

Linux 云计算集群架构师

学神 IT 教育：从零基础到实战，从入门到精通！

版权声明：

本系列文档为《学神 IT 教育》内部使用教材和教案，只允许 VIP 学员个人使用，禁止私自传播。否则将取消其 VIP 资格，追究其法律责任，请知晓！

免责声明：

本课程设计目的只用于教学，切勿使用课程中的技术进行违法活动，学员利用课程中的技术进行违法活动，造成的后果与讲师本人及讲师所属机构无关。倡导维护网络安全人人有责，共同维护网络文明和谐。

联系方式：

学神 IT 教育官方网站: <http://www.xuegod.cn>

Linux 云计算架构师进阶学习群 QQ 群: 1072932914



学习顾问：小语老师

学习顾问：边边老师

学神微信公众号

微信扫码添加学习顾问微信，同时扫码关注学神公众号了解最新动态，获取更多学习资料及答疑就业服务！

第三章 文件的基本管理和 XFS 文件系统备份恢复

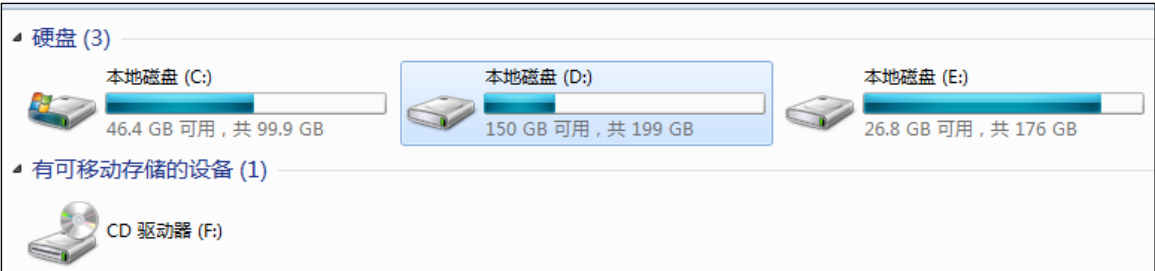
本节所讲内容:

- 3.1 Linux 系统目录结构和相对路径与绝对路径
- 3.2 文件的管理
- 3.3 查看文件
- 3.4 实战: xfs 文件系统的备份和恢复

3.1 Linux 系统目录结构和相对路径与绝对路径

3.1.1 系统目录结构

在 WIN 系统中, 查看文件先进入相应的盘符, 然后进入文件目录



在 WIN 中, 它是多根 c:\ d:\ e:\

Linux 只有一个/根目录

使用 tree 命令查看 linux 目录结构, 如果没有这个命令, 需要手工安装一下

[root@xuegod63 ~]# yum install tree #安装 tree 命令

查看/tmp 目录结构

[root@xuegod63 ~]# tree /tmp/

查看根下有哪些文件:

[root@xuegod63 ~]# ls /

bin dev home lib64 mnt proc run srv tmp var

boot etc lib media opt root sbin sys usr

根下的目录作用说明:

目 录	说 明
/	处于 linux 系统树形结构的最顶端, 它是 linux 文件系统的入口, 所有的目录、文件、设备都在 / 之下
/bin	bin 是 Binary 的缩写。常用的二进制命令目录。比如 ls、cp、mkdir、cut 等; 和/usr/bin 类似, 一些用户级工具
/boot	存放的系统启动相关的文件, 例如: kernel.grub(引导装载程序)
/dev	dev 是 Device 的缩写。设备文件目录, 比如声卡、磁盘.....在 Linux 中一切都被看做文件。终端设备、磁盘等等都被看做文件 设备文件: /dev/sda,/dev/sda1,/dev/tty1,/dev/tty2,/dev/pts/1, /dev/zero, /dev/null, /dev/cdrom
/etc	常用系统及二进制安装包配置文件默认路径和服务器启动命令目录, 如: /etc/passwd 用户信息文件 /etc/shadow 用户密码文件 /etc/group 存储用户组信息

