

## 愚人乙 linux 基础个人学习笔记 (www.net-add.com)

一、 Linux 文件 7 种类型及帮助.....	2
二、 linux 目录结构及作用.....	2
三、 文件、目录等相关操作.....	3
● 通配符和正则表达式概念理解.....	3
● 查看、统计、排序、去重文件内容.....	4
● 创建文件、复制文件、剪切文件、删除文件、修改时间（略）.....	4
● vim 编辑器.....	4
四、 用户管理.....	5
五、 权限管理.....	6
5.1 基本权限.....	6
5.2 高级权限.....	6
5.3 acl 文件访问控制列表.....	6
六、 各种服务应用.....	7
6.1 Mysql 数据库.....	7
1、 mysql 命令.....	7
2、 相关概念.....	8
3、 SQL 语言（从库表、库表属性、记录三个层面理解）.....	8
4、 mysql 的备份.....	10

## 一、Linux 文件 7 种类型及帮助

— 普通文件      d 目录文件      b 块设备文件      c 字符设备文件

L 符号连接文件      p 管道文件      S 套接字网络文件

**ls** -l 详细显示 -a 显示隐藏文件 -A 显示所有文件 (不包括.和..) -h 人性显示

-d 显示当前目录 -t 按时间排序 -r 反向排序 -R 递归查询

系统帮助

1、ls -help

2、info ls

3、type ls 类型      which ls 位置路径

4、man ls

man 1 用户命令      \*

man 2 系统调用

man 3 库调用

man 4 特殊文件

man 5 配置文件      \*

man 6 游戏

man 7 杂项

man 8 系统管理员命令      \*

## 二、Linux 目录结构及作用

/bin    /usr/bin    /usr/local/bin      所有用户可执行的目录

/sbin    /usr/sbin    /usr/local/sbin      系统管理员可执行的目录

/lib /usr/bin    /usr/local/lib      库文件 (32 位)

/lib64 /usr/lib64 /usr/local/lib64      库文件 (64 位)

/etc    配置文件目录

    /etc/bashrc    全局 shell 级别换将配置文件

    /etc/profile    全局登录级别环境配置文件

    /etc/fstab(文件系统挂载信息)

    /etc/passwd&/etc/shadow(账号信息)

    /etc/shells(shell 命令的解释环境)

    /etc/rc.local 是/etc/rc.d/rc.local 的软连接 (启动系统执行的最后一个脚本)、/etc/login.defs(创建用户时用户的默认配置)

    /etc/mdadm.conf(raid)的配置文件

    /etc/ntp.conf (ntp) 时间服务的配置文件

    /etc/rsyslog.conf 日志服务的配置文件

    /etc/exports (NFS) 服务的配置文件

    /etc/samba/smb.conf (samba) 文件的配置文件

    /etc/ssh/sshd\_config (SSH) 服务的配置文件

    /etc/dhcp/dhcpd.conf (dhcp) 服务的配置文件

    /etc/vsftpd/vsftpd.conf (VSFTP) 服务的配置文件

    /etc/httpd/conf/httpd.conf (APACHE) 服务的配置文件

    /etc/named.conf (DNS) 服务的主配置文件

- /etc/named.rfc1912.zones (DNS) 服务的域名定义配置文件
- /etc/my.cnf (mysql) 数据库的配置文件
- /boot 启动目录 (引导程序、内核、启动文件)
  - 内核文件及 grub
  - 建议单独分区, 分区大小 100M 即可
- /usr 应用程序目录
  - /usr/bin(存放用户应用程序)
  - /usr/share(存放共享数据)
  - /usr/lib(一些程序运行所必须的一些函数库文件)
  - /usr/local(源码软件包的安装路径)
  - 建议单独分区, 设置较大的磁盘空间
- /var 服务器数据目录 (数据、日志)
  - /var/log(系统日志文件如 messages 各种消息、secure 安全相关、wtmp 保密存储等)
  - /var/ftp/(VSFTP)服务的数据目录
  - /var/www/html/ (Apache) 服务的数据服务
  - /var/named/ (DNS) 服务的文件解析记录
  - /var/lib/mysql/ (mysql) 数据库的数据目录
  - /var/lib/mysql/mysql.sock (mysql) 数据库的套接字默认存放位置
  - 建议单独分区, 设置较大的磁盘空间
- /home 普通用户的家目录
  - 建议单独分区, 设置较大的磁盘空间, 方便用户存放数据
- /cgroup 资源控制目录
- /dev 设备文件目录 (块设备、字符设备)
- /tmp 临时文件目录
- /proc 伪文件系统, 进程信息和内存信息目录
  - /proc/cpuinfo(cpu 信息)
- /sys 伪文件系统, 进程信息和内存信息目录, 硬件信息
- /root root 的家目录
- /media 媒介目录
- /mnt 其它挂载点
- /opt 第三方软件
- /selinux selinux 组件的目录
- /misc 和/net autofs 服务使用的目录
- /srv 服务启动后需要访问的数据目录, 如 WWW 服务需要访问的网页存在/srv/www

