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！

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行业动态，获取更多学习资料及答疑就业服务！

第四章　Vim 编辑器和恢复 ext4 下误删除文件-Xmanager 工具

本节所讲内容：

4.1　vim 的使用
4.2　实战：恢复 ext4 文件系统下误删除的文件
4.3　实战：使用 xfs_undelete 恢复误删除文件
4.4　实战：使用 xmanager 等远程连接工具管理 Linux

4.1　vim 主要模式介绍

vim 命令模式
问：vi 和 vim 是同一个软件包安装的吗?
答：NO，vim 是 vi 的增加版，最明显的区别就是 vim 可以语法加亮，它完全兼容 vi

查看一个命令，是哪个软件包，安装的：
[root@xuegod63 ~]# rpm -qf /usr/bin/vim
[root@xuegod63 ~]# which vim
[root@xuegod63 ~]# rpm -qf `which vim`　#`反引号，esc 按键下的键，反引号中可执行命令
[root@xuegod63 ~]# rpm -qf $(which vi)　#$(可执行命令)
$() 与`` 都可以进行命令替换，命令替换与变量替换差不多，都是用来重组命令行的，先完成引号里的命令行，然后将其结果替换出来，再重组成新的命令行

4.1.1　vim 编辑器的四种操作模式

1.Vim 常用 4 种模式.
正常模式(Normal mode，俗称命令模式) ，命令行模式(Command-line mode)
插入模式(Insert mode ，俗称编辑模式)， 可视模式(Visual mode，俗称可视块模式)

[root@xuegod63 ~]# cp /etc/passwd a.txt
[root@xuegod63 ~]# vim a.txt
首次进入文件　　----正常模式(Normal mode，俗称命令模式)
按下 I 键，出现“Insert”　　----插入模式(Insert mode ，俗称编辑模式)
按 Esc 键，再输入冒号:　　----命令行模式(Command-line mode)

例 1 从编辑模式到命令行模式怎样切换?
编辑模式->esc->命令模式->: ->命令行模式
注意在命令模式下，输入命令无效时，检查下输入法是不是中文输入法，切换为英文输入法
例 2 字符操作(怎样进入编辑模式?)
进入编辑模式 a i o　A I O
说明：
i 当前字符之前插入（光标前）
I 行首插入　（行首）
a 当前字符之后插入（光标后）
A 行尾插入(行尾)

o 下一行插入（另起一行）

O 上一行插入(上一行插入)

x 向后删除一个字符　　等同于 delete

X 向前删除一个字符

u 撤销一步　每按一次就撤销一次

ctrl+r 恢复，每按一次就恢复一次

r 替换

4.1.2　在正常模式下做的操作：

1、光标定位

hjkl　左下上右

0 和 home 键表示切换到行首，　$和 end 键表示切换到行尾

gg 快速定位到文档的首行，　G 定位到末行

3gg 或者 3G　快速定位到第 3 行

/string(字符串)　-----找到或定位你要找的单词或内容然后敲回车，如果相符内容比较多，我们可以通过 N、n 来进行向上向下查找，并且 vim 会对查找到的内容进行高亮显示，取消高亮用 :noh

/^d　----尖括号^意思表示以什么开头，查找以字母 d 开头的内容

/bash$　-----$意思表示以什么结尾，查找以字母 bash 结尾的内容

vim + /etc/passwd　打开文件后，光标会自动位于文件的最后一行。 了解一下这个技巧。

vim +23 /etc/passwd　打开文件后，光标会自动位于文件的第 23 行，方便后期排错。如：服务器启动报错，第 23，有语法错误。使用 vim +23 /etc/passwd 可以快速定位到 23 行。

分享心得：我更喜欢 vim 打开文件，然后按 G，跳到最后。 因这个 vim + a.txt　技巧不常用，过一段时间肯定会忘。Linux 中有太多的小技巧，大家应该记那些常用的。

2、在正常模式对文本进行编辑

删除、复制、粘贴、撤销

yy 复制整行

复制 N 行：　Nyy　，比如：　2yy　，表示复制 2 行

dd（删除，以行为单位，删除当前光标所在行）

删除 N 行：　Ndd　，比如：　2dd　，表示删除 2 行

p ：　P 粘贴

剪切：　dd

x 删除光标所在位置的字符

D 从光标处删除到行尾

u　撤销操作

ctrl+r　还原撤销过的操作，将做过的撤销操作再还原回去，也就是说撤销前是什么样，再还原成什么样

r 替换，或者说用来修改一个字符

总结：vim 如何进入其它模式

a A　o O　i I 都是可以进行插入，编辑模式

：进入命令行模式

ctrl+v 进入可视块模式

R 擦除、改写，进入替换模式

你进入以上模式后，想要退出 ，按 esc

4.1.3 Visual mode 可视块模式

编程或修改服务器配置文件的时候，需要进行多行注释，会使用 Visual 模式。

1、进入 Visual 模式的批量删除，方法如下：

删除：再按 ctrl+v 进入可视块模式；向下或向上移动光标 ；选中部分内容,然后按 d，就会删除注释符号。

例：将 sshd_config 文件中 17 行到 20 行前面的#号删除

[root@xuegod63 ~]# vim /etc/ssh/sshd_config

改：

```
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

为：

```
Port 22
AddressFamily any
ListenAddress 0.0.0.0
ListenAddress ::
```

2、进入 Visual 模式的批量修改，方法如下：

1)、ctrl+v 进入列编辑模式

2)、向下或向上移动光标，把需要注释、编辑的行的开头选中起来

4)、然后按大写的 I

5)、再插入注释符或者你需要插入的符号,比如"#"

6)、再按 Esc,就会全部注释或添加了

例：在 sshd_config 文件中 17 行到 20 行前面加一个#号

[root@xuegod63 ~]# vim /etc/ssh/sshd_config

改：

```
Port 22
AddressFamily any
ListenAddress 0.0.0.0
ListenAddress ::
```

为：

```
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

4.1.4 命令行模式 Command-line 操作技巧

1、命令行模式 Command-line 操作技巧

:w 保存 save

:w! 强制保存

:q 没有进行任何修改，退出 quit

:q! 修改了，不保存，强制退出

:wq 保存并退出

:wq! 强制保存并退出

:x 保存退出

:e! 复原，恢复到文件打开后，没有进行修改时的状态。 修改了很多，不想保存，想复原，按：e!