	<p>/etc/fstab 系统开机启动自动挂载分区列表</p> <p>/etc/hosts 设定用户自己的 IP 与主机名对应的信息</p>
/home	普通用户的家目录默认存放目录
/lib	库文件存放目录,函数库目录
/mnt /media	<p>/mnt 和/media 一般用来临时挂载存储设备的挂载目录, 比如有 cdrom、U 盘等目录</p> <p>/mnt/hgfs: 安装 vmware-tools 后, 使用共享文件夹功能, 可以共享虚拟机和真机的目录, 实现文件相互复制, 一般这个功能没有用。我们直接使用 xshell 来上传文件</p>
/opt	表示的是可选择的意思, 有些软件包也会被安装在这里。如: gitlab
/proc	<p>操作系统运行时, 进程 (正在运行中的程序) 信息及内核信息 (比如 cpu、硬盘分区、内存信息等) 存放在这里。/proc 目录是伪装的文件系统 proc 的挂载目录, proc 并不是真正的文件系统。因此, 这个目录是一个虚拟的目录, 它是系统内存的映射, 我们可以通过直接访问这个目录来获取系统信息。也就是说, 这个目录的内容不在硬盘上而是在内存里</p> <p>查看咱们的 CPU 信息</p> <p>cat /proc/cpuinfo</p>
/sys	系统目录, 存放硬件信息的相关文件
/run	运行目录, 存放的是系统运行时的数据, 比如进程的 PID 文件
/srv	服务目录, 存放的是我们本地服务的相关文件
/sbin	大多数涉及系统管理的命令都存放在该目录中, 它是超级权限用户 root 的可执行命令存放地, 普通用户无权限执行这个目录下的命令, 凡是目录 sbin 中包含的命令都是 root 权限才能执行的
/tmp	<p>该目录用于存放临时文件, 有时用户运行程序的时候, 会产生一些临时文件。/tmp 就是用来存放临时文件的。/var/tmp 目录和该目录的作用是相似的, 不能存放重要数据, 系统会定期删除这个目录下的没有被使用的文件。它的权限比较特殊</p> <p>[root@xuegod63 ~]# ls -ld /tmp</p> <p>drwxrwxrwt 10 root root 12288 Oct 3 20:45 /tmp/ → 粘滞位 (sticky bit) 目录的 sticky 位表示这个目录里的文件只能被文件的 owner (所有者) 和 root 删除</p>
/var	系统运行和软件运行时产生的日志信息, 该目录的内容是经常变动的, 存放的是一些变化的文件。比如/var 下有/var/log 目录用来存放系统日志的目录, 还有 mail、/var/spool/cron
/usr	<p>存放应用程序和文件,</p> <p>/usr/bin 普通用户使用的应用程序</p> <p>/usr/sbin 管理员使用的应用程序</p> <p>/usr/lib 库文件 Glibc(32 位)</p> <p>/usr/lib64 库文件 Glibc</p>
/lib /lib64 是 /usr/lib 和 /usr/lib64	<p>这个目录里存放着系统最基本的动态链接共享库, 包含许多被/bin/和/sbin/中的程序使用的库文件, 目录/usr/lib/中含有更多用于用户程序的库文件。作用类似于 windows 里的 DLL 文件, 几乎所有的应用程序都需要用到这些共享库</p> <p>注: lib***.a 是静态库, lib***.so 是动态库</p> <p>静态库在编译时被加载到二进制文件中</p>

的软链接	动态库在运行时加载到进程的内存空间中 简单的说: 这些库是为了让你的程序能够正常编译运行的 其实类似于 WIN 中.dll 文件, 几乎所有的应用程序都需要用到这些共享库
------	---

3.1.2 绝对路径和相对路径

路径: 在我们平时使用计算机时要找到需要的文件就必须知道文件的位置, 而表示文件的位置的方式就是路径

绝对路径: 在 Linux 中, 绝对路径是从 "/" 开始的, 比如 /usr、/etc/passwd。如果一个路径是从根 (/) 开始的, 它一定是绝对路径。

相对路径: 相对路径是以 . 或 .. 开始的,

[root@xuegod63 etc]# pwd #判断用户当前所处的位置

绝对路径: 从/开始的路径 /home/user1

相对路径: 相对于当前目录开始, a.txt ./a.txt ../root/b.txt 当前目录在/etc

[root@xuegod63 ~]# cd /etc/

[root@xuegod63 etc]# ll passwd

-rw-r--r-- 1 root root 2116 11 月 16 14:57 passwd

[root@xuegod63 etc]# ll /etc/passwd

-rw-r--r-- 1 root root 2116 11 月 16 14:57 /etc/passwd

3.2 文件的管理

文件管理方式有多种:

改变目录: cd

创建/修改/移动/删除: touch mkdir mv vi rm cp

3.2.1 创建文件和文件夹

命令之: touch ; touch [t\t] 触摸;接触;

