# MYSQL杂记

DELIMITER $$

意为从此语句开始，直到”$$”才是语句结束。本质猜测为转义。一般用于定义存储过程。

在存储过程定义完成后输入”$$”，并输入DELIMITER ;将语句结束标识换回”;”。

查询建表语句

SHOW CREATE table person;

@a=2 or @a=:2

一般定义变量或变量赋值时两种均可。

但若定义或赋值在select语句中时，一定要选择第二种。

# Git杂记

命令（git bash下可用linux shell相关命令）

Cmd 部分

创建目录 mkdir <dirname>

创建文件 type nul><filename>

删除目录 rd <dirname>

删除文件 del <filename>

Shell部分

创建目录 mkdir <dirname>

创建文件 touch <filename>

创建并修改文件 vi <filename>

删除文件 rm <filename>

Git部分

初始化本地库 git init

添加文件git add <file>

提交添加的文件至本地库git commit –m “<comments>”

检查文件状态 git status

检查文件与之前版本的不同 git diff

操作日志，会有版本号信息 git log

版本穿梭 git reset --hard <commit\_id>

查看命令历史git reflog

丢弃工作区的修改git checkout -- <file>

删除文件后对版本库中该文件同步删除 git rm→git commit：

拉代码 git clone <url>

创建并切换分支 git checkout –b <branchname>

创建分支 git branch <branchname>

切换分支 git checkout <branchname>

查看分支信息,会显示所有分支，当前分支名前有标记 git branch

合并指定分支到当前分支 git merge <branchname>

删除分支 git branch –d <branchname>

抓取远程feature1分支至本地feature2分支 git fetch origin feature1:feature2

查看所有分支（本地和远程） git branch –a

查看分支合并图 git log –graph

关联远程库git remote add <远程库名 默认origin> <url>

创建本地分支并关联远程分支 git checkout -b ca origin/ca

添加全部文件 git add .

撤销工作区文件 git checkout .

撤销工作区文件 git clean –fd

撤销暂存区指定文件git reset HEAD \*.log

代码提交git push

创建远程分支 git push origin <branchName>

免密码pull或push(不同远程库帐号密码会区分记录)

git config –global credential.helper “store”

删除远程分支 git push origin –-delete <branchName> 或

git push origin :<remote branchName>

设置本地分支与远程分支关联(疑似指令成功后还得push一次才能正式关联)

git branch --set-upstream-to=<remote branchName> <local branchName>

完整拉取分支文件流程。

1. mkdir <dirName>创建库文件夹
2. git init 初始化库
3. 创建文件
4. git add /git commit -m “xxx” 添加文件 提交（使库完成master分支初始化）
5. git remote add <远程库在本地别名> <url> 添加远程库关联
6. git fetch拉取信息
7. git checkout -b <本地分支名> <远程库别名/远程要关联的分支名>
8. git pull

提交流程（默认已完成关联）

1. git status 查看哪些文件有变化
2. git add .(输入“.”代表add全部有变化文件，或可指定那个或那些文件要关联)
3. git reset HEAD <文件名表达式> 撤销符合表达式的文件的add状态
4. git checkout ./git clean -fd 将文件从暂存区移除
5. git commit -m “xx” 提交
6. git push