### 三、文件、目录等相关操作

- 通配符和正则表达式概念理解

通配符和正则表达式是两回事儿, 通配符是用 shell 作为解析器解析的, 常见的有\*?【】, 用于文件名的通配, 而且默认只通配本级目录, 通常用在 shell 命令和 shell 命令脚本中, 而正则表达式是由支持正则表达式的命令解析的, 用于字符串的匹配操作, 如 grep、awk 命令。在 shell 里输入命令后, shell 把整条命令读入, 再进行解析, 通常先由 shell 进行通配解析, 再传递给命令, 但可以通过 ‘ ‘ \脱义后直接交给命令解析。

● 查看、统计、排序、去重文件内容

- 1、 **more** /root/install.log | 回车 向下翻行| 空格 向下翻页 | **b** 向上翻页 | **q** 退出
- 2、 **less** /root/install.log | 可以上下翻页, 查找等 | **q** 退出
- 3、 **head** /root/install.log (默认显示前 install.log 文件的前 10 行)  
**head -n 30 /root/install.log** 显示 install.log 的前 30 行
- 4、 **tail** /root/install.log(默认显示 install.log 的后 10 行)  
**tail -f -n 20 /var/log/messages**(动态显示日志后 20 行, **service network restart** 看效果)
- 5、 **cat** /root/install.log  
**cat -n /root/install.log** 显示 install.log 的行号 (所有行)  
**cat -b /root/install.log** 显示 install.log 的行号 (有效行)  
 例: **cat -n /root/install.log | head -n 1000 | tail -n 1** 显示 install.log 的第 1000 行
- 5、 **wc** -l -w -c /etc/passwd(依次统计 passwd 文件的行数、单词数、字符数)
- 6、 **sort** aa.txt(按第一个字符给 aa.txt 排序)|-n -r 按第一个数字逆向排序|-u 合并重复行|
- 7、 **uniq** aa.txt(合并 aa.txt 的连续重复行)|-u 去掉连续重复行|-d 显示重复行|-d-c 统计重复行
- 8、 **grep** root /etc/passwd(显示 passwd 里有 root 的行)  
**grep ^root /etc/passwd**(以 root 开头, root\$以 root 结尾)  
 -c 找到字符的个数, -n 加行号, 程序调试时-n-A3-B2 找的位置显示行号, 并上 3 下 2 行
- 9、 **cut** -b 字节 -c 字符 -f 段  
**cut -d: -f 1-3,6 /etc/passwd**(取 passwd 中以冒号分割后的第 1 到 3 段和第 6 段)

● 创建文件、复制文件、剪切文件、删除文件、修改时间 (略)

- 例: **mkdir -v abc{1..10}** 看结果  
**mkdir -v {a,b,c}{1..10}** 看结果  
**mkdir -pv aaa/bbb -v** 显示过程 -p 递归  
**date** 月日時分年.秒  
**date 032416502015.30** 3月24日16时50分2015年  
**date -s** 单独修改时间 **hwclock** 或 **clock -w** 同步到 cmos  
**mv /tmp/aaa /tmp/bbb** 移动文件或文件夹同时改名  
**cp -r -a /tmp/aaa /tmp/bbb** 递归复制文件夹同时改名保持属性 -r 递归 -a 保持属性

