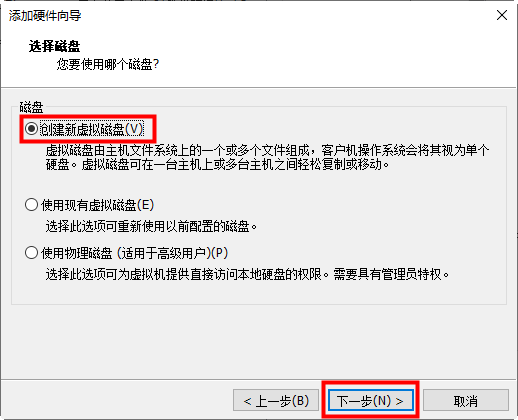
[root@localhost ~]# poweroff

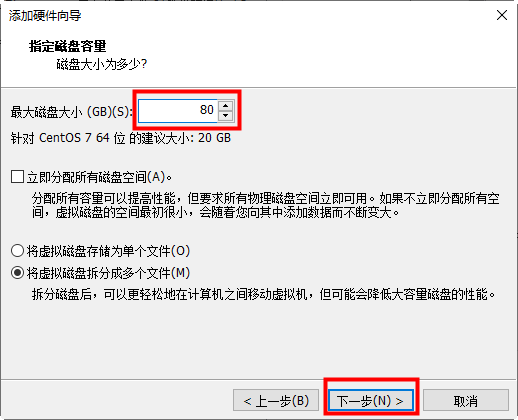


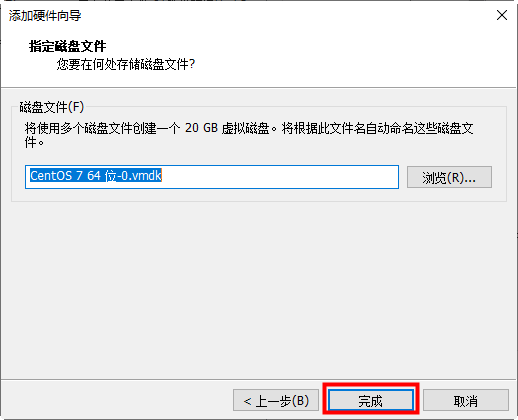














[root@localhost ~]# lsblk

sdb 8:16 0 80G 0 disk

[root@localhost ~]# fdisk /dev/sdb #三个主分区10G 一个逻辑分区20G

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +10G

p 查看分区表

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +10G

p 查看分区表

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +10G

p 查看分区表

**n 创建扩展分区 ----->回车---->起始回车----->结束回车 将所有空间给扩展分区**

p 查看分区表

n 创建逻辑分区----->起始回车------>结束+20G

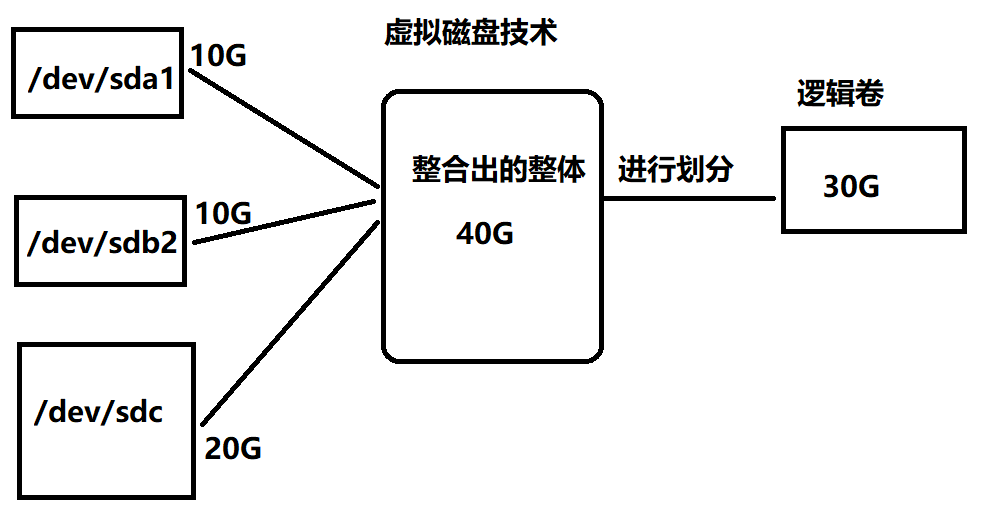
p 查看分区表

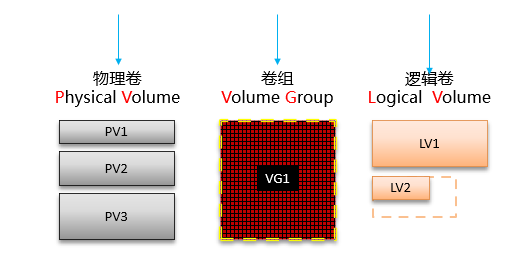
w 保存并退出

[root@localhost ~]# lsblk

四、逻辑卷管理

作用：1.整合分散空间（参与者为分区或硬盘）2.空间可以扩展





制作思路：将众多的物理卷组建成卷组，再从卷组中划分出逻辑卷

|  |  |  |  |
| --- | --- | --- | --- |
| **功能** | **物理卷管理** | **卷组管理** | **逻辑卷管理** |
| Scan 扫描 | pvscan | vgscan | lvscan |
| Create 创建 | pvcreate | vgcreate | lvcreate |
| Display 显示 | pvdisplay | vgdisplay | lvdisplay |
| Remove 删除 | pvremove | vgremove | lvremove |
| Extend 扩展 | / | vgextend | lvextend |

五、逻辑卷的建立

1.制作卷组

格式：vgcreate 卷组的名字 设备路径......

successfully:成功

[root@localhost ~]# vgcreate systemvg /dev/sdb1 /dev/sdb2

[root@localhost ~]# pvs #查看系统中物理卷基本信息

[root@localhost ~]# vgs #查看系统中卷组基本信息

2.制作逻辑卷

格式：lvcreate -L 逻辑卷的大小 -n 逻辑卷的名字 卷组的名字

[root@localhost ~]# lvcreate -L 16G -n vo systemvg