作用: 常用来创建空文件, 如果文件存在, 则修改这个文件的时间

语法: touch 文件名

[root@xuegod63 ~]# cd /opt/

[root@xuegod63 opt]# touch a.txt

[root@xuegod63 opt]# touch file1 file2

[root@xuegod63 opt]# touch file{6..20} #创建 file6 到 file20 的文件

[root@xuegod63 opt]# ls

a.txt file10 file12 file14 file16 file18 file2 file6 file8 rh

file1 file11 file13 file15 file17 file19 file20 file7 file9

[root@xuegod63 ~]# touch a.txt

[root@xuegod63 ~]# ll a.txt

-rw-r--r--. 1 root root 0 8 月 10 21:13 a.txt #这个时间是文件的修改时间。

[root@xuegod63 ~]# stat a.txt #查看文件属性 (其中包括文件时间属性)

文件: "a.txt"

大小: 4 块: 8 IO 块: 4096 普通文件

设备: 803h/2051d Inode: 18217320 硬链接: 1

权限: (0755/-rwxr-xr-x) Uid: (0/ root) Gid: (0/ root)

最近访问: 2021-03-09 23:03:19.747577494 +0800

最近更改: 2021-03-09 23:03:58.216892266 +0800

最近改动: 2021-03-09 23:04:12.116010166 +0800

注:

访问时间: atime 查看内容 cat a.txt

修改时间: mtime 修改内容 vim a.txt

改变时间: ctime 文件属性, 比如权限 change time. chmod +w a.txt

[root@xuegod64 ~]# echo \$LANG #查看当前语言设置

zh_CN.UTF-8

[root@xuegod64 ~]# LANG=en_US.UTF-8 #切换为英文

[root@xuegod63 ~]# stat a.txt

[root@xuegod64 ~]# LANG=zh_CN.UTF-8 #切换为中文

3.2.2 使用 vim 和重定向创建一个新文件

用 vim 命令创建一个新文件

[root@xuegod63 mnt]# vim xuegod-1.txt

用重定向创建一新文件

[root@xuegod63 mnt]# echo aaa > xuegod.txt

3.2.3 命令之: mkdir

作用: 创建目录

语法: mkdir (选项) 文件夹名子

例:

[root@xuegod63 opt]# mkdir dir1

[root@xuegod63 opt]# mkdir dir2 dir3 /home/dir4

[root@xuegod63 opt]# ls /home/

dir4 user1

[root@xuegod63 opt]# mkdir /tmp/a/b/c

mkdir: 无法创建目录"/tmp/a/b/c": 没有那个文件或目录

[root@xuegod63 opt]# mkdir -p /tmp/a/b/c #在创建一个目录的时候, 如果这个目录的上一

级不存在的话, 要加参数-p

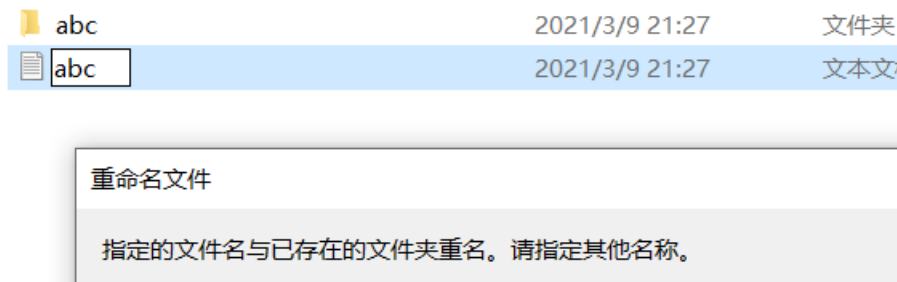
[root@xuegod63 opt]# tree /tmp/a/b/c

互动: 在当前目录下, 已经存在一个文件后, 可以创建同名的目录文件吗?

[root@xuegod63 opt]# touch aaa

[root@xuegod63 opt]# mkdir aaa

#不可以创建成功, 因为目录下不能存在两个同名的文件。



3.2.4 删除文件和目录

用到的命令: `rm`

作用: 可以删除一个目录中的一个或多个文件或目录, 对于链接文件, 只是删除整个链接文件, 而原文件保持不变的

语法: `rm (选项) 处理对象`

选项:

`-f` 强制删除, 没有提示

`-r` 删除目录, 递归删除, 把目录下的文件和文件夹也删除

例子:

```
[root@xuegod63 opt]# rm -rf a.txt
```

```
[root@xuegod63 opt]# rm -rf a.txt dir
```

```
[root@xuegod63 opt]# rm -rf file*
```

`rm -rf` (慎用, 一定要在删除以前确定一下所在目录, 防止误删除重要数据)

3.2.5 复制文件

命令: `cp 源文件/目录 目标文件/目录`

选项: `-R/r`: 递归处理, 将指定目录下的所有文件与子目录一并处理。复制目录时会使用 `r` 参数

例子:

```
[root@xuegod63 ~]# cp /etc/passwd /opt/ #复制文件
```

```
[root@xuegod63 ~]# cp -r /boot/grub2 /opt/ #复制目录
```

```
[root@xuegod64 ~]# cp -r /boot/grub2 /etc/passwd /opt #复制多个文件到目录
```