# maven杂记

通过maven构建父子项目关系。父项目中的<modules>标签指定了子项目，子项目中的parent标签指定了父项目

Mvn指令

跳过测试的install Mvn clean install –DskipTests

运行 mvn spring-boot:run

# Idea 快捷键

快捷键

idea keyMap快捷键

基于Eclipse模板修改

other->Show Intention Actions |F2

Main menu->Code→reformat code |Ctrl+I

Main menu→View→Compare Files |Alt+C

其他设置

注释模版

File and code template 右侧选 File Header

回车格式

Code style 根据系统需求设置回车格式

在idea中出现 Empty test suite的解决办法

File→ Project structure →将文件夹选为test类型（绿色的test图标）

# Linux

## 简单命令

创建目录 mkdir <dirname>

创建文件 touch <filename>

创建并修改文件 vi <filename>

删除文件 rm <filename>

显示所有java进程号与所执行的文件 jps -l（jdk安装后可用）

## 操作实记

1、项目部署

将项目build好，生成jar包

本机用SecureCrt进入linux虚拟机

检查是否已经装了rz和sz，rpm -qa|grep  sz和 rpm -qa|grep  rz

若没有安装则输入yum install sz和yum install rz

命令行输入rz，会弹出对话框，选择本机需要传输的jar文件。

文件传输正常之后即可。

检查是否有之前的java进程需要关闭，jps -l

若有，则输入 kill -9 <pid> 关闭进程

之后可选择执行.sh文件 ./xxxx.sh

或者命令行输入 nohup java -jar xxx.jar <启动参数> &

2、JDK配置

查看当前linux系统位数（getconf LONG\_BIT检查当前系统多少位）

准备好jdk的压缩文件。（官网下载，对应好系统类型与位数，找文件名以tar.gz结尾的）

搞到linux系统中（windows传入Linux使用rz命令（需安装），linux互传可以使用scp命令，使用VM可以直接拖曳进ubuntu）

解压jdk压缩文件（参考压缩于解压章节 ，命令：tar -xzf xxxxx.tar.gz）

编辑/etc/profile文件（如vim /etc/profile）

可以考虑使用vi，也可以使用vim或gedit等文本编辑工具，若需要安装vim等工具则考虑使用yum(yum本身有可能也需要安装)。(yum install vim -y)。刚进入文件是浏览模式，此时无法编辑，浏览模式下点击”i”开始编辑，点击esc返回到浏览模式，若要保存并退出在浏览模式输入” :wq”,若不保存仅退出输入”:q”。

在profile文件最下方空白处添加如下：

export JAVA\_HOME=/data/software/jdk1.8.0\_131

export PATH=$JAVA\_HOME/bin:$PATH

export CLASSPATH=$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

JAVA\_HOME写实际jdk路径，其余可以照抄。

之后可以使用注销来重新加载profile文件。或在/etc文件夹下输入指令”source profile”来强制重新加载profile文件。

## Ssh



## Scp

==================   
scp 命令   
==================   
scp 可以在 2个 linux 主机间复制文件；   
  
命令基本格式：   
       scp [可选参数] file\_source file\_target   
  
======   
从 本地 复制到 远程   
======   
\* 复制文件：   
        \* 命令格式：   
                scp local\_file remote\_username@remote\_ip:remote\_folder   
                或者   
                scp local\_file remote\_username@remote\_ip:remote\_file   
                或者   
                scp local\_file remote\_ip:remote\_folder   
                或者   
                scp local\_file remote\_ip:remote\_file   
  
                第1,2个指定了用户名，命令执行后需要再输入密码，第1个仅指定了远程的目录，文件名字不变，第2个指定了文件名；   
                第3,4个没有指定用户名，命令执行后需要输入用户名和密码，第3个仅指定了远程的目录，文件名字不变，第4个指定了文件名；   
        \* 例子：   
                scp /home/space/music/1.mp3 root@www.cumt.edu.cn:/home/root/others/music   
                scp /home/space/music/1.mp3 root@www.cumt.edu.cn:/home/root/others/music/001.mp3   
                scp /home/space/music/1.mp3 www.cumt.edu.cn:/home/root/others/music   
                scp /home/space/music/1.mp3 www.cumt.edu.cn:/home/root/others/music/001.mp3   
  
\* 复制目录：   
        \* 命令格式：   
                scp -r local\_folder remote\_username@remote\_ip:remote\_folder   
                或者   
                scp -r local\_folder remote\_ip:remote\_folder   
  
                第1个指定了用户名，命令执行后需要再输入密码；   
                第2个没有指定用户名，命令执行后需要输入用户名和密码；   
        \* 例子：   
                scp -r /home/space/music/ root@www.cumt.edu.cn:/home/root/others/   
                scp -r /home/space/music/ www.cumt.edu.cn:/home/root/others/   
  
                上面 命令 将 本地 music 目录 复制 到 远程 others 目录下，即复制后有 远程 有 ../others/music/ 目录   
  
  
======   
从 远程 复制到 本地   
======   
从 远程 复制到 本地，只要将 从 本地 复制到 远程 的命令 的 后2个参数 调换顺序 即可；   
  
例如：   
        scp root@www.cumt.edu.cn:/home/root/others/music /home/space/music/1.mp3   
        scp -r www.cumt.edu.cn:/home/root/others/ /home/space/music/

最简单的应用如下 :   
  
scp 本地用户名 @IP 地址 : 文件名 1 远程用户名 @IP 地址 : 文件名 2   
  
[ 本地用户名 @IP 地址 :] 可以不输入 , 可能需要输入远程用户名所对应的密码 .   
  
可能有用的几个参数 :   
  
-v 和大多数 linux 命令中的 -v 意思一样 , 用来显示进度 . 可以用来查看连接 , 认证 , 或是配置错误 .   
  
-C 使能压缩选项 .   
  