Logical volume "vo" created.

[root@localhost ~]# vgs #查看卷组信息

[root@localhost ~]# lvs #查看逻辑卷信息

[root@localhost ~]# lsblk

3.逻辑卷的使用

[root@localhost ~]# ls /dev/systemvg/vo #查看逻辑卷设备文件

[root@localhost ~]# ls -l /dev/systemvg/vo

[root@localhost ~]# mkfs.xfs /dev/systemvg/vo #格式化

[root@localhost ~]# blkid /dev/systemvg/vo #查看文件系统类型

[root@localhost ~]# vim /etc/fstab

/dev/systemvg/vo /mylv xfs defaults 0 0

[root@localhost ~]# mount -a

mount: 挂载点 /mylv 不存在

[root@localhost ~]# mkdir /mylv

[root@localhost ~]# mount -a #检测

[root@localhost ~]# df -h #查看正在挂载的所有设备信息

六、逻辑卷的扩展

情况一：卷组有足够的剩余空间

1.扩展空间的大小

[root@localhost ~]# vgs

[root@localhost ~]# lvs

[root@localhost ~]# lvextend -L 18G /dev/systemvg/vo

[root@localhost ~]# lvs

2.扩展文件系统的大小（刷新文件系统）

xfs\_growfs:扩展xfs文件系统

resize2fs:扩展ext4文件系统

[root@localhost ~]# df -h

[root@localhost ~]# xfs\_growfs /dev/systemvg/vo

[root@localhost ~]# df -h

情况二：卷组没有足够的剩余空间

1.扩展卷组空间

[root@localhost ~]# vgextend systemvg /dev/sdb3 /dev/sdb5

[root@localhost ~]# vgs

2.扩展空间的大小

[root@localhost ~]# lvextend -L 25G /dev/systemvg/vo

[root@localhost ~]# lvs

3.扩展文件系统的大小（刷新文件系统）

[root@localhost ~]# df -h

[root@localhost ~]# xfs\_growfs /dev/systemvg/vo

[root@localhost ~]# df -h

七、逻辑卷的删除

删除卷组时：需要首先删除基于此卷组创建的所有逻辑卷

[root@localhost ~]# lvremove /dev/systemvg/vo #逻辑卷正在使用

Logical volume systemvg/vo contains a filesystem in use.

