# Day28笔记

## Linux上jdk,mysql,tomcat安装【重点】

Windows 控制面板 添加/卸载程序

进行程序的安装、更新、卸载、查看

rpm命令：相当于windows的添加/卸载程序

进行程序的安装、更新、卸载、查看

本地程序安装：rpm -ivh 程序名

本地程序查看：rpm -qa

本地程序卸载：rpm -e --nodeps 程序名

yum命令：相当于可以联网的rpm命令

相当于先联网下载程序安装包、程序的更新包

自动执行rpm命令

#### 准备工作：

因为JDK，TOMCAT，MYSQL的安装过程中需要从网上下载部分支持包才可以继续，所以要求同学们提前安装下载好依赖

yum install glibc.i686

yum -y install libaio.so.1 libgcc\_s.so.1 libstdc++.so.6

yum update libstdc++-4.4.7-4.el6.x86\_64

yum install gcc-c++

|  |  |
| --- | --- |
| glibc.i686 | 8MB+21MB |
| libaio.so.1 libgcc\_s.so.1 libstdc++.so.6 | 417KB |
| libstdc++-4.4.7-4.el6.x86\_64 | 仅更新检查，  无下载内容 |
| yum install gcc-c++ | 19MB |
|  |  |

#### Linux系统上安装JDK：

#### 安装步骤：

##### 卸载OpenJDK

执行命令查看:

* rpm –qa | grep java



* 卸载OPENJDK

rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.i686

rpm -e --nodeps java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.i686

##### 创建JDK的安装路径：

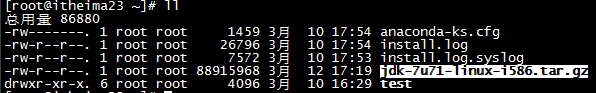
在/usr/local/ 创建文件夹java目录

mkdir -p /usr/local/java

##### 上传安装文件到linux:



* cd ~
* ll



##### 解压tar.gz

首先需要安装依赖：【1~5分钟】

* tar -zxvf jdk-7u71-linux-i586.tar.gz -C /usr/local/java

##### 配置环境变量：

vim /etc/profile

#set java environment

JAVA\_HOME=/usr/local/java/jdk1.7.0\_71

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

PATH=$JAVA\_HOME/bin:$PATH

export JAVA\_HOME CLASSPATH PATH

重新加载配置文件：【否则环境变量不会重新执行】

source /etc/profile

#### Linux系统上安装MySQL:

#### 安装MySQL:

##### 卸载自带mysql:

查询mysql的安装情况

rpm -qa | grep mysql –-color

卸载原生的MySQL

rpm -e --nodeps mysql-libs-5.1.71-1.el6.i686

##### 上传mysql到Linux:



##### 安装依赖（准备阶段已经安装过了）：

yum -y install libaio.so.1 libgcc\_s.so.1 libstdc++.so.6

yum update libstdc++-4.4.7-4.el6.x86\_64

##### 安装mysql的服务端：

安装服务端

rpm -ivh MySQL-server-5.5.49-1.linux2.6.i386.rpm

##### 安装mysql的客户端：

rpm -ivh MySQL-client-5.5.49-1.linux2.6.i386.rpm

##### 启动mysql的服务：

启动MySQL服务

service mysql start

##### 设置mysql初始密码并登陆MySQL:

/usr/bin/mysqladmin -u root password '123456'

##### 设置开机自动启动mysql:

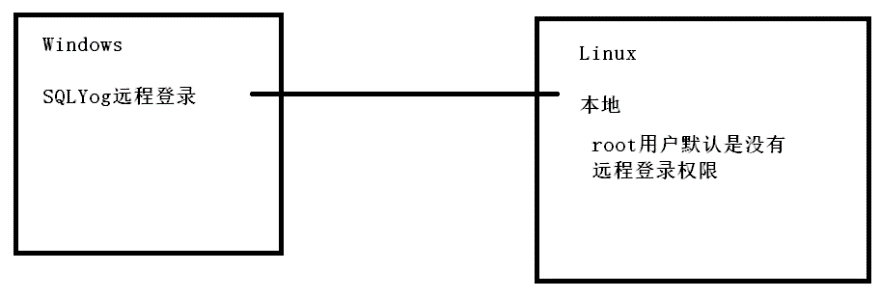
加入到系统服务：

chkconfig --add mysql

自动启动：

chkconfig mysql on

##### 开启远程服务：



登录mysql:

//赋予root用户所有权限，远程登录密码是123456

grant all privileges on \*.\* to 'root' @'%' identified by '123456';

flush privileges;

##### 设置Linux的防火墙:

Linux防火墙默认拦截3306端口

/sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

/etc/rc.d/init.d/iptables save

#### Linux系统上安装tomcat：

#### 安装tomcat:

##### 上传tomcat的安装文件：

##### 创建tomcat的安装路径：

mkdir -p /usr/local/tomcat

##### 解压tomcat

tar -zxvf apache-tomcat-7.0.57.tar.gz -C /usr/local/tomcat

##### 设置防火墙：

开放8080端口

/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT

保存配置

/etc/rc.d/init.d/iptables save

##### 启动tomcat：

启动tomcat

执行 /usr/local/tomcat/apache-tomcat-7.0.57/bin/startup.sh

关闭tomcat

执行 /usr/local/tomcat/apache-tomcat-7.0.57/bin/shutdown.sh

## redis简述及安装

关系型数据库（SQL）：

Mysql，oracle

特点：数据和数据之间，表和字段之间，表和表之间是存在关系的

例如：部门表 001部分， 员工表 001

用户表，用户名、密码

分类表 和 商品表， 一对多关系。

优点：①数据之间有关系，进行数据的增删改查时非常方便的。

②关系型数据库 有事务操作。 保证数据的完整性

缺点：1、因为数据和数据之间有关系的，关系是由底层大量算法保证

大量算法会拉低系统运行速度

大量算法会消耗系统资源

2、海量数据的增删改查时会显得无能为力。

很可能宕机

3、海量数据环境下对数据表进行维护/扩展，也会变得无能为力

update product set cname = ‘手机数码’;//修改所有数据

把商品表的cname字段，由varchar(64), char(100)