-P 选择端口 . 注意 -p 已经被 rcp 使用 .   
  
-4 强行使用 IPV4 地址 .   
  
-6 强行使用 IPV6 地址 .

注意两点：  
1.如果远程服务器防火墙有特殊限制，scp便要走特殊端口，具体用什么端口视情况而定，命令格式如下：  
#scp -p 4588 remote@www.abc.com:/usr/local/sin.sh /home/administrator  
2.使用scp要注意所使用的用户是否具有可读取远程服务器相应文件的权限。

## Cd 进入\退出目录

## Rm 删除文件

## Mv 移动

## Cp 复制

## More/Less

一、More指令

more功能类似 cat ，cat命令是整个文件的内容从上到下显示在屏幕上。 more会以一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能 。more命令从前向后读取文件，因此在启动时就加载整个文件。

1．命令格式：

more [-dlfpcsu ] [-num ] [+/ pattern] [+ linenum] [file ... ]

2．命令功能：

more命令和cat的功能一样都是查看文件里的内容，但有所不同的是more可以按页来查看文件的内容，还支持直接跳转行等功能。

3．命令参数：

+n      从笫n行开始显示

-n       定义屏幕大小为n行

+/pattern 在每个档案显示前搜寻该字串（pattern），然后从该字串前两行之后开始显示

-c       从顶部清屏，然后显示

-d       提示“Press space to continue，’q’ to quit（按空格键继续，按q键退出）”，禁用响铃功能

-l        忽略Ctrl+l（换页）字符

-p       通过清除窗口而不是滚屏来对文件进行换页，与-c选项相似

-s       把连续的多个空行显示为一行

-u       把文件内容中的下画线去掉

4．常用操作命令：

Enter    向下n行，需要定义。默认为1行

Ctrl+F   向下滚动一屏

空格键  向下滚动一屏

Ctrl+B  返回上一屏

=       输出当前行的行号

：f     输出文件名和当前行的行号

V      调用vi编辑器

!命令   调用Shell，并执行命令

q       退出more

5．命令实例：

实例1：显示文件中从第3行起的内容

命令：

aijian.shi@U-aijian-shi:~/ALM$ cat test.log                   #显示所有日志内容  
aijian.shi@U-aijian-shi:~/ALM$ more +3 test.log               #从第三行开始显示日志内容

输出：

[复制代码](javascript:void(0);)

1 aijian.shi@U-aijian-shi:~/ALM$ cat test.log

2 2016-8-1 aijian.shi

3 2016-8-2 yafang.wei

4 2016-8-3 hong.zhan

5 2016-8-4 yuyan.zhang

6 2016-8-5 senlin.zhao

7 2016-8-6 yanbin.liu

8 2016-8-7 hui.liu

9 2016-8-8 yanhua.liu

10 2016-8-9 baoixn.cui

11 2016-8-10 ge.song

12 2016-8-11 zhongjun.zhen

13 2016-8-12 qiu.liao  
aijian.shi@U-aijian-shi:~/ALM$ more +3 test.log  
2016-8-3 hong.zhan  
2016-8-4 yuyan.zhang  
2016-8-5 senlin.zhao  
2016-8-6 yanbin.liu  
2016-8-7 hui.liu  
2016-8-8 yanhua.liu  
2016-8-9 baoixn.cui  
2016-8-10 ge.song  
2016-8-11 zhongjun.zhen  
2016-8-12 qiu.lia

[复制代码](javascript:void(0);)

实例2.将日志内容设置为每屏显示4行

命令：

aijian.shi@U-aijian-shi:~/ALM$ more -4 test.log

输出：

[复制代码](javascript:void(0);)

1 aijian.shi@U-aijian-shi:~/ALM$ more -4 test.log

2 2016-8-1 aijian.shi

3 2016-8-2 yafang.wei

4 2016-8-3 hong.zhan

5 2016-8-4 yuyan.zhang

6

7 ...skipping one line              #这里使用ctrl+F或者空格键来滚屏

8 2016-8-6 yanbin.liu

9 2016-8-7 hui.liu

10 2016-8-8 yanhua.liu

11 2016-8-9 baoixn.cui

12

13 ...skipping one line

14 2016-8-11 zhongjun.zhen

15 2016-8-12 qiu.liao

[复制代码](javascript:void(0);)

实例3.从文件中查找第一个出现"liu"字符串的行，并从该处前两行开始显示输出

命令：

aijian.shi@U-aijian-shi:~/ALM$ more +/liu test.log

输出：

[复制代码](javascript:void(0);)