[root@localhost ~]# umount /mylv/ #卸载

[root@localhost ~]# lvremove /dev/systemvg/vo

Do you really want to remove active logical volume systemvg/vo? [y/n]: y

Logical volume "vo" successfully removed

[root@localhost ~]# lvs

[root@localhost ~]# vgs

[root@localhost ~]# vgs

[root@localhost ~]# vgremove systemvg #删除卷组

Volume group "systemvg" successfully removed

[root@localhost ~]# vgs

[root@localhost ~]# vim /etc/fstab

/dev/systemvg/vo /mylv xfs defaults 0 0 #删除整行内容

八、卷组划分空间单位（了解）

卷组划分空间单位为PE，默认大小为4M

1PE=4M

[root@localhost ~]# vgcreate systemvg /dev/sdb1

Volume group "systemvg" successfully created

[root@localhost ~]# vgdisplay systemvg #查看卷组详细信息

PE Size 4.00 MiB

[root@localhost ~]# vgchange -s 1M systemvg

Volume group "systemvg" successfully changed

[root@localhost ~]# vgdisplay systemvg #查看卷组详细信息

PE Size 1.00 MiB

•创建逻辑卷的时候指定PE个数

–lvcreate  -l  PE个数  -n  逻辑卷名  卷组名

[root@localhost ~]# lvcreate -l 2 -n nb systemvg #由2个PE组成

Logical volume "nb" created.