适合处理一般量级数据，安全。

钱

非关系型数据库（NOSQL）：

为了处理海量数据，需要将关系型数据库的关系 去掉。

非关系型数据库设计之初 是为了替代 关系型数据库的

Redis

优点：1、海量数据的增删改查，非常轻松应对

2、海量数据的维护非常轻松。

缺点：1、数据和数据之间没有关系，所以不能一目了然

2、非关系型数据库，没有关系，没有强大的事务保证数据的 完整和安全

适合处理海量数据，效率。不一定安全

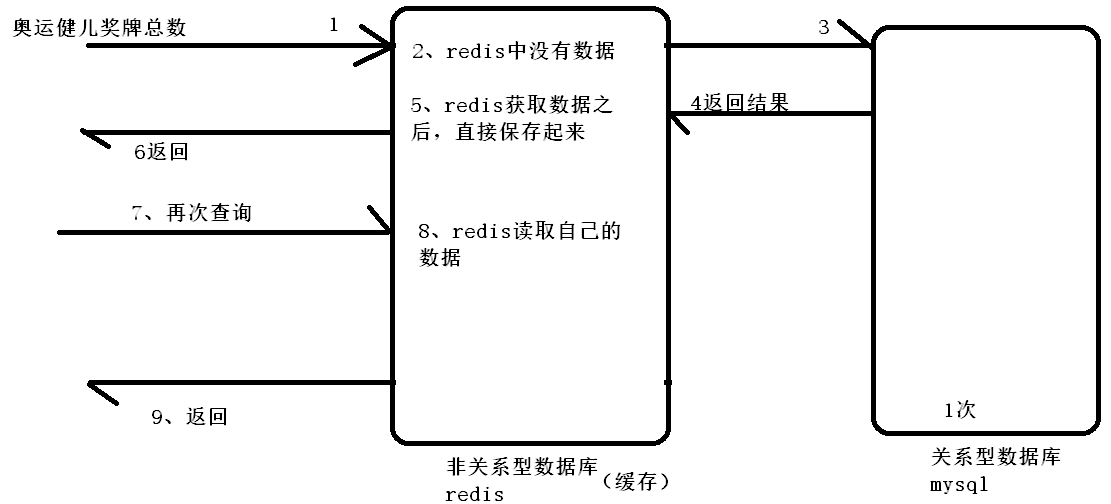
奥运健儿奖牌总数

关系型数据库+非关系型数据库 ====》项目

重要数据 海量操作数据，不重要

Redis使用环境：

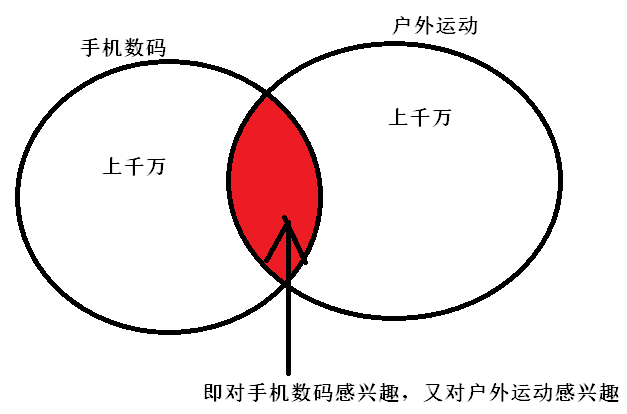
1、关系型数据库的缓存存在



2、可以做任务队列



3、大量数据运算



4、排行榜

Redis非常擅长做大量数据的排行榜

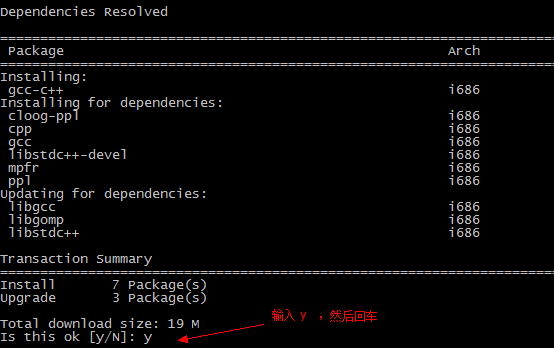
redis安装：

redis是C语言开发，安装redis需要先将官网下载的源码进行编译，编译依赖gcc环境。如果没有gcc环境，需要安装gcc：（环境已经导入完成）

yum install gcc-c++



如果提示是否下载，输入y。



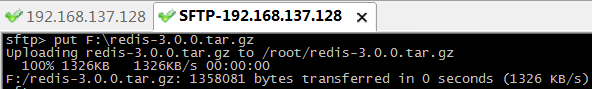
如果提示是否安装，输入y



* 步骤1：将Windows下下载的压缩文件上传到Linux下。通过secureCRT进行上传，步骤如下：

 alt + p

 put F:/redis-3.0.0.tar.gz

 

* 步骤2： 解压文件

 tar –zxvf redis-3.0.0.tar.gz



* 步骤3： 编译redis (编译，将.c文件编译为.o文件)

 进入解压文件夹，cd redis-3.0.0

 执行**make**



1）如果没有安装gcc，编译将出现错误提示。（如果安装失败，必须删除文件夹，重写解压）

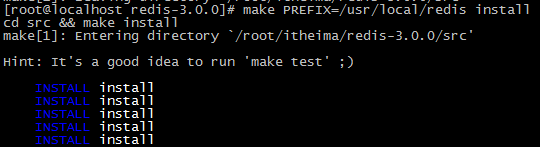


2）安装成功



* 步骤4： 安装

**make PREFIX=/usr/local/redis install**

 