1 aijian.shi@U-aijian-shi:~/ALM$ more +/liu test.log

2

3 ...skipping

4 2016-8-4 yuyan.zhang

5 2016-8-5 senlin.zhao

6 2016-8-6 yanbin.liu

7 2016-8-7 hui.liu

8 2016-8-8 yanhua.liu

9 2016-8-9 baoixn.cui

10 2016-8-10 ge.song

11 2016-8-11 zhongjun.zhen

12 2016-8-12 qiu.liao

[复制代码](javascript:void(0);)

实例4.当一个目录下的文件内容太多，可以用more来分页显示。这得和管道 | 结合起来

命令：

aijian.shi@U-aijian-shi:~/ALM$ cat test.log | more -5   #“|”表示管道，作用是可以将前面命令的输出当做后面命令的输入

输出：

[复制代码](javascript:void(0);)

1 aijian.shi@U-aijian-shi:~/ALM$ cat test.log | more -5

2 2016-8-1 aijian.shi

3 2016-8-2 yafang.wei

4 2016-8-3 hong.zhan

5 2016-8-4 yuyan.zhang

6 2016-8-5 senlin.zhao

7

8 ...skipping one line

9 2016-8-7 hui.liu

10 2016-8-8 yanhua.liu

11 2016-8-9 baoixn.cui

12 2016-8-10 ge.song

13 2016-8-11 zhongjun.zhen

14 --more--

[复制代码](javascript:void(0);)

二、less指令

less 工具也是对文件或其它输出进行分页显示的工具，应该说是linux正统查看文件内容的工具，功能极其强大。less 的用法比起 more 更加的有弹性。 在 more 的时候，我们并没有办法向前面翻， 只能往后面看，但若使用了 less 时，就可以使用 [pageup] [pagedown] 等按 键的功能来往前往后翻看文件，更容易用来查看一个文件的内容！除此之外，在 less 里头可以拥有更多的搜索功能，不止可以向下搜，也可以向上搜。

1．命令格式：

less [参数]  文件

2．命令功能：

less 与 more 类似，但使用 less 可以随意浏览文件，而 more 仅能向前移动，却不能向后移动，而且 less 在查看之前不会加载整个文件。

3．命令参数：

-b <缓冲区大小> 设置缓冲区的大小

-e  当文件显示结束后，自动离开

-f  强迫打开特殊文件，例如外围设备代号、目录和二进制文件

-g  只标志最后搜索的关键词

-i  忽略搜索时的大小写

-m  显示类似more命令的百分比

-N  显示每行的行号

-o <文件名> 将less 输出的内容在指定文件中保存起来

-Q  不使用警告音

-s  显示连续空行为一行

-S  行过长时间将超出部分舍弃

-x <数字> 将“tab”键显示为规定的数字空格

/字符串：向下搜索“字符串”的功能

?字符串：向上搜索“字符串”的功能

n：重复前一个搜索（与 / 或 ? 有关）

N：反向重复前一个搜索（与 / 或 ? 有关）

b  向后翻一页

d  向后翻半页

h  显示帮助界面

Q  退出less 命令

u  向前滚动半页

y  向前滚动一行

空格键 滚动一行

回车键 滚动一页

[pagedown]： 向下翻动一页

[pageup]：   向上翻动一页

4．使用实例：

实例1：ps查看进程信息并通过less分页显示同时显示行号

命令：

aijian.shi@U-aijian-shi:~/ALM$ ps -ef|less -N

输出：

[复制代码](javascript:void(0);)

1 1 UID PID PPID C STIME TTY TIME CMD

2 2 root 1 0 0 Aug08 ? 00:00:00 /sbin/init

3 3 root 2 0 0 Aug08 ? 00:00:00 [kthreadd]

4 4 root 3 2 0 Aug08 ? 00:00:02 [ksoftirqd/0]

5 5 root 6 2 0 Aug08 ? 00:00:00 [migration/0]

6 6 root 7 2 0 Aug08 ? 00:00:00 [watchdog/0]

7 7 root 8 2 0 Aug08 ? 00:00:00 [migration/1]

8 8 root 9 2 0 Aug08 ? 00:00:00 [kworker/1:0]

9 9 root 10 2 0 Aug08 ? 00:00:01 [ksoftirqd/1]

10 10 root 11 2 0 Aug08 ? 00:00:00 [watchdog/1]

11 11 root 12 2 0 Aug08 ? 00:00:00 [migration/2]

12 12 root 14 2 0 Aug08 ? 00:00:01 [ksoftirqd/2]

13 13 root 15 2 0 Aug08 ? 00:00:00 [watchdog/2]