[root@localhost ~]# lvs

九、RAID磁盘阵列（通过服务器硬件RAID卡实现）

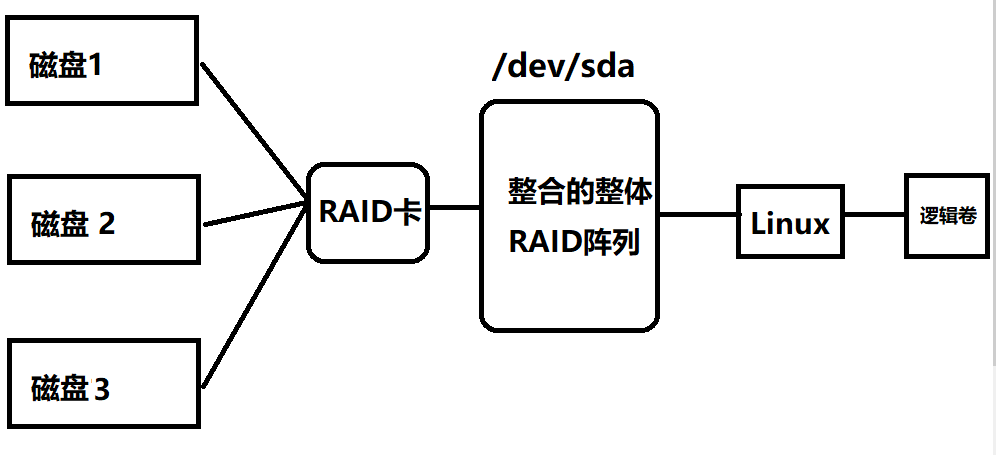
•廉价冗余磁盘阵列

–Redundant Arrays of Inexpensive Disks

–通过硬件/软件技术，将多个较小/低速的磁盘整合成一个大磁盘

–阵列的价值：提升I/O效率、硬件级别的数据冗余

–不同RAID级别的功能、特性各不相同



•RAID 0，条带模式

–同一个文档分散存放在不同磁盘

–并行写入以提高效率

–至少需要两块磁盘

•RAID 1，镜像模式

–一个文档复制成多份，分别写入不同磁盘

–多份拷贝提高可靠性，效率无提升

–至少需要两块磁盘

•RAID5，高性价比模式

–相当于RAID0和RAID1的折中方案

–需要至少一块磁盘的容量来存放校验数据

–至少需要三块磁盘

•RAID6，高性价比/可靠模式

–相当于扩展的RAID5阵列，提供2份独立校验方案

–需要至少两块磁盘的容量来存放校验数据

–至少需要四块磁盘

•RAID 0+1与 RAID 1+0

–整合RAID 0、RAID 1的优势

–并行存取提高效率、镜像写入提高可靠性

–至少需要四块磁盘

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **对比项** | **RAID 0** | **RAID 1** | **RAID 10** | **RAID 5** | **RAID 6** |
| 磁盘数 | ≧ 2 | ≧ 2 | ≧ 4 | ≧ 3 | ≧ 4 |
| 存储利用率 | 100% | ≦ 50% | ≦ 50% | n-1/n | n-2/n |
| 校验盘 | 无 | 无 | 无 | 1 | 2 |
| 容错性 | 无 | 有 | 有 | 有 | 有 |
| IO性能 | 高 | 低 | 中 | 较高 | 较高 |

•硬RAID：由RAID控制卡管理阵列

–主板---> 阵列卡--->磁盘 --->操作系统--->数据

•软RAID：由操作系统来管理阵列

–主板---> 磁盘---> 操作系统--->RAID软件--->数据

十、进程管理

程序：静态的代码 占用硬盘空间

进程：动态的代码 占用内存与CPU资源

进程的唯一标识：PID 编号

父进程与子进程 整体也是树型结构

僵尸进程：运行异常的进程，假死状态

•查看进程树pstree — Processes Tree

–格式：pstree  [选项]  [PID或用户名]

•常用命令选项

–-a：显示完整的命令行

–-p：列出对应PID编号

systemd：上帝进程（所有进程的父进程）

[root@localhost ~]# pstree -p

[root@localhost ~]# pstree lisi

bash───vim

[root@localhost ~]# pstree -p lisi

bash(8335)───vim(8373)

[root@localhost ~]# pstree -ap lisi

bash,8335

└─vim,8373 a.txt

[root@localhost ~]#

•查看进程快照ps — Processes Snapshot

–格式：ps  [选项]...

•常用命令选项

–aux：显示当前终端所有进程（a）、当前用户在所有终端下的进程（x）、以用户格式输出（u）

–-elf：显示系统内所有进程（-e）、以长格式输出（-l）信息、包括最完整的进程信息（-f）

•ps  aux 操作，显示的信息非常完善

用户  进程ID  %CPU  %内存  虚拟内存  固定内存  终端  状态  起始时间  CPU时间  程序指令

•ps  -elf 操作，显示其中有PPID列，为父进程的PID

•进程动态排名top 交互式工具

–格式：top  [-d  刷新秒数]  [-U  用户名]