安装完后，在/usr/local/redis/bin下有几个可执行文件

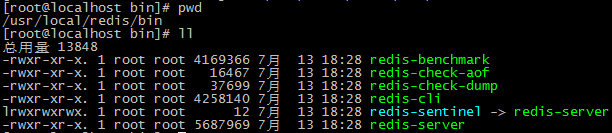
redis-benchmark ----性能测试工具

redis-check-aof ----AOF文件修复工具

redis-check-dump ----RDB文件检查工具（快照持久化文件）

**redis-cli** ----命令行客户端

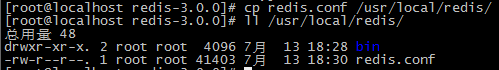
**redis-server** ----redis服务器启动命令



* 步骤5： copy文件

redis启动需要一个配置文件，可以修改端口号等信息。

cp redis.conf /usr/local/redis



**注：如果没有配置文件redis也可以启动，不过将启用默认配置，这样不方便我们修改端口号等信息**

* 前端启动：（了解）-无法部署集群



启动redis，客户端连接： 连接6379端口

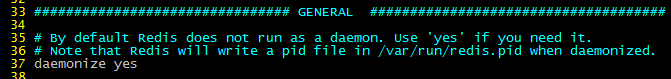
redis-cli -h ip地址 -p 端口

./bin/redis-cli -p 6379

运行客户端（默认连接本机6379端口）：./bin/redis-cli

* Redis启动-后端模式
* 修改redis.conf配置文件， daemonize yes 以后端模式启动。

vim /usr/local/redis/redis.conf



* 启动时，指定配置文件

cd /usr/local/redis/

./bin/redis-server ./redis.conf



* Redis默认端口6379，通过当前服务进行查看

ps -ef | grep -i redis

redis的关闭：

1. 查询到PID,kill -9 pid 【断电，非正常关闭，一般不用，否则造成数据丢失】

②正常关闭 【正常关闭，数据保存】

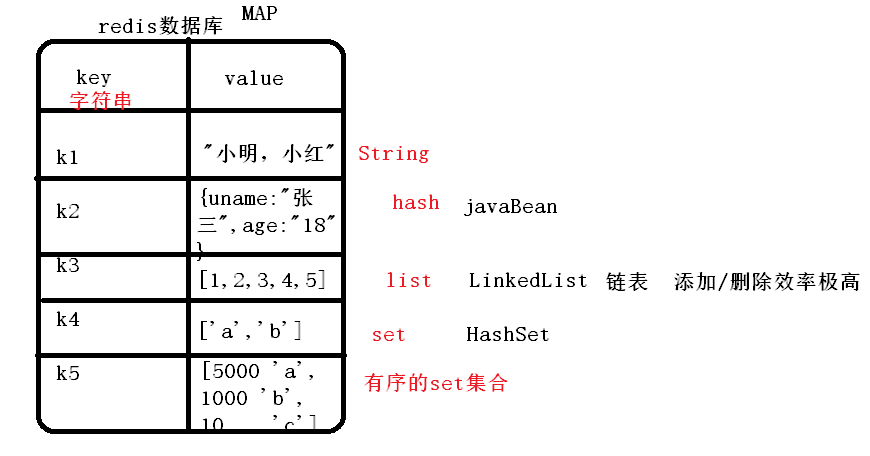
**./bin/redis-cli shutdown**

## 3、redis数据类型【重点】

redis 使用的是键值对 保存数据。（map）

key:全部都是字符串

value:有五种数据类型



