**Linux下回收站功能的实现**

本文仿照 Windows 回收站的功能，运用 Bash 脚本在 Linux 上做了实现，创建 delete 脚本代替 rm 命令对文件或目录进行删除操做。该脚本实现了以下功能：对大于 2G 的文件或目录直接删除，否则放入$HOME/trash 目录下；恢复 trash 目录中的被删除文件到原目录下；文件存放在 trash 目录中超过七天被自动删除。

1. 概述

删除是危险系数很高的操作，一旦误删可能会造成难以估计的损失。在 Linux 系统中这种危险尤为明显，一条简单的语句：rm –rf /\* 就会把整个系统全部删除，而 Linux 并不会因为这条语句的不合理而拒绝执行。 在 Windows 中，为了防止误删，系统提供了回收站功能。用户在执行删除操作后，文件并不会直接从硬盘中删除，而是被放到回收站中。在清空回收站前，如果发现有文件被误删，用户可以将回收站中的文件恢复到原来的位置。而 Linux 并没有提供类似功能，删除命令 rm 一旦确认执行，文件就会直接从系统中删除，很难恢复。

1. 回收站构成

本文共用三个脚本实现了回收站的主要功能：Delete 脚本、logTrashDir 脚本和 restoreTrash 脚本。其中 Delete 脚本是核心脚本，其作用是重新封装 rm 命令。相对于 rm 的直接删除，该命令会先将文件或目录移动到$home/trash 目录下。如果用户想要将文件直接删除，可以用 -f 选项，delete 脚本会直接调用 rm –f 命令将文件从硬盘上删除。logTrashDir 脚本用于将被删除文件的信息记录到 trash 目录下的一个隐藏文件中。restoreTrash 脚本用来将放入 trash 中的文件或目录重新恢复到原路径下。在 Linux 系统中，只要将这三个脚本放到/bin/目录下，并用 chmod +X filename 赋予可执行权限，即可直接使用。下面将介绍每个脚本的主要部分。

Delete脚本

1. 创建目录

首先要创建目录来存放被删除的文件，本文在用户根目录$HOME 下建立 trash 目录来存放文件。先判断目录是否已建立，如未建立，即第一次运行该脚本，则创建 trash 目录。变量 realrm 存放了 Linux 的 rm 脚本位置，用于在特定条件下调用以直接删除文件或目录。代码如下，

realrm="/bin/rm"

if [ ! -d ~/trash ]

then

mkdir -v ~/trash

chmod 777 ~/trash

fi

1. 输出帮助信息

该脚本在用户仅输入脚本名而未输入参数执行时，输出简要帮助信息，该脚本的运用格式是 delete 后跟要删除的文件或目录的路径，中间用空格隔开。

if [ $# -eq 0 ]

then

echo "Usage: delete file1 [file2 file3...]"

echo "If the options contain -f,then the script will exec 'rm' directly"

fi

1. 直接删除文件

有些用户确认失效并想直接删除的文件，不应放入回收站中，而应直接从硬盘中删除。如果用户在命令中加入了-f 选项，则 delete 脚本会直接调用 rm 命令将文件或目录直接删除。如代码中所示，所有的参数包括选项都会传递给 rm 命令。所以只要选项中包括选项-f 就等于调用 rm 命令，可以使用 rm 的所有功能。如：delete –rfv filename 等于 rm –rfv filename。Delete 脚本提供了-f 选项来执行这项操作：

while getopts "dfiPRrvW" opt

do

case $opt in

f)

exec $realrm "$@"

;;

\*)

# do nothing

;;

esac

done

1. 用户交互

需要与用户确认是否将文件放入回收站。相当于 Windows 的弹窗提示，防止用户误操作。

1. 判断文件类型并直接删除大于2G的文件

本脚本只对普通文件和目录做操作，其他类型文件不做处理。先对每个参数做循环，判断他们的类型，对于符合的类型再判断他们的大小是否超过 2G，如果是则直接从系统中删除，避免回收站占用太大的硬盘空间。该脚本用不同的命令分别判断目录和文件的大小。鉴于目录的大小应该是包含其中的文件以及子目录的总大小，所以运用了'du -sb'命令。两种情况都使用了 awk 来获取特定输出字段的值来作比较。4、5的代码如下

echo -ne "Are you sure you want to move the files to the trash?[Y/N]:\a"

read reply

if [ $reply = "y" -o $reply = "Y" ]

then

#####

for file in $@

do

if [ -f "$file" -o -d "$file" ]

then

if [ -f "$file" ] && [ `ls -l $file|awk '{print $5}'` -gt 2147483648 ]

then

echo "$file size is larger than 2G,will be deleted directly"