选项: `\`: 覆盖文件, 不提示

例子:

```
[root@xuegod64 ~]# \cp -r /boot/grub2 /etc/passwd /opt #反斜杠, 覆盖文件, 不提示
```

选项: `-p`: 复制文件会保留文件属性

例子:

```
[root@xuegod64 ~]# touch c.txt
```

```
[root@xuegod64 ~]# chmod a+w c.txt #添加写权限
```

```
[root@xuegod64 ~]# ll c.txt #有写权限
```

```
[root@xuegod64 ~]# cp c.txt /opt/ #复制到目录
```

互动: 复制到目录后, 这个文件会带上 `w` 写权限吗?

```
[root@xuegod64 ~]# ll /opt/c.txt #检查, 没有写权限
```

```
[root@xuegod64 ~]# \cp -p c.txt /opt # -p 选项会保留文件属性
```

```
[root@xuegod64 ~]# ll /opt/c.txt      #检查, 有写权限
[root@xuegod63 opt]# mkdir dir1 dir2 dir3
[root@xuegod63 opt]# \cp -rp/opt/!(dir1 | dir2) /root/  #除了 dir1 或 dir2 以外其他文件
都复制到 root 目录下
```

3.2.6 移动文件

```
[root@xuegod63 ~]# cd /opt
[root@xuegod63 opt]# mv passwd bbbpasswd #重命名
[root@xuegod63 opt]# mv bbbpasswd /opt/passwd #在移动文件的时候支持改名操作
[root@xuegod63 opt]# touch file1 file2 file3 file4
[root@xuegod63 opt]# mv /opt/!(file1 | file2) /root/ #除了 file1 或 file2 以外其他文件都
移动到 root 目录下
```

3.3 查看文件

3.3.1 命令之: cat

语法: cat 文件名
作用: 查看文件内容, 一次显示整个文件的内容
例子: [root@xuegod63 ~]# cat /etc/passwd

3.3.2 命令之: more

作用:以分页形式显示文件内容
语法:more + 文件名
说明: 按下回车刷新一行, 按下空格刷新一屏, 输入 q 键退出
例:
[root@xuegod63 ~]# cat /proc/cpuinfo | more
[root@xuegod63 ~]# ifconfig | more
[root@xuegod63 ~]# more /proc/cpuinfo

3.3.3 命令之: less

作用:和 more 功能一样
语法:less +文件名
说明:linux 中 more 与 less 的区别
more:不支持后退, 但几乎不需要加参数, 空格键是向下翻页, Enter 键是向下翻一行, 在不需要后退的情况下比较方便
less: 支持前后翻滚, 既可以向上翻页 (pageup 按键), 也可以向下翻页 (pagedown 按键), 空格键是向下翻页, Enter 键是向下翻一行

3.3.4 命令之: head

作用: 用于显示文件的开头的内容。在默认情况下, head 命令显示文件的头 10 行内容
语法:head(选项)文件名

参数: -n 显示从文件头开始的行数

```
[root@xuegod63 opt]# head /etc/passwd
```

```
[root@xuegod63 opt]# head -n 3 /etc/passwd #显示前 3 行
```

3.3.5 命令之: tail

作用: 用于显示文件中的尾部内容。默认在屏幕上显示指定文件的末尾 10 行

语法:tail (选项)文件名

参数:

-n 显示文件尾部多少行的内容(n 为数字)

-f 动态显示数据 (不关闭) ,常用来查看日志

```
[root@xuegod63 ~]# tail -n 3 /var/log/secure #查看最后 3 行记录
```

```
[root@xuegod63 ~]# tail -3 /var/log/secure #查看最后 3 行记录
```

```
[root@xuegod63 ~]# tail -f /var/log/secure #在一个终端执行此命令动态查看文件内容
```

```
[root@xuegod63 ~]# tailf /var/log/secure #在一个终端执行此命令动态查看文件内容
```

```
[root@xuegod63 ~]# ssh root@192.168.1.63 #在另一个终端远程登录 Linux, 登录成功后
```

```
[root@xuegod63 ~]# tail -f /var/log/secure #可以动态查看到登录成功的日志
```

```
Nov 17 00:08:32 xuegod63 sshd[2924]: Accepted password for root from 192.168.1.63  
port 39904 ssh2
```