在正常模式下，按下大写的 ZZ，也可以保存并退出

例: wq! 强制保存并退出

```
[root@xuegod63 ~]# ll /etc/shadow
----------. 1 root root 1179 9 月  19 12:57 /etc/shadow
[root@xuegod63 ~]# vim /etc/shadow
```

例 1：调用外部文件或命令

语法：在命令行模式下输入： !+命令

例：在 vim 编辑文档写要写入 MAC 地址。

```
[root@xuegod63 ~]# cp /etc/passwd a.txt
[root@xuegod63 ~]# vim a.txt
:!ifconfig        #在 vim 中调用 ifconfig 命令
```

读取其他文件。(把其他文件中的内容追加到当前文档中)

```
:r /etc/hosts
```

2、文本替换

格式 ：范围（其中%所有内容） s 分隔符 旧的内容 分隔符 新的内容 （分隔符可以自定义）

默认是每一行的第一个符合要求的词 (/g 全部)

```
[root@xuegod63 ~]# vim a.txt
:1,3 s/bin/xuegod        #替换第 1 到 3 行中出现的第一个 bin 进行替换为 xuegod
:1,3 s/bin/xuegod/g      #替换第 1 到 3 行中查找到所有的 bin 进行替换为 xuegod
:3 s/xue/aaaaa/g           #只把第 3 行中所有 xue 替换为 aaaaa 了
```

修改 a.txt ，先在文件中随意插入几个 do 和 DO 字符

```
:% s/do/xuegod/g        #将文本中所有的 do 替换成 xuegod
:% s/do/xuegod/gi       #将文本中所有的 do 替换成 xuegod，并且忽略 do 的大小写
:% s@xuegod@do@g    #将文本中所有的 xuegod 替换成 do，替换时，也可以使用@做分隔符
```

4.1.5　自定义 vim 使用环境

1、临时设置

[root@xuegod63 ~]# vim a.txt

:set nu　设置行号

:set nonu 取消设置行号

:noh　　取消高亮显示

2、永久设置环境

vim /etc/vimrc　#设置后会影响到系统所有的用户

~/.vimrc　　#在用户的家目录下，创建一个.vimrc。这样只影响到某一个用户，没有自己建一个

例 1：临时定制 vim 开启显示行号功能

[root@xuegod63 ~]# echo "set nu" >　/root/.vimrc

[root@xuegod63 ~]# vim　/etc/passwd　#发现默认已经有行号了

```
  1 root:x:0:0:root:/root:/bin/bash
  2 bin:x:1:1:bin:/bin:/sbin/nologin
  3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
  4 adm:x:3:4:adm:/var/adm:/sbin/nologin
  5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
  6 sync:x:5:0:sync:/sbin:/bin/sync
```

3、vim 打开多个文件

方法 1：以上下形势，打开两个文档

[root@xuegod63 ~]# vim -o /etc/passwd /etc/hosts

```
 13 nobody:x:65534:65534:Kerne
/etc/passwd
  1 127.0.0.1   localhost loca
  2 ::1         localhost loca
~
~
~
~
~
~
/etc/hosts
:qa
```

注：输入　：qa 一次退出所有打开的文件

方法 2：以左右方式打开两个文档

[root@xuegod63 ~]# vim -O /etc/passwd /etc/hosts

注：ctrl+ww　在两文档之间进行切换编辑。大写 O 左右分屏，小写的 o 上下分屏

比较两个文件内容

[root@xuegod63 ~]# cp /etc/passwd mima.txt

[root@xuegod63 ~]# echo aaa >> mima.txt

方法 1：

[root@xuegod63 ~]# diff /etc/passwd mima.txt

40a41

> aaa

方法 2：

[root@xuegod63 ~]# vimdiff /etc/passwd mima.txt

4.1.6　其它编辑器

nano 编辑器

emacs 编辑器

GHOME 编辑器 gedit

例：

[root@xuegod63 ~]# gedit /etc/passwd

4.1.7　实战 1：解决上传 windows 中文文档乱码

实验环境：centos8 现在系统默认使用的语言是汉语。(系统中必须安装好中文包)。

将同目录下“aaa 此文件在 windows 下打开正常-到 linux 下 vim 打开是乱码.txt”上传到 Linux 服务器上。使用 ssh 远程连接到 Linux 上，使用 vim 打开显示乱码。