`rm -rf $file`

elif [ -d "$file" ] && [ `du -sb $file|awk '{print $1}'` -gt 2147483648 ]

then

echo "The directory:$file is larger than 2G, will be deleted directly"

`rm -rf $file`

fi

fi

done

fi

1. 移动文件到回收站并做记录

该部分是 Delete 脚本的主要部分，主要完成以下几个功能

1. 获取参数的文件名。因为用户指定的参数中可能包含路径，所以要从中获取到文件名，用来生成 mv 操作的参数。该脚本中运用了字符串正则表达式'${file##\*/}'来获取。
2. 生成新文件名。在原文件名中加上日期时间后缀以生成新的文件名，这样用户在浏览回收站时，对于每个文件的删除日期即可一目了然。
3. 生成被删文件的绝对路径。为了以后可能对被删文件进行的恢复操作，要从相对路径生成绝对路径并记录。用户输入的参数可能有三种情况：只包含文件名的相对路径，包含点号的相对路径以及绝对路径，脚本中用字符串处理对三种情况进行判断，并进行相应的处理。
4. 调用 logTrashDir 脚本，将回收站中的新文件名、原文件名、删除时间、原文件绝对路径记录到隐藏文件中
5. 将文件通过 mv 命令移动到 Trash 目录下。

now=`date +%Y%m%d\_%H\_%M\_%S`

filename="${file##\*/}"

newfilename="${file##\*/}\_${now}"

mark1="."

mark2="/"

if [ "$file" = ${file/$mark2} ]

then

fullpath="$(pwd)/$file"

elif [ "$file" != ${file/$mark1} ]

then

fullpath="$(pwd)${file/$mark1}"

else

fullpath="$file"

fi

echo "The full path of this file is : ${fullpath}"

if mv -f $file ~/trash/$newfilename

then

$(/bin/logTrashDir "$newfilename $filename $now $fullpath")

echo "files: $file is deleted"

else

echo "the operation is failed"

fi

logTrashDir脚本

该脚本较简单，仅是一个简单的文件写入操作，之所以单独作为一个脚本，是为了以后扩展的方便，该脚本先建立.log 隐藏文件，然后往里添加删除文件的记录。

#!/bin/bash

if [ ! -f ~/trash/.log ]

then

touch ~/trash/.log

chmod 700 ~/trash/.log

fi

echo $1 $2 $3 $4>> ~/trash/.log

restoreTrash脚本

该脚本主要完成以下功能：

1. 从.log 文件中找到用户想要恢复的文件对应的记录。此处依然使用 awk，通过正表达式匹配找到包含被删除文件名的一行
2. 从记录中找到记录原文件名的字段，以给用户提示
3. 将回收站中的文件移动到原来的位置，在这里运用了 mv –b 移动文件，之所以加入-b 选项是为了防止原位置有同名文件的情况。
4. 将.log 文件中与被恢复文件相对应的记录删除

#!/bin/bash

originalPath=$(awk /$filename/'{print $4}' "$HOME/trash/.log")

filenameNow=$(awk /$filename/'{print $1}' "$HOME/trash/.log")

filenamebefore=$(awk /$filename/'{print $2}' "$HOME/trash/.log")

echo "you are about to restore $filenameNow,original name is $filenambefore"

echo "original path is $originalPath"

echo "Are you sure to do that?[Y/N]"

read reply

if [ $reply = "y" ] || [ $reply = "Y" ]

then

$(mv -b "$HOME/trash/$filenameNow" $originalPath)

$(sed -i /$filenameNow/'d' "$HOME/trash/.log")

else

echo "no files restored"

fi

自动定期清理 trash 目录

因为 delete 操作并不是真正删除文件，而是移动操作，经过一段时间的积累，trash 目录可能会占用大量的硬盘空间，造成资源浪费，所以定期自动清理 trash 目录下的文件是必须得。本文的清理规则是：在回收站中存在 7 天以上的文件及目录将会被自动从硬盘中删除。运用的工具是 Linux 自带的 crontab。

Crontab 是 Linux 用来定期执行程序的命令。当安装完成操作系统之后，默认便会启动此任务调度命令。Crontab 命令会定期检查是否有要执行的工作，如果有要执行的工作便会自动执行该工作。而 Linux 任务调度的工作主要分为以下两类：

1. 系统执行的工作：系统周期性所要执行的工作，如备份系统数据、清理缓存
2. 个人执行的工作：某个用户定期要做的工作，例如每隔 10 分钟检查邮件服务器是否有新信，这些工作可由每个用户自行设置。首先编写 crontab 执行时要调用的脚本 cleanTrashCan.如清单 10 所示，该脚本主要完成两项功能：a.判断回收站中的文件存放时间是否已超过 7 天，如果超过则从回收站中删除。b.将删除文件在.log 文件中相应的记录删除，保持其中数据的有效性，提高查找效率。