3.4 实战: xfs 文件系统的备份和恢复

XFS 提供了 **xfsdump** 和 **xfsrestore** 工具协助备份 XFS 文件系统的数据。xfsdump 按 inode 顺序备份一个 XFS 文件系统。

centos7 开始选择 xfs 格式作为默认文件系统, 而且不再使用以前的 ext, 仍然支持 ext4, xfs 专为大数据产生, 每个单个文件系统最大可以支持 8EB(1EB=1024PB,1PB=1024TB,1TB=1024GB), 单个文件可以支持 16TB, 不仅数据量大, 而且扩展性高。还可以通过 xfsdump, xfsrestore 来备份和恢复。

与传统的 UNIX 文件系统不同, XFS 不需要在备份前被卸载; 对使用中的 XFS 文件系统做备份就可以保证镜像的一致性。XFS 的备份和恢复的过程是可以被中断然后继续的, 无须冻结文件系统。

xfsdump 甚至提供了高性能的多线程备份操作——它把一次 dump 拆分成多个数据流, **每个数据流可以被发往不同的目的地**

首先了解一下 xfsdump 的备份级别有以下两种, 默认为 0 (即完全备份)

0 级别代表: 完全备份

1 到 9 级别代表: 增量备份

xfsdump level 级别

对完整的文件系统进行备份, 用 xfsdump。

他除了可以进行完整备份, 还可以进行累积性备份。第一次使用 xfsdump, 是完整备份, 定义为 level 0, 而第二次备份, 则不会完整备份, 而是和第一次的备份 (level 0) 进行比较, 仅备份有差异的文件 (level 1)。

xfsdump 常用参数

-L: xfsdump 会记录每次备份的 session Label, 这里可以填写针对此文件系统的简易说明;

-M: xfsdump 可以记录存储 Media Label, 这里可以填写此媒体的简易说明。

-l: 是 L 的小写, 就是指定 level, 有 0~9 共 10 个等级, 默认为 0, 即完整备份。

- f: 后面接产生的文件和 destination file 。例如/dev/st0 设备文件名或其他一般文件文件名
- l: 大写的“i”，从/var/lib/xfsdump/inventory 列出目前备份的信息状态。

扩展:

完全备份: 每次都把指定的备份目录完整的复制一遍, 不管目录下的文件有没有变化; 如: cp

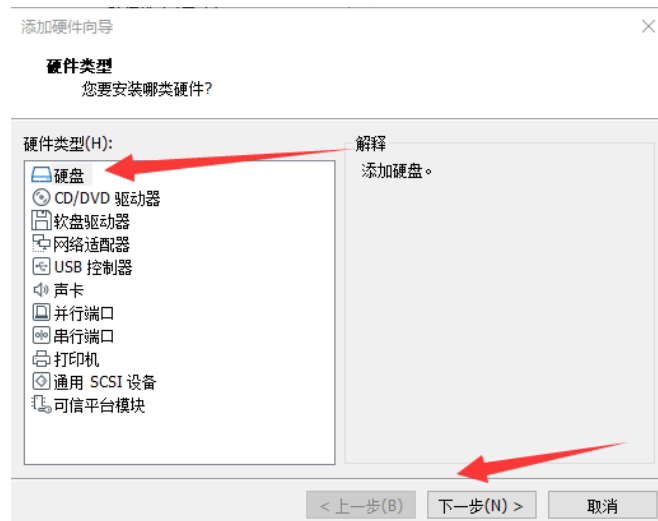
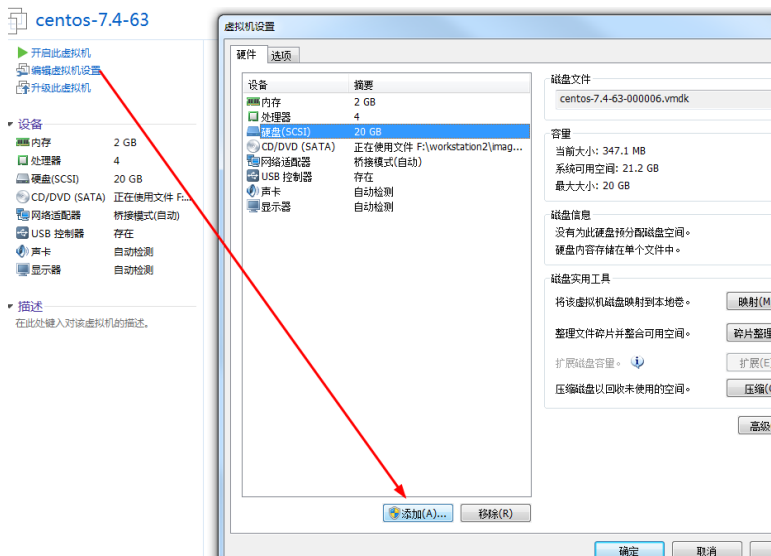
增量备份: 每次将之前 (第一次、第二次、直到前一次) 做过备份之后有变化的文件进行备份;