● vim 编辑器

配置文件 /etc/vimrc

基本模式:命令模式 输入模式 末行模式

命令模式

home 键 行首 end 键 行尾

u 撤销一步操作 ctrl+r 重做

yy 复制一行 30yy 复制 30 行 p 粘贴(下一行) P 粘贴(上一行)

dd 剪切或者删除一行 30dd 剪切 30 行

^ 行首 \$ 行尾

d^ 删除当前字符到行首(不包含当前字符) d\$ 删除当前字符到行尾(包含当前字符)

y^ y\$ 同上

gg 首行 G 尾行 750G 定位行

dgg 删除当前行到首行 dG 删除当前行到尾行

YGG 同上

命令模式进入末行模式(shift+:)

```

:w :q :wq 保存退出
:w! :q! :wq!
:e /root/aa.txt 打开文件
:w /root/new.txt 另存文件
:20,40w /root/kk.txt 另存 20-40 行
:r /root/kk.txt 追加读入文件
:e! 重新打开当前文件
设置行号 :set nu :set nonu 显示符号 :set list 不区分大小写 :set ic
永久保存设置
vim 配置文件:/etc/vimrc
vim /etc/vimrc
set nu
set autoindent
查找 /或者?查找的字符串 n N
替换 :%s/ab/xx/g %所有行 s 替换 g 全局
:3,8s/ab/xx/g 3 行到 8 行替换 交互式替换
:%s/a/#a/ 将 a 替换为#a :%s/^/#/在每行前边加# :3,7s/^/#/ 3 到 7 行行首加#
:%s/ //g 替换所有的空格 :%s/^ //g 去掉所有行开头一个空格 l 的空格
:%s/^ *// 去掉所有行开头的空格 :%s/\t//g 替换所有的制表符
:3,5s/\n// 替换 3-5 行的换行符
:g/^$/d 删除空行(真正意义的空行) :g/\s*$/d 删除空行
    
```

#### 四、用户管理

ld

uid=0	管理员	gid=0	管理组
uid=1-499	系统用户	gid=1-499	系统组
uid=500-60000	普通用户	gid=500-60000	普通用户组

保存用户信息文件 /etc/passwd

1、 useradd/userdel/groupadd/groupdel 略

2、 **usermod -g** root robin(将 robin 用户的主组改为 root)

usermod -G aaa robin(将 robin 的附加组覆盖为 aaa)

usermod -a -G aaa robin(将 robin 的附加组追加 aaa 组)

usermod -u 2000 robin(修改 uid),-c 修改用户描述,-s 修改登录 shell,-d 修改家目录

3、用户锁定

**usermod -L** robin (锁) **usermod -U** robin (解锁) shadow 密码前加! , 只能锁有密码的

**passwd -l** robin(锁) **passwd -u** robin(解锁) shadow 密码前加!! , 有无密码全锁

**passwd -S** 看密码状态

5、 **chown** 修改所属用户和组

**chown** robin file/dir 属于 robin 用户

**chown** .uplooking file/dir 属于 uplooking 组

**chown** robin.uplooking file/dir 属于 robin 用户、uplooking 组

**chown -R** robin.uplooking file/dir 递归 dir 下所有的目录与文件

例: `usermod -G root,aaa,bbb robin` 将 robin 用户附属于 root,aaa,bbb 的用户组  
`gpasswd -M robin,root,bbb aaa` 将 aaa 组包含 robin、root、bbb 多个用户

## 五、权限管理

### 5.1 基本权限

#### 1、系统目录的权限 权限无递归，只能作用到目录一级

/	r-xr-xr-x
/etc/	rwxr-xr-x
/tmp/	rw-rw-rw-
/var/	rwxr-xr-x
/root/	r-xr-x---
/home/robin/	rw-----
/etc/shadow	-----
/etc/passwd	rw-r--r--
root 用户创建文件的权限	rw-r--r--
root 用户创建目录的权限	rwxr-xr-x
普通用户创建文件的权限	rw-rw-r--
普通用户创建目录的权限	rw-rw-r--