原因：编码的问题

通过 iconv 命令转码

参数：

-f, --from-code=名称 原始文本编码

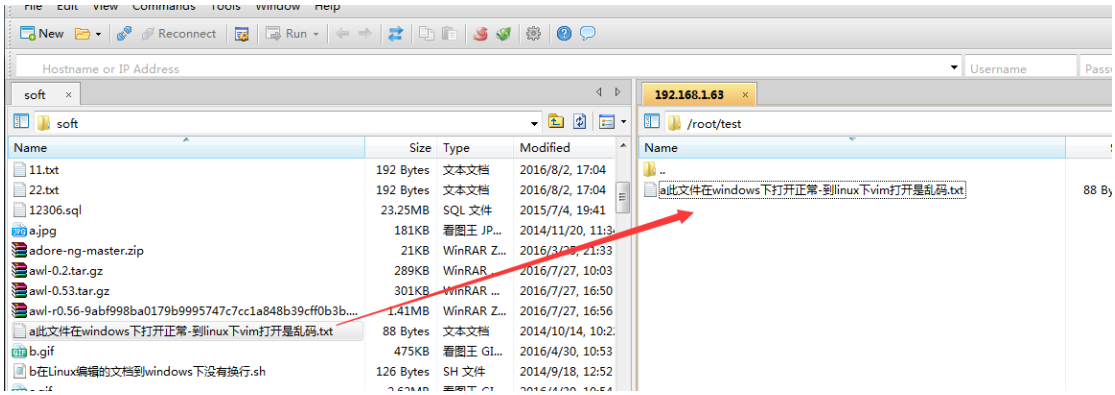
-t, --to-code=输出编码

-o, --output=FILE 输出文件名

[root@xuegod63 ~]# mkdir test　#创建一个测试目录

[root@xuegod63 ~]# cd test/

将测试的文件上传到 Linux 服务器上：



[root@xuegod63 ~]# iconv -f gb2312　-t utf8 aaa 此文件在 windows 下打开正常-到 linux 下 vim 打开是乱码.txt　-o abc.txt

```
[root@xuegod63 ~]# cat abc.txt
#!/bin/bash
echo "学神 IT"
```

4.2 实战：在 Centos6/RHEL6 上恢复 ext4 文件系统下误删除的文件


```
[root@xuegod63 ~]# rm -rf /      #这个可以执行成功吗?      执行不成功的，
rm: 在"/" 进行递归操作十分危险
rm: 使用 --no-preserve-root 选项跳过安全模式
[root@xuegod63 ~]# rm -rf /*      #这个可以执行成功。
```

ext4 文件系统上删除文件，可以恢复: extundelete ， ext3 恢复使用：ext3grep
windows 恢复误删除的文件： final data v2.0 汉化版 和 easyrecovery
xfs 文件系统上删除文件，暂时没有太好的办法进行完全恢复，需要找专业数据恢复公司

扩展:
Linux 文件系统由三部分组成：文件名，inode，block
windows 也由这三部分组成。
a.txt -->inode --> block
文件名 存放文件元数据信息 真正存放数据
查看文件文件名:

```
[root@xuegod63 ~]# cp /etc/passwd a.txt
[root@xuegod63 ~]# ls a.txt
a.txt
```

查看 inode 号:
常识： 每个文件，有一个 inode 号。

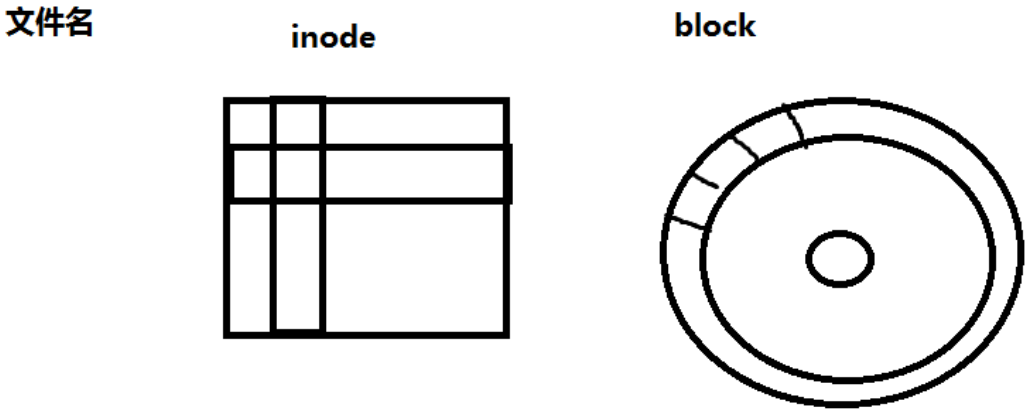
```
[root@xuegod63 ~]# ls -i a.txt
440266 a.txt
```

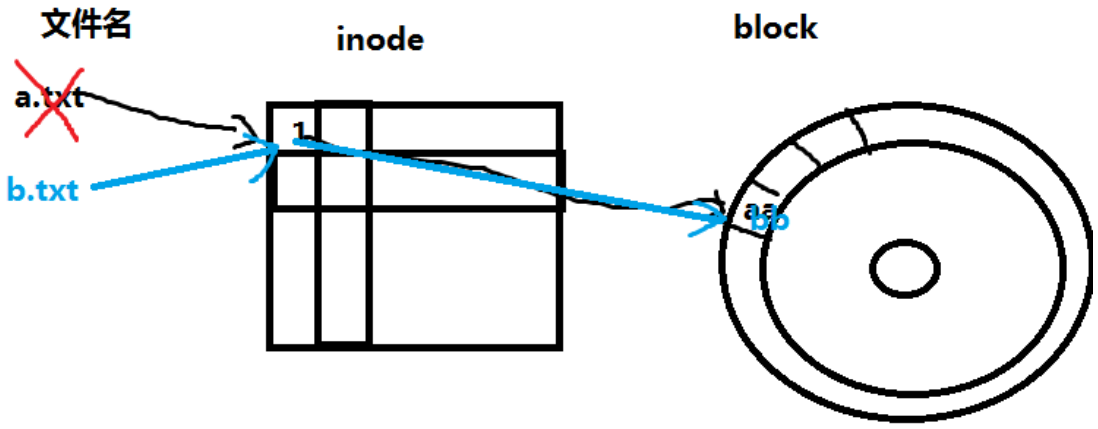
查看 inode 中的文件属性； 通过 stat 命令查看 inode 中包含的内容

```
[root@xuegod63 ~]# stat a.txt     #查看 inode 信息:
```

[root@xuegod63 ~]# ls -l a.txt
-rw-r--r-- 1 root root 1720 Oct 25 10:21 a.txt

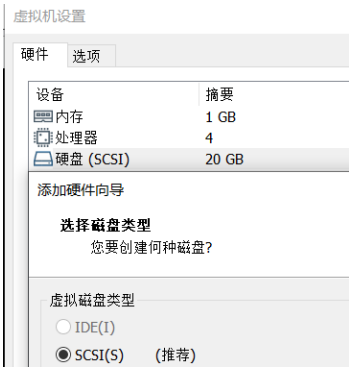
block 块：真正存储数据的地方
逻辑删除：
为什么删除比复制快?