[root@localhost ~]# top -d 1

按P(大写)按照占用CPU进行排序

按M(大写)按照占用内存进行排序

•检索进程pgrep — Process Grep

–用途：pgrep  [选项]...  查询条件

•常用命令选项

–-l：输出进程名，而不仅仅是 PID

–-U：检索指定用户的进程

–-x：精确匹配完整的进程名

[root@localhost ~]# pgrep a

[root@localhost ~]# pgrep -l a #过滤进程名字包含a的

[root@localhost ~]# pgrep -lU lisi #查找lisi用户开启的进程

[root@localhost ~]# pstree -ap lisi

[root@localhost ~]# pgrep -x crond #严格匹配进程名为crond

[root@localhost ~]# pgrep -lx crond

•**进程的前后台调度**

•后台启动

–在命令行末尾添加“&”符号，不占用当前终端。正在运行放入后台

•Ctrl + z 组合键

–挂起当前进程（暂停并转入后台）

•jobs 命令

–查看后台任务列表

•fg 命令

–将后台任务恢复到前台运行

•bg 命令

–激活后台被挂起的任务

[root@localhost ~]# sleep 2000 & #正在运行的进程放入后台

[root@localhost ~]# jobs #查看后台的进程

[root@localhost ~]# sleep 1000

^Z #按Ctrl +z 将正在运行的进程，暂停放入后台

[2]+ 已停止 sleep 1000

[root@localhost ~]# jobs

[root@localhost ~]# bg 2 #将后台编号为2的进程，在后台继续运行

[root@localhost ~]# jobs #查看后台的进程

[root@localhost ~]# fg 1 #将后台编号为1的进程，恢复到前台运行

sleep 2000

^C #按Ctrl+c结束正在运行的进程

[root@localhost ~]# jobs

[root@localhost ~]# fg 2 #将后台编号为2的进程，恢复到前台运行

sleep 1000

^C

•干掉进程的不同方法

–Ctrl+c 组合键，中断当前命令程序

–kill  [-9]  PID... 、kill  [-9]  %后台任务编号

–killall  [-9]  进程名...

–pkill [-9]    查找条件

[root@localhost ~]# sleep 1000 &

[root@localhost ~]# sleep 1000 &

[root@localhost ~]# sleep 1000 &

[root@localhost ~]# jobs -l

[root@localhost ~]# kill 11723

[1] 已终止 sleep 1000

[root@localhost ~]# jobs -l

[root@localhost ~]# kill -9 11730

[2]- 已杀死 sleep 1000

•杀死一个用户开启的所有进程（强制踢出一个用户）

[root@localhost ~]# killall -9 -u lisi

十一、VDO 了解内容

•Virtual Data Optimizer（虚拟数据优化器）

–一个内核模块，目的是通过重删减少磁盘的空间占用，以及减少复制带宽

–VDO是基于块设备层之上的，也就是在原设备基础上映射出mapper虚拟设备，然后直接使用即可

•重复数据删除

–输入的数据会判断是不是冗余数据

–判断为重复数据的部分不会被写入，然后对源数据进行更新，直接指向原始已经存储的数据块即可

•压缩

–对每个单独的数据块进行处理

[root@svr7 ~]# yum  -y  install  vdo  kmod-kvdo    #所需软件包

•制作VDO卷

•vdo基本操作：参考man vdo 全文查找/example

–vdo  create  --name=VDO卷名称  --device=设备路径 --vdoLogicalSize=逻辑大小

–vdo  list

–vdo  status  -n  VDO卷名称

–vdo  remove  -n  VDO卷名称

–vdostatus  [--human-readable] [/dev/mapper/VDO卷名称]

•VDO卷的格式化加速（跳过去重分析）：

–mkfs.xfs  –K   /dev/mapper/VDO卷名称

–mkfs.ext4  -E  nodiscard  /dev/mapper/VDO卷名称

前提制作VDO需要2G以上的内存

[root@nb ~]# vdo create --name=vdo0 --device=/dev/sdc --vdoLogicalSize=200G

[root@nb ~]# mkfs.xfs -K /dev/mapper/vdo0

[root@nb ~]# mkdir /nsd01

[root@nb ~]# mount /dev/mapper/vdo0 /nsd01

[root@nb ~]# df -h

[root@nb ~]# vdostats --hum /dev/mapper/vdo0 #查看vdo设备详细信息

[root@svr7 ~]# vim /etc/fstab

/dev/mapper/vdo0  /nsd01  xfs  defaults,\_netdev 0 0

课后习题：

案例1：复制、粘贴、移动

以root用户新建/exam/目录，在此目录下新建king.txt文件，并进一步完成下列操作

1）将“I Love hehe”写入到文件king.txt

2）将king.txt重命名为my.txt

3）将/etc/passwd、/boot、/etc/group同时拷贝到/exam/目录下

4）将ifconfig命令的前两行内容，追加写入king.txt

5）将主机名永久配置文件，拷贝到/exam/目录下

6）将存放组基本信息的配置文件，拷贝到/exam/目录下

7）将开机自动挂载配置文件，拷贝到/exam/目录下

[root@www ~]# mkdir /exam

[root@www ~]# touch /exam/king.txt

[root@www ~]# vim /exam/king.txt

[root@www ~]# cat /exam/king.txt

I Love hehe

[root@www ~]# mv /exam/king.txt /exam/my.txt

[root@www ~]# ls /exam/

my.txt

[root@www ~]# cp -r /etc/passwd /boot /etc/group /exam/

[root@www ~]# ls /exam/

boot group my.txt passwd

[root@www ~]# ifconfig | head -2

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

ether 00:0c:29:ba:cf:14 txqueuelen 1000 (Ethernet)