#### 2、系统权限反掩码

`umask 022 (root)`  
`umask 002 (普通用户)`

### 5.2 高级权限

#### 1、suid

suid 必须作用在 2 进制可执行文件，只能给拥有者添加

`chmod u+s file`

`chmod u-s file`

作用:任何用户在执行拥有 suid 权限的命令时,都以该命令拥有者的身份运行

#### 2、sgid

sgid 必须作用在目录上，只能给所属组添加

`chmod g+s dir/test`

`chmod g-s dir/test`

作用:任何用户在拥有 sgid 权限的目录下创建的文件,都要继承目录的 1 组

#### 3、sticky(t) 会递归作用在目录下面的子目录以及子文件

t 权限只能作用在目录上，只能添加到 other 上

`chmod o+t dir/`

`chmod o-t dir/`

作用:任何用户在拥有 t 权限的目录下创建的文件,其他人无权删除,只能自己删除

例 `chmod 3777 test/` 将目录 test 增加 sgid、sticky 超级，并将基本权限改为 777

### 5.3 acl 文件访问控制列表

挂载时通过参数控制是否支持 acl 功能，例如通过 `dumpe2fs -h /dev/sda1` 查看超级块是否支持 acl,如果不支持，可以通过 `mount -o acl,rw /dev/sda1 /` 重新挂载。

作用：对于某一文件，设定其针对特定用户特定组的权限，通过 getfacl 查看，会产生一个 acl 访问列表，这个列表与基本权限叠加使用，权限情况按顺序查看。

```
例 setfacl -m u:robin:rw aa.txt      添加针对用户 robin 的读写权限
   getfacl aa.txt                    查看权限
   setfacl -m g:uplooking:rw aa.txt  添加针对 uplooking 组的读写权限
   setfacl -x u:robin aa.txt         删除针对 robin 用户的 acl 权限
   setfacl -x g:uplooking aa.txt     删除针对 uplooking 组的 acl 权限
   setfacl -b aa.txt                 移除 aa.txt 的所有 acl 权限
```

## 六、各种服务应用

### 6.1 Mysql 数据库

rpm 包安装包：

```
rpm -ivh mysql-server-5.1.66-2.el6_3.x86_64.rpm 服务端
rpm -ivh mysql-5.1.66-2.el6_3.x86_64.rpm 客户端
```

mysql 配置文件：/etc/my.cnf

mysql 的数据目录：/var/lib/mysql/

mysql 的套接字文件位置：/var/lib/mysql/mysql.sock

mysql 端口：3306

#### 1、mysql 命令

##### 1) shell 命令

```
mysql -u root -p123 -S /var/lib/mysql/mysql.sock 登录数据库，制定用户密码和 socket
mysql -u root -p123 -e "select * from mysql.user\G" (在 shell 下执行 sql 语句)
mysqladmin -u root -p123 password abc (shell 下修改密码)
mysqladmin -u root -p123 proc (查看所有用户的活动线程)
```

##### 2) 登陆后的一些命令

```
show databases; 查看库      show create database bd; 查看 bd 库创建属性
show tables; 查看库中表     show create table t1; 查看 t1 表的创建属性
show index from t1; 查看 t1 表中索引
use bd; 切换库到 bd
desc t1; 描述表 t1 的栏目属性
select user();函数，查看当前用户
select database();函数，查看当前数据库
```

##### 用户相关

```
create user zorro@localhost identified by '123';创建本地用户 zorro 密码是 123
set password for zorro@localhost=password('111')修改本地 zorro 密码为 111
```

##### 注：mysql 丢失密码后的找回

将 mysql 停掉，以跳过权限表的方式登录

```
Service mysql stop
```

```
mysql_safe --skip-grant-tables --user=mysql &
```

mysql 登录后

```
update mysql.user set password=password('123') where user='root' and host='localhost';
```

重启 mysql 正常登录

## 2、相关概念

### 1) 索引

**普通索引:**针对某列建索引,无约束,方便查询,一张表可以有多个。

**Unique 索引:**索引列不允许有重复内容,一张表可以有多个。

**主键索引:**索引列不允许有重复内容,而且不能为 null,相当 unique+not null,一张表只有一个主键,可以将多列搭建成一个主键。

### 2) 外键

本表的一个字段建立外键,关联另外一个表的字段,被关联表字段必须唯一,一般为主键,一张表中视情况可以有多个外键,只能是 InnoDB 引擎支持外键。可以在 creat 创建表时建,也可以用 alter 建立。