#!/bin/bash

arrayA=($(find ~/trash/\* -mtime +7 | awk '{print $1}'))

for file in ${arrayA[@]}

do

$(rm -rf "${file}")

filename="${file##\*/}"

echo $filename

$(sed -i /$filename/'d' "$HOME/trash/.log")

done

脚本编写完成后通过 chmod 命令赋予其执行权限，然后运过 crontab –e 命令添加一条新的任务调度：

10 18 \* \* \* /bin/ cleanTrashCan

该语句的含义为，在每天的下午 6 点 10 分执行 cleanTrashCan 脚本，通过这条任务调度，trash 的大小会得到有效的控制，不会持续增大以致影响用户的正常操作。

1. 实际应用

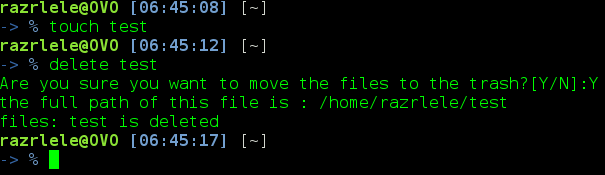
首先要将 delete 脚本，logTrashDir 脚本，restoreTrash 脚本和 cleanTrashCan 放到/bin 目录下，然后用 chmod +x delete restoreTrash logTrashDir cleanTrashCan 命令赋予这三个脚本可执行权限。

运用 delete 脚本删除文件，例如要删除在/usr 目录下的 useless 文件。根据用户目前所在的位置，可以用相对路径或绝对路径来指定参数，如：delete useless，delete ./useless 或者 delete /usr/useless。

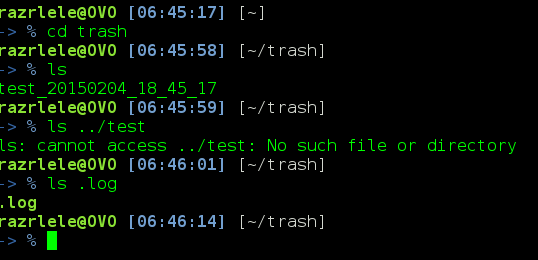
赋予脚本权限如下，

chmod +x delete restoreTrash logTrashDir cleanTrashCan

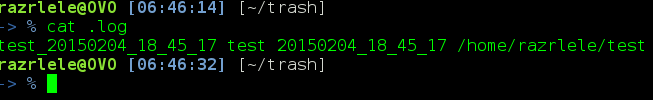
delete脚本执行过程如下图，



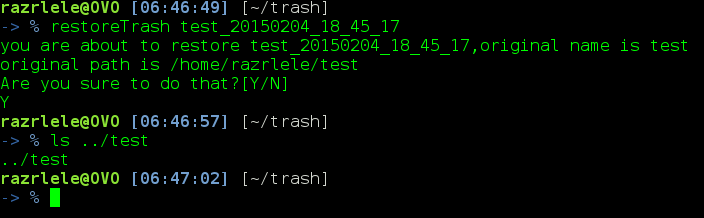
执行之后，useless 文件会从原目录中删除，被移动到$HOME/trash 下，并被重命名，如下图，



生成.log记录，



如果用户在七天之内发现该文件还有使用价值，则可以使用 restoreTrash 命令将被删除文件恢复到原路径下：restoreTrash ~/trash/useless\_20140923\_06\_28\_57。restoreTrash脚本执行情况如下图，



查看/usr 目录，可以发现 useless 文件已经被恢复至此。

useless 文件被恢复

最后再用systemd/Timers定期每星期日下午六点执行cleanTrashCan脚本，首先编辑

/etc/system/system/cleanTrash.service

内容如下，

[Unit]

Description=Clean Trash Can

[Service]

Type=simple

ExecStart=/bin/cleanTrashCan

再编辑，

/etc/system/system/cleanTrash.timer

内容如下，

[Unit]

Description=Runs cleanTrashCan every week

[Timer]

OnCalendar=Sun, 18:00

Unit=cleanTrash.service

[Install]

WantedBy=multi-user.target

执行，

systemctl start cleanTrash.timer

systemctl enable cleanTrash.timer

1. 参考文献

[1] 李朝阳. 回收站功能在 Linux 中的实现[EB/OL]. <https://www.ibm.com/developerworks/cn/linux/1410_licy_linuxtrash/#icomments,2014-10-30>

[2] razrlele. Shell Script实现回收站功能[EB/OL]. <https://blog.razrlele.com/p/776,2015-02-04>

[附件文件]