差异备份: 每次都从第一次完整备份以来有变化的文件进行备份。如: rsync 备份

3.4.1 环境准备

实验环境:

添加磁盘大小: 20G → 分区 → 格式化 → 挂载



选择硬盘, 下一步。



类型选择



默认即可



下一步



位置和名称默认即可



添加第二块硬盘后效果

开机后对新添加的硬盘进行格式化:

```
[root@xuegod63 ~]# fdisk /dev/sdb
```

#指定分区的设备,centos8 第二块硬盘的第一分区名为/dev/sdb1

欢迎使用 fdisk (util-linux 2.23.2)。

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。

Device does not contain a recognized partition table

使用磁盘标识符 0x06d5a427 创建新的 DOS 磁盘标签。

命令(输入 m 获取帮助): **n** #创建一个新的分区

Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

Select (default p): **p** #创建一个主分区

分区号 (1-4, 默认 1): #直接回车

起始 扇区 (2048-41943039, 默认为 2048):

将使用默认值 2048

Last 扇区, +扇区 or +size{K,M,G} (2048-41943039, 默认为 41943039): **+1G** #指定分区

大小

分区 1 已设置为 Linux 类型, 大小设为 1 GiB

命令(输入 m 获取帮助): **p** #打印分区表

磁盘 /dev/sdb: 21.5 GB, 21474836480 字节, 41943040 个扇区

Units = 扇区 of 1 * 512 = 512 bytes

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: 0x06d5a427

设备	启动	起点	末尾	扇区	大小	Id	类型
/dev/sdb1	2048	2099199	2097152	1G	83	Linux	

命令(输入 m 获取帮助): **w** #保存

The partition table has been altered!

Calling ioctl() to re-read partition table.

正在同步磁盘。

```
[root@xuegod63 ~]# ls /dev/sdb*
```

```
/dev/sdb /dev/sdb1
```

使用新的分区, 格式化分区, 并进行挂载

```
[root@xuegod63 ~]# mkfs.xfs /dev/sdb1 #使用 xfs 格式对分区进行格式化
```

```
[root@xuegod63 ~]# blkid /dev/sdb1 #查看分区信息
```

```
[root@xuegod63 ~]# mkdir /test #创建挂载点
```

```
[root@xuegod63 ~]# mount /dev/sdb1 /test #挂载
```

```
[root@xuegod63 ~]# df -hT #或者用 lsblk 命令
```

准备备份测试文件

```
[root@xuegod63 ~]# cd /test/ #进入 test 目录
```

```
[root@xuegod63 test]# cp /etc/passwd ./ #拷贝 passwd 文件到当前目录
```

```
[root@xuegod63 test]# mkdir ./xuegod #在当前目录内创建 xuegod 目录
```

```
[root@xuegod63 test]# touch ./xuegod/a #在当前目录的 xuegod 目录内创建 a 文件
```

```
[root@xuegod63 test]# tree /test/
```

```
/test/
├── passwd
└── xuegod
    └── a
```

3.4.2 备份

1、备份整个分区。(这个功能就像是虚拟机的快照, 服务器被黑后, 进行快速恢复)

语法格式: `xfsdump -f` 这里写备份后保存的位置和 **文件名** 这里写要备份的文件路径或设备

注意: 备份的路径这里不能写成 `/test/`。可以是 `/dev/sdb1` 或 `/test` 即目录后不能有 `/`

```
[root@xuegod63 test]# xfsdump -f /opt/dump_sdb1 /dev/sdb1
```

#备份的文件放在/opt下, 要备份的设备是/dev/sdb1

xfsdump: using file dump (drive_simple) strategy

xfsdump: version 3.1.8 (dump format 3.0) - type ^C for status and control

```
=====dumplabeldialog=====
=====
please enter label for this dump session (timeout in 300 sec)
-> dump_sdb1 这里写对此次备份的说明。如: dump_sdb1 , 可以随意写。

session label entered: "dump_sdb1"
----- end dialog -----
xfsdump: level 0 dump of xuegod63.cn:/test
xfsdump: dump date: Tue Mar 10 16:03:07 2020
xfsdump: session id: e7d1aab1-dfbd-4b77-9379-56d63828a013
xfsdump: session label: "dump_nvme0n2p1"
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: skipping (no pruning necessary)
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 25856 bytes
===== media label dialog =====
please enter label for media in drive 0 (timeout in 300 sec)
-> sdb1 这里写对要备份的设备做一个说明。如: sdb1, 可以随意写。
media label entered: "sdb1"
----- end dialog -----
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 26072 bytes
xfsdump: dump size (non-dir files) : 3104 bytes
xfsdump: dump complete: 27 seconds elapsed
xfsdump: Dump Summary:
xfsdump:  stream 0 /opt/dump_sdb1 OK (success)
xfsdump: Dump Status: SUCCESS
```