### 3) 存储引擎(针对表)

Mysql 表在存储时支持不同的存储引擎,默认是 myisam,可以通过 show create table t1 查看。

**Myisam** 不支持事物、不支持外键、可跨平台进行数据物理(copy 数据)迁移、支持表级锁定、支持全文类型索引、效率更高。

**InnoDB** 支持事物、支持外键、支持行级锁、不可跨平台进行物理数据(copy 数据)迁移。

**Mrg\_myisam** 支持将字段属性完全相同的表变成一张 mrg\_myisam 引擎的表,其实新建表本身只是连接,不占数据,被连接的表不用是 mrg\_mysiam 引擎。

**Memory** 引擎,内存级别的存储引擎,速度快,但是数据不保存。

**注:事务**处理是一种机制,用来管理成批执行的 sql 语句,以保证数据库不包含不完整的操作结果,他们或者为整体执行完成,或者完全不执行(如果没有错误发生整组语句提交到数据库,如果发生错误,则进行回退,以保证数据的安全)。

## 3、SQL 语言(从库表、库表属性、记录三个层面理解)

### 1) ddl 数据库定义语言 Data Definition Language (针对库表、属性)

#### create

create database bd character set = 'utf8';创建数据库 bd,采用 utf-8 编码

create table t1(id int zerofill not full default 100,name char(10),index(id)) engine=innodb;创建表 t1,有 id 和 name 两个字段, id 不许为空,默认是 100,以零补齐数位,并在 id 列建立索引,存储引擎是 innodb

create table t2(id int unique,name char(10));创建表 t2,将 id 作为 unique 索引

Create table t3(id int primary key auto\_increment,name char(10)) engine=innodb;创建表 t3,id 作为主键索引,并且自增长

create table t4(id int,fk int,foreign key (fk) references t3(id)) engine=innodb;创建表 t4,将 fk 作为外键,被关联的是 t3 表的主键 id,被关联表与本表都是 innodb 存储引擎。

create user robin@'localhost' identified by '123';创建 robin 用户,密码是 123。

#### drop

drop database bd; 删除数据库

drop table t1; 删除表

drop user robin; 删除用户

#### alter (库表属性操作)

alter database bd character set='utf8'; 修改数据库的字符编码为 uft-8;

alter talbe t1 rename t2;将表 t1 改名为表 t2

alter table t1 add name char(10) after id;在 t1 表增加 name 字段放在 id 字段的后面(字段的增改删)

alter table t1 modify name varchar(10);更改字段 name 的数据类型为 varchar(10)(字段的增改删)

alter table t1 change name xm varchar(10);将 t1 的 name 字段改名为 xm (字段的增改删)  
 alter table t1 drop xm;将 t1 的 xm 字段删除 (字段的增改删)  
 alter table t1 add index(id);将 t1 表的 id 字段设置索引 (索引的增删)  
 show index from t1;查询 id 字段索引名 (索引的增删)  
 alter table t1 drop index 索引名;删除 id 字段的索引 (索引的增删)  
 alter table t1 add primary key(id);将 t1 表的 id 列设置为主键 (主键的增删)  
 alter table t1 drop primary key ;将 t1 表的主键删除 (主键的增删)  
 alter table t1 add foreign key(id) references t2(id);将表 t1 的 id 字段设为外键, 指向 t2 表的 id 字段, (外键的增删)  
 show create table t1;查看外键的名字 (外键的增删)  
 alter table t1 drop foreign key 外键名字 (外键的增删)

## 2) dml 数据操纵语言 Data Manipulation Language (针对数据记录)

### Select

select id,name from t1 order by id; 按 id 排序  
 select id,name from t1 order by id,name;先按 id 排序 id 相同在按 name 排序  
 select id,name from t1 order by id desc; 按 id 反向排序  
 select id,name from t1 order by id desc,name; 先按 id 反向排序再按名字排序  
 select id,name,sal from t1 order by sal desc limit 1;查找工资最高的人  
 select id,name,sal from t1 where id=10 or id=20 and sal > 5000; id 为 10 的 或者 id 为 20 并且薪水大于 5000 的;and 优先执行  
 select id,name,sal from t1 where (id=10 or id=20) and sal > 5000;id 为 10 或者 20 并且薪水大于 5000 的记录  
 select id,name,sal from t1 where id in (10,20,30);搜索 id 在 10 20 30 中的记录  
 select id,name,sal from t1 where id not in (10,20,30); id 不在 10 20 30 的记录  
 select sex,count(sex),avg(sal) group by sex;查找男女的人数以及平均工资  
 函数 sum () max () min () avg () count () year () month () day () hour () minute () second ()