[root@www ~]# ifconfig | head -2 > /exam/king.txt

[root@www ~]# cat /exam/king.txt

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

ether 00:0c:29:ba:cf:14 txqueuelen 1000 (Ethernet)

[root@www ~]# cp /etc/hostname /exam/

[root@www ~]# ls /exam

boot group hostname king.txt my.txt passwd

[root@www ~]# cp /etc/group /exam/

cp：是否覆盖"/exam/group"？ y

[root@www ~]# ls /exam/

boot group hostname king.txt my.txt passwd

[root@www ~]# cp /etc/fstab /exam/

[root@www ~]# ls /exam/

boot fstab group hostname king.txt my.txt passwd

[root@www ~]#

案例2:虚拟机上操作,采用GPT分区模式，利用parted规划分区

添加一块30G的硬盘并规划分区：

划分2个2G的主分区；1个5G的主分区;

parted /dev/sdb

GNU Parted 3.1

使用 /dev/sdb

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) mktable

新的磁盘标签类型？ gpt

(parted) mkpart

分区名称？ []? sdb

文件系统类型？ [ext2]? xfs

起始点？ 0

结束点？ 2G

警告: The resulting partition is not properly aligned for best

performance.

忽略/Ignore/放弃/Cancel? Ignore

(parted) mkpart

分区名称？ []? sdb2

文件系统类型？ [ext2]? xfs

起始点？ 2G

结束点？ 4G

(parted) mkpart

分区名称？ []? sdb3

文件系统类型？ [ext2]? xfs

起始点？ 4G

结束点？ 9G

(parted) print

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdb: 32.2GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Disk Flags:

Number Start End Size File system Name 标志

1 17.4kB 2000MB 2000MB sdb

2 2001MB 4000MB 2000MB sdb2

3 4000MB 9000MB 5000MB sdb3

(parted) w

align-check TYPE N check partition N for

TYPE(min|opt) alignment

help [COMMAND] print general help, or

help on COMMAND

mklabel,mktable LABEL-TYPE create a new disklabel

(partition table)

mkpart PART-TYPE [FS-TYPE] START END make a partition

name NUMBER NAME name partition NUMBER

as NAME

print [devices|free|list,all|NUMBER] display the partition

table, available devices, free space, all found

partitions, or a particular partition

quit exit program

rescue START END rescue a lost partition

near START and END

resizepart NUMBER END resize partition NUMBER

rm NUMBER delete partition NUMBER

select DEVICE choose the device to

edit

disk\_set FLAG STATE change the FLAG on

selected device

disk\_toggle [FLAG] toggle the state of

FLAG on selected device

set NUMBER FLAG STATE change the FLAG on

partition NUMBER

toggle [NUMBER [FLAG]] toggle the state of

FLAG on partition NUMBER

unit UNIT set the default unit to

UNIT

version display the version

number and copyright information of GNU Parted

(parted) q

信息: You may need to update /etc/fstab.

[root@www ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 20G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 19G 0 part

├─centos-root

253:0 0 17G 0 lvm /

└─centos-swap

253:1 0 2G 0 lvm [SWAP]