扩展: 备份前先查看分区的文件系统类型。

```
[root@xuegod63 ~]# df -T
```

文件系统	类型	1K-块	已用	可用	已用%	挂载点
devtmpfs	devtmpfs	975248	0	975248	0%	/dev
tmpfs	tmpfs	1003548	0	1003548	0%	/dev/shm
tmpfs	tmpfs	1003548	9908	993640	1%	/run
tmpfs	tmpfs	1003548	0	1003548	0%	/sys/fs/cgroup
/dev/sda3	xfs	52403200	5352704	47050496	11%	/
/dev/sda1	ext4	999320	175832	754676	19%	/boot
(发现 boot 默认是 ext4 文件系统类型, 因此 boot 分区不能执行 xfs 备份)						
/dev/sdb1	xfs	1038336	40296	998040	4%	/test

2、指定备份时免交互操作, 方便后期做定时备份

```
[root@xuegod63 test]# xfsdump -f /opt/dump_sdb1_v2 /dev/sdb1 -L dump_sdb1_v2 -M sdb1_v2
```

-L : <session label> 备份的会话标签, 这里写对此次备份的说明。如: dump_sdb1_v2 , 可以随意写。

-M : <media label> 储存设备的标签, 这里写对备份设备的简易描述。如: sdb1_v2 , 可以随意写。

3、指定只备份分区中某个目录

参数: -s 文件路径 只对指定的文件进行备份, -s 指定时, 路径写的是相对路径(-s 可以是文件或目录)

```
[root@xuegod63 ~]# xfsdump -f /opt/xuegod_dump1 -s xuegod /dev/sdb1 -L dump_xuegod -M sdb1
```

4、查看备份信息与内容

备份成功后, 我们就可以在/var/lib/xfsdump/inventory 目录下看到生成的档案信息

```
[root@xuegod63 opt]# xfsdump -I (字母大写 I) 不是 L
```

可以看到 session 信息

```
session 3:
  mount point:  xuegod63.cn:/test
  device:       xuegod63.cn:/dev/sdb1
  time:         Tue Jun 23 22:48:51 2020
  session label: "dump_xuegod"
  session id:   e5bcc655-df6f-465d-9b85-78e12dd7a019
  level:        0
  resumed:      NO
  subtree:      YES
  streams:      1
  path name:    /opt/xuegod_dump1
  start:        ino 133 offset 0
  end:          ino 134 offset 0
  interrupted:  NO
  media files:  1
  media file 0:
    mfile index: 0
    mfile type:  data
    mfile size:  22344
    mfile start: ino 133 offset 0
    mfile end:   ino 134 offset 0
    media label: "sdb1"
    media id:    3018ddf6-d2cf-44e4-
xfsdump: Dump Status: SUCCESS
```

测试恢复: 先删除之前创建的内容

```
[root@xuegod63 test]# ls
```

```
passwd xuegod
```

```
[root@xuegod63 test]# pwd
```

```
/test
```

```
[root@xuegod63 sdb1]# rm -rf ./*
```

3.4.3 文件系统恢复

语法: xfsrestore -f 指定恢复文件的位置 指定存放恢复后的文件的路径

```
[root@xuegod63 opt]# xfsrestore -f /opt/dump_sdb1 /test/
```

#查看恢复情况

```
[root@xuegod63 ~]# ls /test/
```

恢复单个文件如下:

```
[root@xuegod63 ~]# mkdir /var/test/
```

```
[root@xuegod63 ~]# xfsrestore -f /opt/xuegod_dump1 -s xuegod /var/test/ #恢复目录
```

录

查看备份文件的内容:

```
[root@xuegod63 ~]# xfsrestore -f /opt/dump_sdb1 -t
```

注:

使用 xfsdump 时, 请注意下面下面的几个限制:

- 1、xfsdump 不支持没有挂载的文件系统备份! 所以只能备份已挂载的!
- 2、xfsdump 必须使用 root 的权限才能操作 (涉及文件系统的关系)
- 3、xfsdump 只能备份 XFS 文件系统
- 4、xfsdump 备份下来的数据 (档案或储存媒体) 只能让 xfsrestore 解析
- 5、xfsdump 是透过文件系统的 UUID 来分辨各个备份档的, 因此不能备份两个具有相同 UUID 的文件系统