误删除文件后，第一件事要做什么？？？　你不心删除把存了几十年的大片删除了。
避免误删除的文件内容被覆盖。 如何避免?
卸载需要恢复文件的分区或以只读的方式挂载

4.2.2　实战：在 ext4 文件系统上恢复被误删除的文件
下载 extundelete
http://sourceforge.net/　　开源软件发布中心
准备测试分区：
先添加一块硬盘




```
[root@xuegod63 /]# fdisk /dev/sdb   #创建一个 sdb1 分区

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p     #查看现有分区表

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000b8b35

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1301    10240000   83  Linux
/dev/sda3            1301        1428     1024000   82  Linux swap / Solaris

Command (m for help): n     #创建一个新分区
Command action
   e   extended
   p   primary partition (1-4)
p   #创建一个主分区
Selected partition 4
First cylinder (1428-2610, default 1428):
Using default value 1428
Last cylinder, +cylinders or +size{K,M,G} (1428-2610, default 2610): +1G   #指定分区大小

Command (m for help): w   #保存
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@xuegod63 ~]#reboot
或
[root@xuegod63 ~]# partx -a /dev/sdb #获得新分区表
扩展：
如果在根下删除文件了，想恢复，怎么办?
方法 1： 立即断电，然后把磁盘以只读方式，挂载到另一个电脑中进行恢复
方法 2：把 extundelete 在虚拟机上（虚拟机系统要和服务器版本一样），提前安装好后再复制到 U 盘中，把 U 盘插入服务器，恢复时，恢复的文件要保存到 U 盘中，（不要让恢复的数据写到/下，那样会覆盖之前删除的文件）

使用新的分区表：
[root@xuegod63 /]# mkdir /tmp/sdb1 #创建挂载点
[root@xuegod63 ~]# mkfs.ext4 /dev/sdb1 #格式化
[root@xuegod63 ~]# mount /dev/sdb1 /tmp/sdb1 #挂载

4.2.3 准备测试环境

复制一些测试文件，然后把这些文件再删除，然后演示恢复：
[root@xuegod63 ~]# cp /etc/passwd /tmp/sdb1
[root@xuegod63 ~]# cp /etc/hosts /tmp/sdb1
[root@xuegod63 ~]# echo aaa > a.txt
[root@xuegod63 ~]# mkdir -p /tmp/sdb1/a/b/c
[root@xuegod63 ~]# cp a.txt /tmp/sdb1/a
[root@xuegod63 ~]# cp a.txt /tmp/sdb1/a/b
[root@xuegod63 ~]# touch /tmp/sdb1/a/b/kong.txt
安装 tree 命令：
[root@xuegod63 ~]# rpm -ivh /mnt/Packages/tree-1.5.3-2.el6.x86_64.rpm
[root@xuegod63 ~]# tree /tmp/sdb1
/tmp/sdb1/
├── a
│ ├── a.txt
│ └── b
│ ├── a.txt
│ ├── c #空目录
│ └── kong.txt #空文件
├── hosts
├── lost+found
└── passwd

lost+found

使用标准的 ext3/ext4 档案系统格式才会产生的一个目录，目的在于当档案系统发生错误时， 将一些遗失的片段放置到这个目录下。

可以删除 rm -rf lost+found

可以创建 mklost+found

删除文件：

[root@xuegod63 ~]# cd /tmp/sdb1/

[root@xuegod63 sdb1]# ls

a hosts lost+found passwd

[root@xuegod63 sdb1]# rm -rf a hosts passwd

误删除文件后，第一件事要做什么？？？

如何避免误删除的文件内容被覆盖？？？

卸载需要恢复文件的分区：或以只读的方式挂载

[root@localhost ~]#cd /root

[root@localhost ~]# umount /tmp/sdb1

4.2.4 安装 extundelet

上传 extundelete 到 linux 中：

从 windows 上传 extundelete 文件到 linux，安装 xmanager v5 或者 C R T

[root@xuegod63 ~]# rpm -ivh /mnt/Packages/lrzsz-0.12.20-27.1.el6.x86_64.rpm

安装后，就有了 r z 命令和 s z 命令

rz ： 上传 windows 中的文件到 linux

sz 文件名 ： 下载，将 linux 中的文件传到 windows

解压并安装 extundelet

[root@centos63~]#mount /dev/sr0 /mnt

[root@centos63~]# vim /etc/yum.repos.d/Centos-6.repo