sdb 8:16 0 30G 0 disk

├─sdb1 8:17 0 1.9G 0 part

├─sdb2 8:18 0 1.9G 0 part

└─sdb3 8:19 0 4.7G 0 part

sr0 11:0 1 8.8G 0 rom /run/media/root/CentOS 7 x86\_64

[root@www ~]#

案例3:虚拟机上操作,交换分区使用

1、案例2中新添加30G硬盘的第一个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

2、案例2中新添加30G硬盘的第二个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

mkswap /dev/sdb1

正在设置交换空间版本 1，大小 = 1953104 KiB

无标签，UUID=e76d4dd0-e60f-4230-a420-bcbe17984272

[root@www ~]# swapon /dev/sdb1

[root@www ~]# swapon

NAME TYPE SIZE USED PRIO

/dev/dm-1 partition 2G 53.5M -1

/dev/sdb1 partition 1.9G 0B -2

[root@www ~]# mkswap /dev/sdb2

正在设置交换空间版本 1，大小 = 1952764 KiB

无标签，UUID=cfa61ddf-215f-456f-81bd-8ecb24fd44f2

[root@www ~]# swapon /dev/sdb2

[root@www ~]# swapon

NAME TYPE SIZE USED PRIO

/dev/dm-1 partition 2G 53.5M -1

/dev/sdb1 partition 1.9G 0B -2

/dev/sdb2 partition 1.9G 0B -3

[root@www ~]# vim /etc/fstab

[root@www ~]# swapon

NAME TYPE SIZE USED PRIO

/dev/dm-1 partition 2G 62M -1

/dev/sdb1 partition 1.9G 0B -2

/dev/sdb2 partition 1.9G 0B -3

[root@www ~]# swapoff

用法：

swapoff [选项] [<指定>]

选项：

-a, --all 禁用 /proc/swaps 中的所有交换区

-v, --verbose verbose mode

-h, --help 显示此帮助并退出

-V, --version 输出版本信息并退出

<指定> 参数包括：

-L <标签> 要使用设备的标签

-U <uuid> 要使用设备的 UUID

LABEL=<标签> 要使用设备的标签

UUID=<uuid> 要使用设备的 UUID

<设备> 要使用设备的名称

<文件> 要使用文件的名称

更多信息请参阅 swapoff(8)。

[root@www ~]# swapoff /dev/sdb1

[root@www ~]# swapoff /dev/sdb2

[root@www ~]# swapon

NAME TYPE SIZE USED PRIO

/dev/dm-1 partition 2G 64.3M -1

[root@www ~]# swapon -a

[root@www ~]# swapon

NAME TYPE SIZE USED PRIO

/dev/dm-1 partition 2G 64.3M -1

/dev/sdb1 partition 1.9G 0B -2

/dev/sdb2 partition 1.9G 0B -3

[root@www ~]#

案例4:虚拟机上操作,文件扩展Swap空间

1. 使用dd命令创建一个大小为2048MB的交换文件，放在/opt/swap.db

dd if=/dev/zero of=/opt/swap.db bs=1M count=2048

1. 将swap.db文件格式化成swap文件系统

mkswap /opt/swap.db

1. 启用swap.db文件，查看swap空间组成

swapon /opt/swap.db

4. 停用swap.db文件，查看swap空间组成

案例5:虚拟机操作，构建 LVM 存储

– 新建一个名为 systemvg 的卷组

– 在此卷组中创建一个名为 vo 的逻辑卷，大小为8G

– 将逻辑卷 vo 格式化为 xfs 文件系统

– 将逻辑卷 vo 挂载到 /vo 目录，并在此目录下建立一个测试文件 votest.txt，内容为“I AM KING.”

– 实现逻辑卷vo开机自动挂载到/vo

案例6:虚拟机操作，构建 LVM 存储(修改PE大小)

– 新的逻辑卷命名为 database，其大小为50个PE的大小，属于 datastore 卷组

– 在 datastore 卷组中其PE的大小为1M

– 使用 EXT4 文件系统对逻辑卷 database 格式化，此逻辑卷应该在开机时自动挂载到 /mnt/database 目录

案例7:虚拟机 server0操作，扩展逻辑卷

– 将/dev/systemvg/vo逻辑卷的大小扩展到20G

案例8:进程管理

1.查看当前系统中整个进程树信息

2.利用pstree查看lisi开启的进程

3.显示当前系统正在运行的所有进程信息

4.检索当前系统中进程，进程名包含cron的PID是多少？

5.开启5个sleep 2000放入后台运行

6.杀死所有sleep进程