3.4.4 增量备份

概念

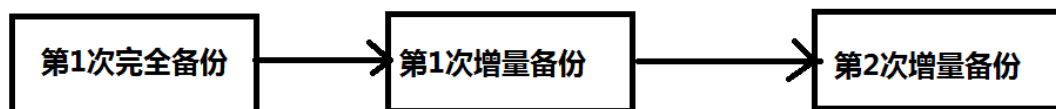
增量备份是指在**一次全备份**或上一次增量备份后, **以后每次的备份只需备份与前一次相比增加或者被修改的文件**。这就意味着, 第一次增量备份的对象是进行全备后所产生的增加和修改的文件; 第二次增量备份的对象是进行第一次增量备份后所产生的增加和修改的文件, 以此类推。

优缺点

优点: 没有重复的备份数据, 因此备份的数据量不大, 备份所需的时间很短。

缺点: 增量备份的数据恢复比较麻烦, 您必须具有上一次全量备份和所有增量备份数据 (一旦丢失或损坏, 就会导致恢复失败) 并且恢复过程中, **必须遵循从最近一次全量备份开始, 并以最近一次全量备份为起点, 按备份等级依次进行数据恢复**。如图:

增量备份-恢复数据过程



实战: 增量备份文件系统

准备一个备份目录进行备份

```
[root@xuegod63 test]# tree /test/
```



```
/test/
```

```
|— passwd  
|— xuegod  
   |— a
```

1 directory, 2 files 对上面的内容进行第一次全备

```
[root@xuegod63 test]# xfsdump -f /opt/test-full /test -L test-full -M sdb1
```

#增加一些内容, 然后进行第 1 次增量备份

```
[root@xuegod63 test]# touch /test/1.txt /test/2.txt
```

```
[root@xuegod63 test]# echo 'i love you' > /test/1.txt
```

```
[root@xuegod63 test]# tree
```

```
|— 1.txt  
|— 2.txt  
|— passwd  
|— xuegod  
   |— a
```

1 directory, 4 files

```
[root@xuegod63 test]# xfsdump -l 1 -f /opt/test-bak1 /test -L test-bak1 -M sdb1
```

-l <level> 小写的 L, 不是大写的 i 做一个等级为 1 的备份

再次增加内容, 然后进行 **level 2 级别** 的增量备

```
[root@xuegod63 test]# touch /test/xuegod/a.txt /test/xuegod/b.txt
```

```
[root@xuegod63 test]# echo 'i hate you' > /test/1.txt
```

```
[root@xuegod63 test]# tree /test/
```

```
/test/
```

```
|— 1.txt  
|— 2.txt  
|— passwd  
|— xuegod  
   |— a  
   |— a.txt  
   |— b.txt
```

1 directory, 6 files

```
[root@xuegod63 test]# xfsdump -l 2 -f /opt/test-bak2 /test -L test-bak2 -M sdb1
```

```
[root@xuegod63 test]# xfsrestore -f /opt/test-bak2 -t
```

```
[root@xuegod63 test]# xfsrestore -f /opt/test-bak1 -t
```

```
[root@xuegod63 ~]# rm -rf /test/* #删除所有数据
```

现在进行恢复, 要想恢复全部全部数据, 包括新添加的文件, 如何恢复?

步骤:

1、先恢复完全备份

2、情况 1: 恢复最后一次增量备份 (如果两次增量备份都是 1 级的, 只需要恢复最后一个增量就可以了。

3、情况 2: 如果你做的是第一次是 1 级备, 第二次是 2 级备, 那么你在恢复的时候就需要先恢复完全备份, 然后是 1 级备, 最后是 2 级备)

注: xfs 增量备份恢复时。要想把数据完全恢复, 必须按顺序恢复。

不按顺序恢复是可以恢复一部分数据, 但是不完整, 比如例子中, 1.txt 的值最后是 I hate you, 如果先恢复 bak2, 然后恢复 bak1, 那么 1.txt 中的值将是 I love you

```
[root@xuegod63 ~]# xfsrestore -f /opt/test-full /test/
```

```
[root@xuegod63 ~]# xfsrestore -f /opt/test-bak1 /test/  #先恢复 bak1, 查看
```

```
[root@xuegod63 ~]# tree /test/  #查看恢复的文件
```

```
[root@xuegod63 ~]# xfsrestore -f /opt/test-bak2 /test/  #再恢复 bak2
```

```
[root@xuegod63 ~]# tree /test/
```

到此, 数据恢复成功了。

总结:

- 3.1 Linux 系统目录结构和相对路径与绝对路径
- 3.2 文件的管理
- 3.3 查看文件
- 3.4 实战: xfs 文件系统的备份和恢复