14 14 root 16 2 0 Aug08 ? 00:00:00 [migration/3]

15 15 root 18 2 0 Aug08 ? 00:00:01 [ksoftirqd/3]

16 16 root 19 2 0 Aug08 ? 00:00:00 [watchdog/3]

17 17 root 20 2 0 Aug08 ? 00:00:00 [migration/4]

18 18 root 22 2 0 Aug08 ? 00:00:00 [ksoftirqd/4]

19 19 root 23 2 0 Aug08 ? 00:00:00 [watchdog/4]

20 20 root 24 2 0 Aug08 ? 00:00:00 [migration/5]

21 21 root 26 2 0 Aug08 ? 00:00:00 [ksoftirqd/5]

22 22 root 27 2 0 Aug08 ? 00:00:00 [watchdog/5]

23 23 root 28 2 0 Aug08 ? 00:00:00 [migration/6]

24 :

[复制代码](javascript:void(0);)

实例2.浏览多个文件

命令：

aijian.shi@U-aijian-shi:~/ALM$ less test2.log test.log

输出：

[复制代码](javascript:void(0);)

1 1 ifconfig

2 2 ping www.baidu.com

3 3 ifconfig

4 4 //10.128.161.108/share

5 5 10.128.161.108/share

6 6 ssh

7 7 keygen

8 8 trsa

9 9 ssh

10 10 .ssh/

11

12

13 test2.log (file 1 of 2) (END) - Next: test.log

[复制代码](javascript:void(0);)

说明：

输入 ：n后，切换到 test.log

输入 ：p 后，切换到test2.log

ps：当正在浏览一个文件时，也可以使用 :e命令 打开另一个文件。

命令：

less file1

:e file2

5．附加备注

1.全屏导航

ctrl + F - 向前移动一屏

ctrl + B - 向后移动一屏

ctrl + D - 向前移动半屏

ctrl + U - 向后移动半屏

2.单行导航

j - 向前移动一行

k - 向后移动一行

3.其它导航

G - 移动到最后一行

g - 移动到第一行

q / ZZ - 退出 less 命令

4.其它有用的命令

v - 使用配置的编辑器编辑当前文件

h - 显示 less 的帮助文档

&pattern - 仅显示匹配模式的行，而不是整个文件

5.标记导航

当使用 less 查看大文件时，可以在任何一个位置作标记，可以通过命令导航到标有特定标记的文本位置：

ma - 使用 a 标记文本的当前位置

'a - 导航到标记 a 处

6.查找

more, less 都具备查找功能，按/ 然后输入要找的字串，再按 Enter 即可，按 n(next) 会继续找，大写的 N 则是往回(上)找，按 q(quit)或者ZZ离开

## Head/Tail

　　head 与 tail 就像它的名字一样的浅显易懂，它是用来显示开头或结尾某个数量的文字区块，head 用来显示档案的开头至标准输出中，而 tail 想当然就是看档案的结尾，看看下面的范例：

　　displays the first 6 lines of a file

　　head -6 readme.txt

　　(2) displays the last 25 lines of a file

　　tail -25 mail.txt

　　范例一是显示档案的前 6 行，范例二则是显示档案最后的 25 行。

　　而下面的范别，结合了 head 与 tail 的指令，显示档案的第 11 行到第 20 行：

　　head -20 file | tail -10

　 　在 tail 的使用手册页中显示了比 head 还多的可用参数，其中有一个很好用的参数 " -f "，使用此参数时，tail 不会回传结束信号，除非我们去自行去中断它；相反的，它会一直等待一段时间，一直到他发现资料自它最后一次被读取后，又被加入新的一行时：

*display ongoing updates to the given*

*log file*

*tail -f /usr/tmp/logs/daemon\_log.txt*

　　上述范例可以动态显示该 log 文件的动态更新。

　　假设该服务程序是一直不断的加入动态资料到*/usr/adm/logs/daemon\_log.txt* 的 log 文件里，在命令列控制窗口中使用 tail -f，它将会以一定的时间实时追踪该档的所有更新。 ( -f 的只有在其输入为档案时才能使用 )。

　　假如你在 tail 后下了多个档案参数，你便能在同一个窗口内一次追踪数个 log 档：

*track the mail log and the server error log*

*at the same time.*

*tail -f /var/log/mail.log /var/log/apache/error\_log*

tac -- 反过来串连?!