```
[CentOS6]
name=CentOS-server
baseurl=file:///mnt
enabled=1
gpgcheck=0
```

[root@xuegod63]# yum -y install e2fsprogs-devel gcc gcc-c++ gcc-g77

[root@xuegod63 extundelete-0.2.4]# tar xf extundelete-0.2.4.tar.bz2

[root@xuegod63 ~]# cd extundelete-0.2.4

[root@xuegod63 extundelete-0.2.4]# ./configure #检查系统安装环境

[root@xuegod63 extundelete-0.2.4]# make -j 4 #编译，把源代码编译成可执行的二进制文件。

-j 4 使用 4 进程同时编译，提升编译速度 或 使用 4 核 CPU 同时编译。

[root@xuegod63 extundelete-0.2.4]# make install #安装

install 和 cp 有什么区别?

install 复制时可以指定权限 cp 不可以

例:

[root@xuegod63 ~]# install -m 777 /bin/find /opt/find

[root@xuegod63 ~]# ll /opt/

4.2.5 恢复数据:

方法 1：通过 inode 结点恢复

方法二：通过文件名恢复

方法三：恢复某个目录，如目录 a 下的所有文件:

方法四：恢复所有的文件

[root@xuegod63 ~]# umount /tmp/sdb1/

[root@xuegod63 ~]# mkdir test #创建一个目录使用于存放恢复的数据

[root@xuegod63 ~]# cd test/

方法 1:

通过 inode 结点查看被删除的文件名字:

[root@xuegod63 test]# extundelete /dev/sdb1 --inode 2

. 2

lost+found 11

passwd 12 Deleted

hosts 13 Deleted

a 7313 Deleted

扩展：ext4 文件系统的分区根目录的 inode 值为 2，xfs 分区根目录的 inode 值为 64

[root@xuegod63 test]# ls -id / #xfs 文件系统

64 /

[root@xuegod63 test]# mount /dev/sdb1 /tmp/sdb1/

[root@xuegod63 test]# ls -id /tmp/sdb1/

2 /tmp/sdb1/

[root@xuegod63 test]# umount /tmp/sdb1/

方法 1：通过 inode 结点恢复

[root@xuegod63 test]# extundelete /dev/sdb1 --restore-inode 12

NOTICE: Extended attributes are not restored.

Loading filesystem metadata ... 9 groups loaded.

Loading journal descriptors ... 63 descriptors loaded.

[root@xuegod63 test]# ls

RECOVERED_FILES

[root@xuegod63 test]# diff /etc/passwd RECOVERED_FILES/file.12

#没有任何输出，说明一样

方法二，通过文件名恢复

[root@xuegod63 test]# extundelete /dev/sdb1 --restore-file hosts

[root@xuegod63 test]# diff /etc/passwd RECOVERED_FILES/hosts

#没有任何输出，说明一样

方法三：恢复某个目录，如目录 a 下的所有文件：

```
[root@xuegod63 test]# extundelete /dev/sdb1 --restore-directory a
[root@xuegod63 test]# tree RECOVERED_FILES/a/
RECOVERED_FILES/a/
├── a.txt
└── b
```

下面是原来的目录结构：

```
[root@xuegod63 ~]# tree /root/sdb1-back/a/
/root/sdb1-back/a/
└── b
    ├── a.txt
    ├── c
    └── kong.txt
```

方法四：恢复所有的文件

```
[root@centos6 test]# rm -rf RECOVERED_FILES/
[root@xuegod63 test]# extundelete /dev/sdb1 --restore-all
[root@centos6 test]# tree RECOVERED_FILES/
```

删除前后的数据：

删除前

```
├── a
│   ├── a.txt
│   └── b
│       ├── a.txt
│       ├── c  #空目录
│       └── kong.txt  #空文件
├── hosts
├── lost+found
└── passwd
```

恢复后的数据：

```
├── a
│   ├── a.txt
│   └── b
│       └── a.txt
├── hosts
└── passwd
```

extundelete 在恢复文件的时候能不能自动创建空文件和目录?

答：不能。

4.3 使用 xfs_undelete 恢复误删除文件

XFS 文件系统的取消删除工具——xfs_undelete。

xfs_undelete 尝试恢复 xfs 文件系统中标记为已删除的所有文件。

恢复的文件存储在子目录中的另一个文件系统中，默认情况下，相当于当前目录 xfs_ undelete。文件名无法恢复，它被作为删除时间、inode 编号和猜测的文件扩展名。

环境需求

先添加一块硬盘，后面会用到

xfs_undelete 是一个小的 Tcl 脚本，因此需要一个 Tcl 解释器。它使用了 Tcl-8.6 的一些特性，所以至少需要这个版本，tclib 包用于解析命令行。

下载 tcl8.6，tcllib 和 xfs_undelete

https://sourceforge.net/projects/tcl/

https://core.tcl-lang.org/tcllib/technote/cd3a11c3065120d491009e64a19f7676176045cd

https://github.com/ianka/xfs_undelete

上传软件包到 centos7 系统

安装 tcl

```
[root@xuegod63 ~]# tar xvf tcl8.6.11-src.tar.gz
[root@xuegod63 ~]# cd tcl8.6.11/unix/
[root@xuegod63 ~]#./configure
[root@xuegod63 ~]# echo $?
[root@xuegod63 ~]# make -j 4 && make install
[root@xuegod63 ~]# echo $?
```

时间很长

使 tclsh 全局生效加入 path 变量

```
[root@xuegod63 ~]# mv /root/tcl8.6.11 /root/tcl
[root@xuegod63 ~]# vim /etc/profile   # 在文件最后追加以下内容，永久生效
export PATH=/root/tcl/unix/:$PATH
[root@xuegod63 ~]# source   /etc/profile    #重新加载配置文件，使用配置生效
[root@xuegod63 ~]# echo $PATH
```

安装 tcllib

```
[root@xuegod63 ~]# tar xf tcllib-1.20.tar.gz
[root@xuegod63 ~]# cd tcllib-1.20/
[root@xuegod63 tcllib-1.20]# ./configure
[root@xuegod63 tcllib-1.20]# echo $?
[root@xuegod63 tcllib-1.20]# make -j 4 && make install
[root@xuegod63 tcllib-1.20]# echo $?
```

安装 xfs_undelete

```
[root@xuegod63 tcllib-1.20]# cd
[root@xuegod63 ~]# unzip xfs_undelete-master.zip
[root@xuegod63 ~]# cd xfs_undelete-master/
[root@xuegod63 xfs_undelete-master]# ./xfs_undelete -h      #查看帮助信息
```

挂载点创建一些测试文件，文件里要有内容，然后删除几个