### Insert

insert into t1(id,name) values(1,'robin');在 t1 表中插入 id、name 两个字段  
 insert into t1(id,name) values(2,'robin'), (3,'tom');连续插入两个记录

### update

update t1 set email = 'kyo@163.com' where member\_id = 3;

### delete

delete from t1 where id = 1;

## 3) dcl 数据库控制语言 Data Control Language (针对用户权限)

### grant

grant 权限 on 库.表 to 用户@主机; 授权同时可创建用户  
 grant 权限 on 库.表 to 用户@主机 identify by '123'; 授权同时创建用户密码为 123  
 grant all on \*.\* to robin@localhost;授权 robin 在本地登录并且有所有库的所有权限

### revoke

revoke 权限 on 库.表 from 用户@主机;  
 revoke select on hr.\* from robin@localhost;去掉 robin 用户在 hr 库的 select 权限  
 revoke all on \*.\* from robin@192.168.1.1;去掉允许 robin 在 192.168.1.1 主机登录的所有权限

## 4) tcl 事物控制语言 Transaction Control Language (针对事物 innodb 存储引擎)

Start transaction 开启事物 (期间只能做 dml 语言, 负责事物会提交退出)

savepoint 设置保存点

rollback 回退到操作点之前, 并退出事物

commit 提交, 并退出事物

例: Start transaction;开启事物

一系列操作

Savepoint s1;保存一个事物点

一系列操作

savepoint s2; 保存第二个事物点

一系列操作

rollback to s2;回到事物点 s2

rollback to s1;回到事物点 s1

rollback; 回到事物最初

#### 4、mysql 的备份

分为物理备和逻辑备, 又可分为离线备和在线备, 又分冷备、温备 (只读不写)、热备 (可读写)  
物理备份区分存储引擎, myisam 可以直接考库备份, 但 innodb 不可以, 因为他们的数据是放在共享 ibdata1 中、事物信息放在 ib\_logfile0 和 ib\_logfile1 中。

##### 1) mysqlhotcopy 物理在线备 (适合 myisam 引擎, 温备 shell 下使用的命令)

需要安装包 perl-DBD-MySQL, 只能备库, 其过程是把所有表锁定, 然后将库拷贝到另外一个目录

例: mysqlhotcopy -u root -p 123 mysql /tmp 将 mysql 库物理备份到/tmp 下

##### 2) 逻辑备份 mysqldump (shell 下备份 sql 语言, 支持温备和 innodb 的热备)

Mysql 自带工具

温备: mysqldump -u root -p123 --all-databases > /tmp/all.sql 备份全部数据库

Mysqldump -u root -p123 -databases mysql db > /tmp/mysql.sql 备份 mysql 和 db 库

Mysqldump -u root -p123 mysql user > /tmp/user.sql 备份 mysql 的 user 表

热备 (innodb)

Mysqldump -u root -p123 --single-transaction --all-databases > /tmp/all.sql

备份恢复: mysqldump -u root -p123 < /tmp/all.sql

##### 常用参数

**mysqldump -u root -p123 --all-databases --lock-all-tables --flush-logs > /tmp/all.sql**

备库时刷新 binlog, 默认是锁一个库备份一个库刷新一次, 加上 --lock-all-tables 全局锁, binlog 只刷新一次

**mysqldump -u root -p123 --all-databases --flush-logs --master-data > /tmp/all.sql**

备库时刷新 binlog, --master-data=(1 或 2)是激活 --lock-all-tables 并讲 change master to 的信息写入备份的 all.sql 文件中, 日志只刷新一次, 常用

**mysqldump -u root -p123 --all-databases --flush-logs --master-data --delete-master-logs > /tmp/all.sql**

在上一条的基础上, 刷新 binlog 的同时, 将之前没用的 binlog 删除掉

##### 3) 表的导入与导出

select \* from t9 into outfile '/backup/a.txt'; 将表 t9 导出在 /backup/a.txt

load data infile '/backup/a.txt' into table t9 fields terminated by '\t' lines terminated by '\n'; 将 a.txt 文件导入到数据库 t9 中, 结构必须一致