　 　cat 倒过来怎么拼 ? 对啦 !! 这就是 tac 的功能啰 ~ 它是把档案的顺序内容反过来串连用的，那么 ~ 它都用在什么状况下呢 ? 任何须要以后进先出的顺序重新排列组件的工作都用得上它 ! 以下面的指令来说，便是以自最后建立的到最先建立的顺序，列出三个最新建的使用者帐号：

[*l*](http://baike.baidu.com/view/59772.htm)*ast 3 /etc/passwd records - in reverse*

*$ tail -3 /etc/passwd | tac*

*curly:x:1003:100:3rd Stooge:/homes/curly:/bin/ksh*

*larry:x:1002:100:2nd Stooge:/homes/larry:/bin/ksh*

*moe:x:1001:100:1st Stooge:/homes/moe:/bin/ksh*

*查看文件内容的特殊方法*  
*相信最基本的cat和less,more你已经很熟悉了，如果有特殊的要求呢：*  
*1. 如果你只想看文件的前5行，可以使用head命令，如：*  
**head -5 /etc/passwd**  
*2. 如果你想查看文件的后10行，可以使用tail命令，如：*  
**tail -10 /etc/passwd**  
*tail -f /var/log/messages*  
*参数-f使tail不停地去读最新的内容，这样有实时监视的效果*

一、UNIX下的TAIL命令

说明：部分内容摘录自"永远的UNIX",http://fanqiang.chinaunix.net/

NAME(名称)

tail - 输出文件的末尾部分

SYNOPSIS(总览)

**../src/tail** [*OPTION*]... [*FILE*]...

DESCRIPTION(描述)

在标准输出上显示每个FILE的最后10行. 如果多于一个FILE,会一个接一个地显示, 并在每个文件显示的首部给出文件名. 如果没有FILE,或者FILE是-,那么就从标准输入上读取.

**--retry**

即使tail开始时就不能访问或者在tail运行后不能访问,也仍然不停地尝试打开文件. **--** 只与**-f**合用时有用.

**-c**, **--bytes**=*N*

输出最后N个字节

**-f**, **--follow[=**{name|descriptor}]

当文件增长时,输出后续添加的数据; **-f**, **--follow**以及 **--follow**=*descriptor* 都是相同的意思

**-n**, **--lines**=*N*

输出最后N行,而非默认的最后10行

**--max-unchanged-stats**=*N*

参看texinfo文档(默认为5)

**--max-consecutive-size-changes**=*N*

参看texinfo文档(默认为200)

**--pid**=*PID*

与**-f**合用,表示在进程ID,PID死掉之后结束.

**-q**, **--quiet**, **--silent**

从不输出给出文件名的首部

**-s**, **--sleep-interval**=*S*

与**-f**合用,表示在每次反复的间隔休眠S秒

**-v**, **--verbose**

总是输出给出文件名的首部

**--help**

显示帮助信息后退出

**--version**

输出版本信息后退出

如果N(字节或者行数)的第一个字符为`+', 那么从每个文件的开头算起的第N项开始显示, 否则, 显示该文件的最后N项. N可以有一个倍数前缀: b表示512,k表示1024,m表示1048576(1兆). 第一个选项**-VALUE** 或+VALUE 以**-n** VALUE或**-n** +VALUE 方式看待,除非VALUE含有[bkm]后缀倍数中的一个,在这种情况下,它被看作 **-c** VALUE 或者**-c** +VALUE

使用**--follow** (-f)时,tail默认后接文件描述符, 这意味着即使tail显示的文件改名了,tail仍然可以追踪到其末尾部分. 如果你确实希望查询文件的实际名而非文件描述符 (例如,日志轮转时), 这种默认的操作就不是你所期望的了. 在这种情况下应使用**--follow**=*name*. 这将导致tail通过周期性地重新打开所指定的文件来追踪其是否被删除了,或者被其他程序重新创建了.

**tail** 的完整文档是以Texinfo手册形式维护的. 如果 **info** 和 **tail** 程序在你那儿都已经安装好了,那么命令:**info tail** 应该会让你访问到整篇手册.

       从上面的说明可以看到，最常用的形式是：tail   -f   sys.log

## kill -9 <pid>

## nohup <程序名> &

让程序在后台运行的命令

## chmod 777 <文件名>

赋予文件权限

## 压缩与解压

Linux下最常用的打包程序就是tar了，使用tar程序打出来的包我们常称为tar包，tar包文件的命令通常都是以.tar结尾的。生成tar包后，就可以用其它的程序来进   
行压缩了，所以首先就来讲讲tar命令的基本用法：   
　　tar命令的选项有很多(用man tar可以查看到)，但常用的就那么几个选项，下面   
来举例说明一下：   
　　# tar -cf all.tar \*.jpg   
　　这条命令是将所有.jpg的文件打成一个名为all.tar的包。-c是表示产生新的包   
，-f指定包的文件名。   
　　# tar -rf all.tar \*.gif   
　　这条命令是将所有.gif的文件增加到all.tar的包里面去。-r是表示增加文件的   
意思。   
# tar -uf all.tar logo.gif   
　　这条命令是更新原来tar包all.tar中logo.gif文件，-u是表示更新文件的意思。   
  