```
[root@xuegod63 xfs_undelete-master]# cd
[root@xuegod63 ~]# gdisk /dev/sdb
[root@xuegod63 ~]# mkfs.xfs /dev/sdb1
```

[root@xuegod63 ~]# mkdir /testsdb1
[root@xuegod63 ~]# mount /dev/sdb1 /testsdb1
[root@xuegod63 ~]# mkdir /testsdb1/kong
[root@xuegod63 ~]# touch /testsdb1/kong.txt
[root@xuegod63 ~]# cp /etc/passwd /testsdb1/
[root@xuegod63 ~]# cp /etc/passwd /testsdb1/kong
[root@xuegod63 ~]# echo "hello world" > /testsdb1/hello.txt
[root@xuegod63 ~]# ls /testsdb1

```
[root@xuegod63 testsdb1]# ls
hello.txt  kong  kong.txt  passwd
```

[root@xuegod63 ~]# rm -rf /testsdb1/*
[root@xuegod63 testsdb1]# cd
[root@xuegod63 ~]# umount /testsdb1
[root@xuegod63 ~]# cd xfs_undelete-master/
[root@xuegod63 xfs_undelete-master]# ./xfs_undelete /dev/sdb1

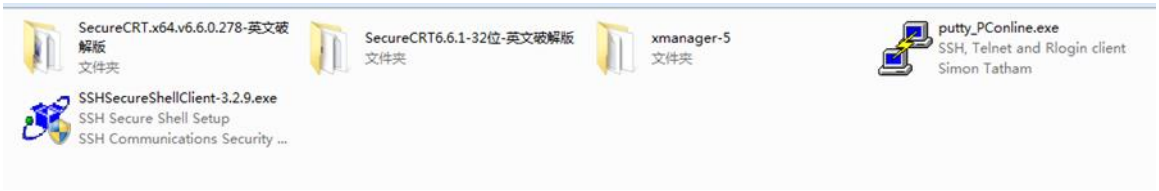