##### 4) binlog 日志 (将数据库的每一步操作进行命令记录)

启动: mysqld\_safe 启动时加 --log-bin 参数, 或在配置文件中加入

```
[mysqld]
Log-bin=/backup/master
Log-bin-index=/backup/master
max-binlog-size=防止 binlog 大小
```

看状态: show binlog events\G

切换日志:flush logs;

Shell 下, mysqladmin -u root -p123 flush-logs

恢复 binlog(shell 下):

```
mysqlbinlog master.000001 |mysql -u root -p123
mysqlbinlog `cat master-bin.index` | mysql
```

如果在日志内有错误,我们可以根据日志里的时间进行部分恢复;

```
mysqlbinlog --stop-datetime='2012-03-22 23:20:44' master-bin.000004 | mysql
```

根据 binlog 日志里的字节数进行恢复

```
mysqlbinlog --stop-position=274 master-bin.000004 | mysql
```

使用 binlog-do-db 可以指定记录的数据库;

使用 binlog-ignore-db 可以指定不记录的数据库。

注意的是: binlog-do-db 和 binlog-ignore-db 一次只指定一个数据库, 指定多个数据库需要多个语句。而且, MySQL 会将所有的数据库名称改成小写, 在指定数据库时必须全部使用小写字母, 否则不会起作用。

### 5) AB 复制

在 mysql 的配置文件/etc/my.cnf 中修改

```
主机
[mysqld]
log-bin=/backup/master
log-bin-index=/backup/master
server-id=1
```

```
从机
[mysqld]
Server-id=2
master-host=192.168.1.111
master-user=slave
master-password=123
```

查看主从机状态: show master status\G show slave status\G

如遇到从机两个进程启动不来, 可手动制定, 先停掉从机 stop slave;指定后再启 start slave;

```
mysql>change master to
->master_host='192.168.1.131',
-> master_user='slave',
->master_password='123',
->master_log_file="",
->master_log_pos=399;
```

互主配置

```
主机 1
[mysqld]
log-bin=mysql_bin
server-id=1
master-host=192.168.100.130
master-user=slave2
master-password=123
auto-increment-increment=2
auto-increment-offset=1
```

```
主机 2
[mysqld]
log-bin=slave
server-id=2
master-host=192.168.100.128
master-user=slave
master-password=123
auto-increment-increment=2
auto-increment-offset=2
```

## 6) AB 复制后实现读写分离

mysql-proxy 实现 mysql 的读写分离

```
rpm -ivh lua-5.0.2-1.el5.rf.i386.rpm    依赖包, 用 lua 语言写的
rpm -ivh lua-devel-5.0.2-1.el5.rf.i386.rpm  依赖包, 用 lua 语言写的
tar zxvf mysql-proxy-0.8.0-linux-rhel5-x86-32bit.tar.gz -C /usr/local/src  绿色软件, 非系统自带
cp -r mysql-proxy-0.8.0-linux-rhel5-x86-32bit /usr/local/mysql-proxy
cd /usr/local/mysql-proxy/share/doc/mysql-proxy
vim rw-splitting.lua (配置文件)
36 if not proxy.global.config.rwsplit then
37     proxy.global.config.rwsplit = {
38         min_idle_connections = 1, 最小空闲线程连接
39         max_idle_connections = 1, 最大线程连接, 基本没用
启动: ./mysql-proxy -P 192.168.1.15:3306 -b 192.168.1.14:3306 -r 192.168.1.140:3306 -s
/usr/local/mysqlproxy/shaer/doc/mysql-proxy/rw-splitting.lua --daemon &
```

-P 决定 mysql 用那个接口进行连接, 指定 mysql-proxy 工作的时候允许访问的 IP 和接口  
模拟成 mysql, 端口用 3306, 要保证自己本地的 mysql 不要运行, 打开后端端口冲突, 运行不起来

-r 只读的后端服务器 192.168.1.140

-b 读写的后端服务器 192.168.1.14

-s 工作的时候指定脚本的位置和名称, 默认是没有的

版本小于 5.1.12, 会出现 bug, 要加上 -bug

--daemon 以一个服务的形式去启动

改变: #ulimit -n 10240 临时改变系统打开文件个数, 永久改变, 要改 PAM

## Transaction Splitting