　　# tar -tf all.tar   
　　这条命令是列出all.tar包中所有文件，-t是列出文件的意思   
　　# tar -xf all.tar   
　　这条命令是解出all.tar包中所有文件，-x是解开的意思   
　　以上就是tar的最基本的用法。为了方便用户在打包解包的同时可以压缩或解压   
文件，tar提供了一种特殊的功能。这就是tar可以在打包或解包的同时调用其它的压   
缩程序，比如调用gzip、bzip2等。   
　　**1) tar调用gzip**　　gzip是GNU组织开发的一个压缩程序，.gz结尾的文件就是gzip压缩的结果。与gzip   
相对的解压程序是gunzip。tar中使用-z这个参数来调用gzip。下面来举例说明一下   
：   
　　# tar -czf all.tar.gz \*.jpg   
　　这条命令是将所有.jpg的文件打成一个tar包，并且将其用gzip压缩，生成一个   
gzip压缩过的包，包名为all.tar.gz   
　　# tar -xzf all.tar.gz   
　　这条命令是将上面产生的包解开。   
　　**2) tar调用bzip2**　　bzip2是一个压缩能力更强的压缩程序，.bz2结尾的文件就是bzip2压缩的结果。   
与bzip2相对的解压程序是bunzip2。tar中使用-j这个参数来调用gzip。下面来举例   
说明一下：   
　　# tar -cjf all.tar.bz2 \*.jpg   
　　这条命令是将所有.jpg的文件打成一个tar包，并且将其用bzip2压缩，生成一个   
bzip2压缩过的包，包名为all.tar.bz2   
　　# tar -xjf all.tar.bz2   
　　这条命令是将上面产生的包解开。   
**3)tar调用compress**　　compress也是一个压缩程序，但是好象使用compress的人不如gzip和bzip2的人   
多。.Z结尾的文件就是bzip2压缩的结果。与 compress相对的解压程序是uncompress   
。tar中使用-Z这个参数来调用compress。下面来举例说明一下：   
　　# tar -cZf all.tar.Z \*.jpg   
　　这条命令是将所有.jpg的文件打成一个tar包，并且将其用compress压缩，生成   
一个uncompress压缩过的包，包名为all.tar.Z   
　　# tar -xZf all.tar.Z   
　　这条命令是将上面产生的包解开   
  
　　**有了上面的知识，你应该可以解开多种压缩文件了，下面对于tar系列的压缩文   
件作一个小结**：   
　　1)对于.tar结尾的文件   
　　tar -xf all.tar   
　　2)对于.gz结尾的文件   
　　gzip -d all.gz   
　　gunzip all.gz   
　　3)对于.tgz或.tar.gz结尾的文件   
　　tar -xzf all.tar.gz   
　　tar -xzf all.tgz   
　　4)对于.bz2结尾的文件   
　　bzip2 -d all.bz2   
　　bunzip2 all.bz2   
　　5)对于tar.bz2结尾的文件   
　　tar -xjf all.tar.bz2   
　　6)对于.Z结尾的文件   
　　uncompress all.Z   
　　7)对于.tar.Z结尾的文件   
　　tar -xZf all.tar.z   
  
　　另外对于Window下的常见压缩文件.zip和.rar，Linux也有相应的方法来解压它   
们：   
　　**1)对于.zip**　　linux下提供了zip和unzip程序，zip是压缩程序，unzip是解压程序。它们的参   
数选项很多，这里只做简单介绍，依旧举例说明一下其用法：   
　　# zip all.zip \*.jpg   
　　这条命令是将所有.jpg的文件压缩成一个zip包   
# unzip all.zip   
　　这条命令是将all.zip中的所有文件解压出来   
　　2)对于.rar   
　　要在linux下处理.rar文件，需要安装RAR for Linux，可以从网上下载，但要记住，RAR for Linux 不是免费的；可从<http://www.rarsoft.com/download.htm>下载RARfor Linux 3.2.   
0，然后安装：   
　　# tar -xzpvf rarlinux-3.2.0.tar.gz   
　　# cd rar   
　　# make   
　　这样就安装好了，安装后就有了rar和unrar这两个程序，rar是压缩程序，unrar 是解压程序。它们的参数选项很多，这里只做简单介绍，依旧举例说明一下其用法：   
  