```
[root@xuegod63 xfs_undelete-master]# ./xfs_undelete /dev/sdb1
Starting recovery.
Recovered file -> xfs_undeleted/2021-06-08-11-02_67.txt
Recovered file -> xfs_undeleted/2021-06-08-11-02_68.txt
Recovered file -> xfs_undeleted/2021-06-08-11-02_71.txt
Done.
```

[root@xuegod63 xfs_undelete-master]# cd xfs_undeleted/
[root@xuegod63 xfs_undeleted]# ls
[root@xuegod63 xfs_undeleted]# head 2021-06-08-11-02_68.txt
hello world
直接执行脚本首先会以只读的方式重新挂载，然后恢复，恢复的不是原文件名，但是内容是一样的。
注意需要进入目录才能看到
注意：不会恢复目录和空文件

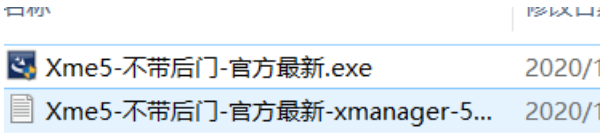
也可以恢复数据到指定的目录
[root@xuegod63 xfs_undeleted]# cd ..
[root@xuegod63 xfs_undelete-master]# ./xfs_undelete -o /opt /dev/sdb1

4.3 实战：使用 xmanager 等远程连接工具管理 Linux

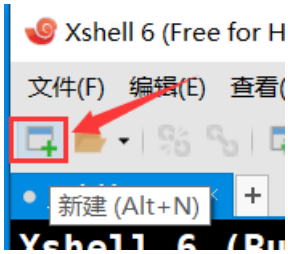
4.3.1 Linux 下常用远程连接工具介绍



4.3.2 xmanager 使用方法

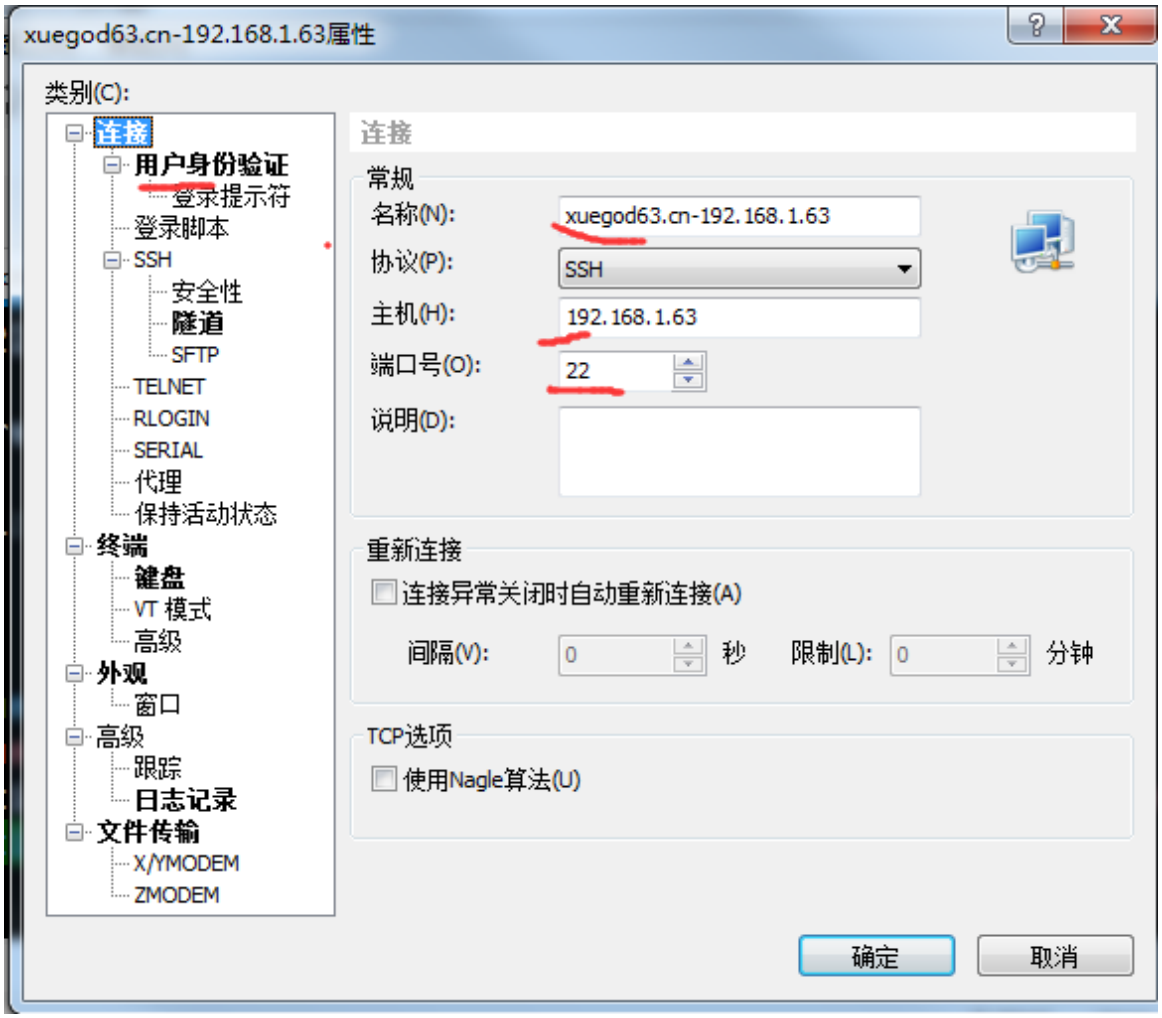


安装以后打开 

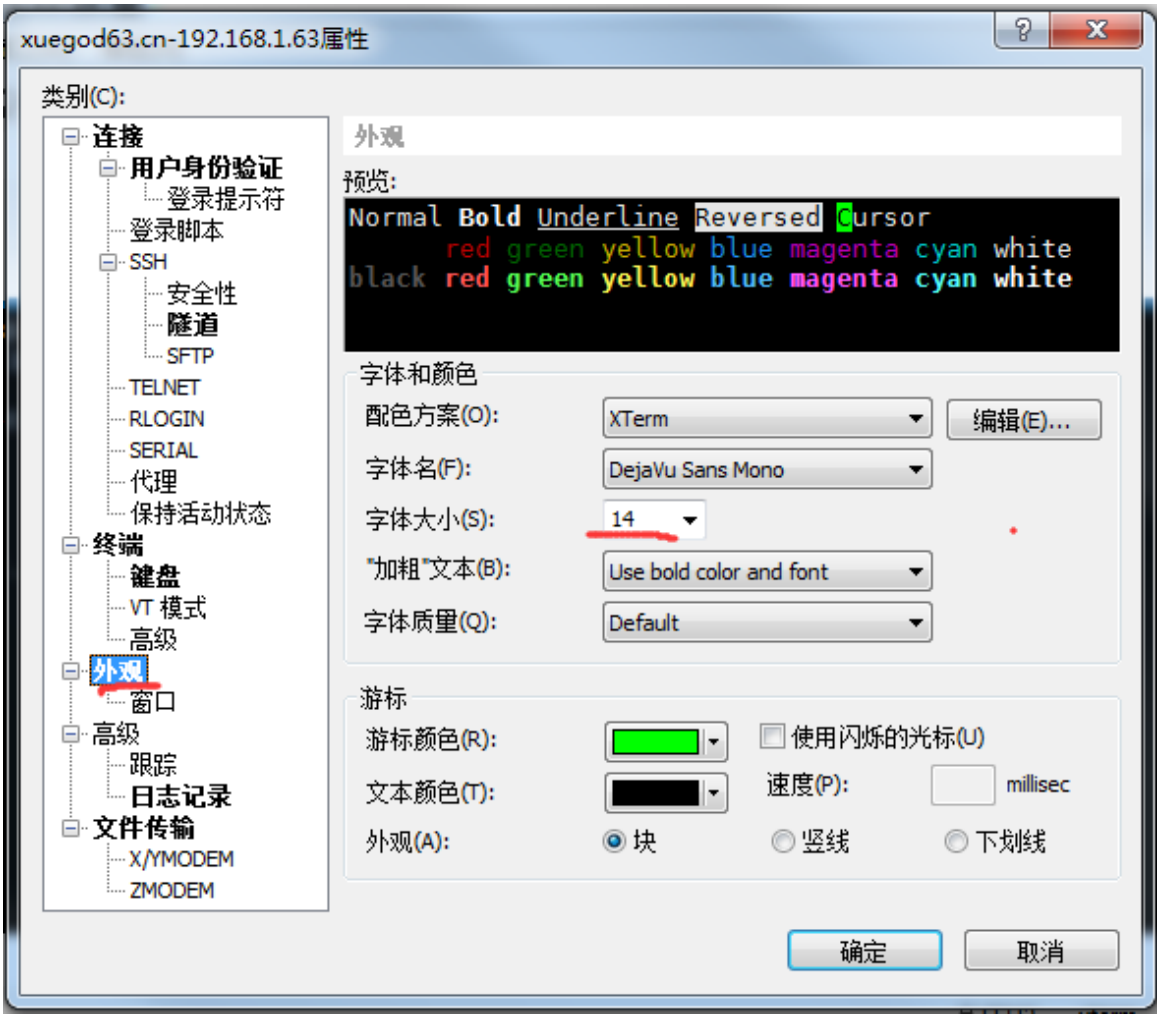


1、xshell 使用方法

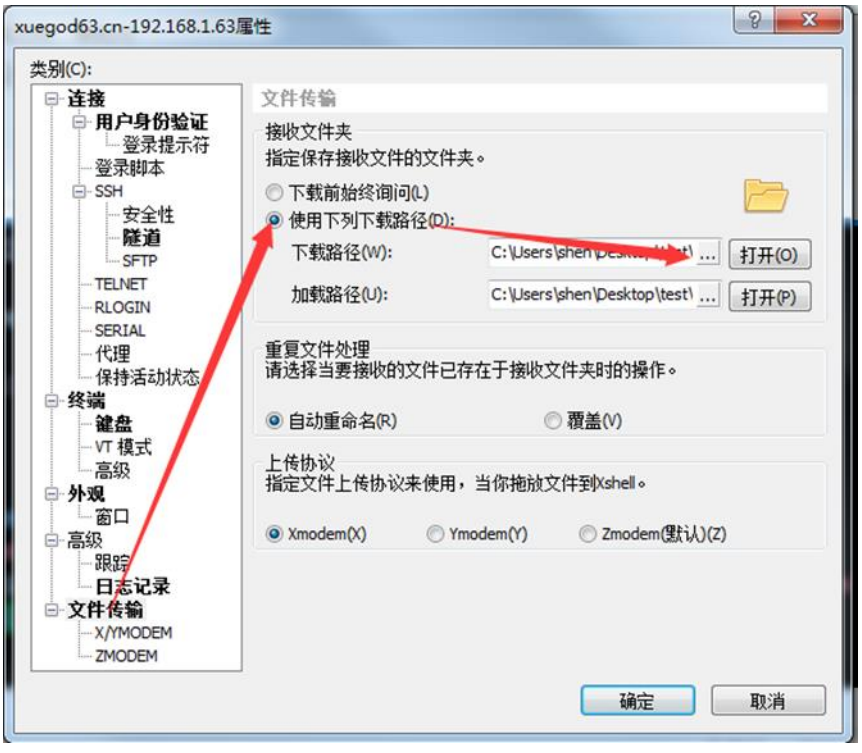
例 1：连接一台新的服务器



例 2：调整 xshell 字体大小



例 3：调整 rz 和 sz 命令的默认路径

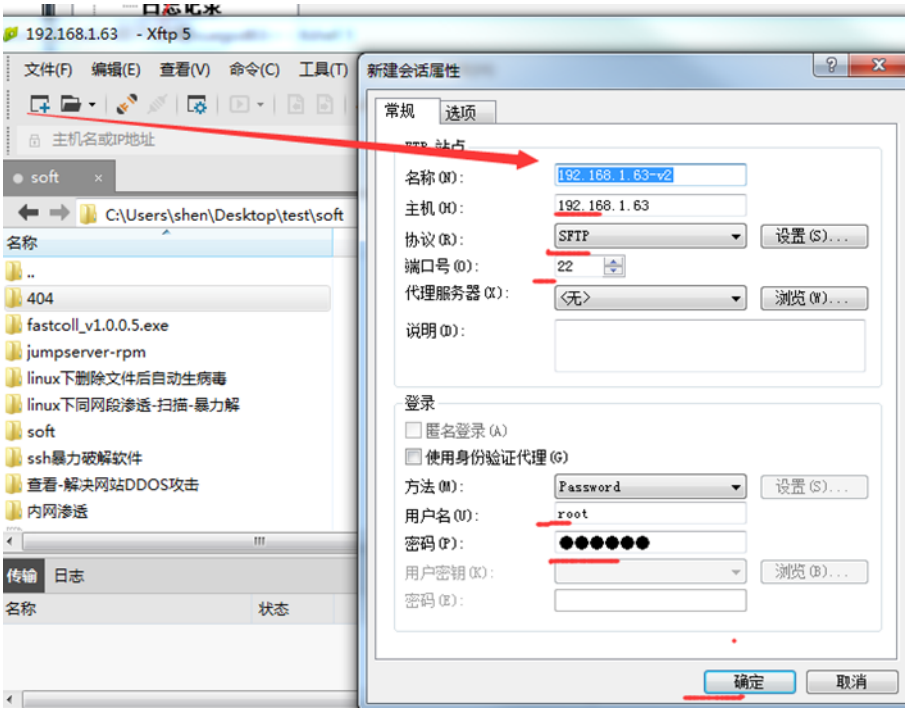


例 4：解决 Xshell 中小键盘无法打出数字的问题

例 5：解决 Xshell 不能使用退格、删除键的问题

2、xftp 使用方法

例 1：上传一个文件夹到 Linux 服务器上



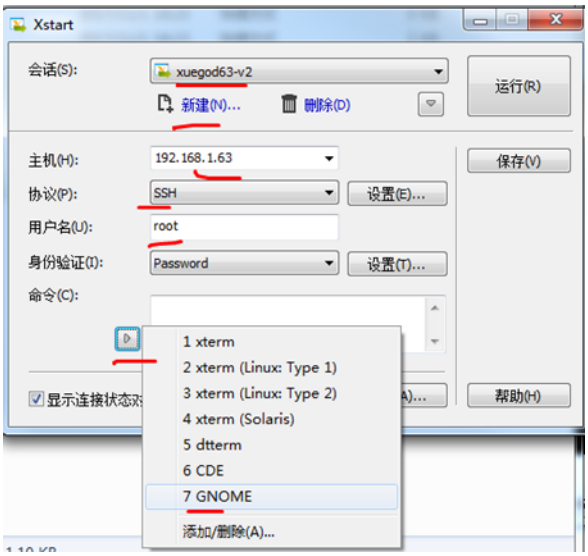
3、xstart 使用方法

方法 1：使用 xshell 直接运行图形界面的程序

例 1：[root@xuegod63 ~]# gnome-terminal

例 2：[root@xuegod63 ~]# firefox &

方法 2：使用 xstart 调用桌面



注：使用 MK 给的 xmanger5 安装后，后期运行，提示更新到新版本， 你不要更新，更新，有可能序列号就不能使用了。

https://www.netsarang.com/zh/free-for-home-school/

也可以下载官方免费版

总结：

- 4.1 vim 的使用
- 4.2 实战：恢复 ext4 文件系统下误删除的文件
- 4.3 实战：使用 xfs_undelete 恢复误删除文件
- 4.4 实战：使用 xmanager 等远程连接工具管理 Linux