　　# rar a all \*.jpg   
　　这条命令是将所有.jpg的文件压缩成一个rar包，名为all.rar，该程序会将.rar   
扩展名将自动附加到包名后。   
　　# unrar e all.rar   
　　这条命令是将all.rar中的所有文件解压出来   
　　到此为至，我们已经介绍过linux下的tar、gzip、gunzip、bzip2、bunzip2、compress 、 uncompress、 zip、unzip、rar、unrar等程式，你应该已经能够使用它们对.tar 、.gz、.tar.gz、.tgz、.bz2、.tar.bz2、. Z、.tar.Z、.zip、.rar这10种压缩文   
件进行解压了，以后应该不需要为下载了一个软件而不知道如何在Linux下解开而烦恼了。而且以上方法对于Unix也基本有效。   
　　本文介绍了linux下的压缩程式tar、gzip、gunzip、bzip2、bunzip2、compress 、uncompress、 zip、 unzip、rar、unrar等程式，以及如何使用它们对.tar、.gz 、.tar.gz、.tgz、.bz2、.tar.bz2、.Z、. tar.Z、.zip、.rar这10种压缩文件进行   
操作。   
  
**以下补充**  
tar   
  
-c: 建立压缩档案   
-x：解压   
-t：查看内容   
-r：向压缩归档文件末尾追加文件   
-u：更新原压缩包中的文件   
  
这五个是独立的命令，压缩解压都要用到其中一个，可以和别的命令连用但只能用其中一个。下面的参数是根据需要在压缩或解压档案时可选的。   
  
-z：有gzip属性的   
-j：有bz2属性的   
-Z：有compress属性的   
-v：显示所有过程   
-O：将文件解开到标准输出   
  
下面的参数-f是必须的   
-f: 使用档案名字，切记，这个参数是最后一个参数，后面只能接档案名。   
# tar -cf all.tar \*.jpg这条命令是将所有.jpg的文件打成一个名为all.tar的包。-c是表示产生新的包，-f指定包的文件名。   
# tar -rf all.tar \*.gif   
这条命令是将所有.gif的文件增加到all.tar的包里面去。-r是表示增加文件的意思。   
# tar -uf all.tar logo.gif   
这条命令是更新原来tar包all.tar中logo.gif文件，-u是表示更新文件的意思。   
# tar -tf all.tar   
这条命令是列出all.tar包中所有文件，-t是列出文件的意思   
# tar -xf all.tar   
这条命令是解出all.tar包中所有文件，-x是解开的意思   
压缩   
tar –cvf jpg.tar \*.jpg //将目录里所有jpg文件打包成tar.jpg   
tar –czf jpg.tar.gz \*.jpg //将目录里所有jpg文件打包成jpg.tar后，并且将其用gzip压缩，生成一个gzip压缩过的包，命名为jpg.tar.gz   
tar –cjf jpg.tar.bz2 \*.jpg //将目录里所有jpg文件打包成jpg.tar后，并且将其用bzip2压缩，生成一个bzip2压缩过的包，命名为jpg.tar.bz2   
tar –cZf jpg.tar.Z \*.jpg //将目录里所有jpg文件打包成jpg.tar后，并且将其用compress压缩，生成一个umcompress压缩过的包，命名为jpg.tar.Z   
rar a jpg.rar \*.jpg //rar格式的压缩，需要先下载rar for linux   
zip jpg.zip \*.jpg //zip格式的压缩，需要先下载zip for linux   
  
解压   
tar –xvf file.tar //解压 tar包   
tar -xzvf file.tar.gz //解压tar.gz   
tar -xjvf file.tar.bz2 //解压 tar.bz2   
tar –xZvf file.tar.Z //解压tar.Z   
unrar e file.rar //解压rar   
unzip file.zip //解压zip   
  
总结   
1、\*.tar 用 tar –xvf 解压   
2、\*.gz 用 gzip -d或者gunzip 解压   
3、\*.tar.gz和\*.tgz 用 tar –xzf 解压   
4、\*.bz2 用 bzip2 -d或者用bunzip2 解压   
5、\*.tar.bz2用tar –xjf 解压   
6、\*.Z 用 uncompress 解压   
7、\*.tar.Z 用tar –xZf 解压   
8、\*.rar 用 unrar e解压   
9、\*.zip 用 unzip 解压