Key名：自定义，key名不要过长，否则影响使用效率

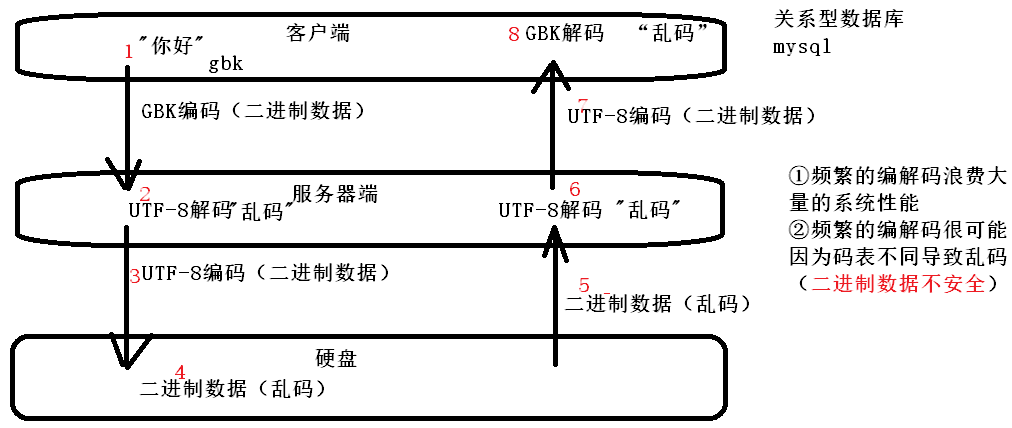
Key名不要太短，最好是有意义。

## 4、redis命令-String命令【重点】

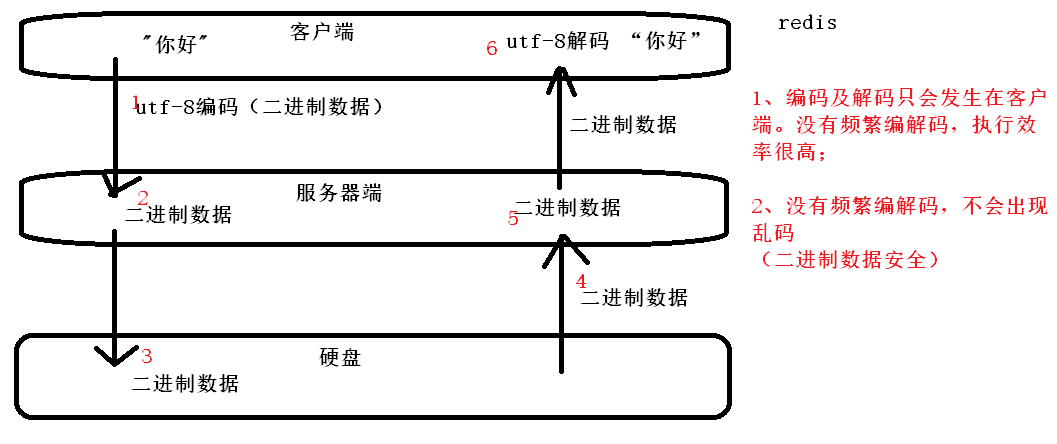
字符串类型是Redis中最为基础、常用的数据存储类型，**字符串在Redis中是二进制安全的**，这便意味着该类型存入和获取的数据相同。在Redis中字符串类型的Value最多可以容纳的数据长度是512M。

二进制安全和数据安全是没有关系的。

MySQL-关系型数据库，二进制不安全。【乱码丢失数据】



Redis 二进制数据安全



赋值：

* **set key value**：设定key持有指定的字符串value，如果该key存在则进行覆盖操作。总是返回”OK”

如果赋予相同的key，新的value会覆盖老的value

取值：

* **get key**：获取key的value。如果与该key关联的value不是String类型，redis将返回错误信息，因为get命令只能用于获取String value；**如果该key不存在，返回(nil)**。

删除：

* **del key** ：删除指定key

返回值是数字类型，表示删了几条数据

扩展：

* getset key value：先获取该key的值，然后在设置该key的值。
* incr key：将指定的key的value原子性的递增1.如果该key不存在，其初始值为0，在incr之后其值为1。如果value的值不能转成整型，如hello，该操作将执行失败并返回相应的错误信息。

相当于 ++i

* decr key：将指定的key的value原子性的递减1.如果该key不存在，其初始值为0，在incr之后其值为-1。如果value的值不能转成整型，如hello，该操作将执行失败并返回相应的错误信息。

相当于 –i

Incr和decr 只能对字符串是数字的 进行操作。

* **append key value**：拼凑字符串。如果该key存在，则在原有的value后追加该值；如果该key不存在，则重新创建一个key/value

String使用环境：

**主要用于保存json格式的字符串**

## 5、redis命令-hash（了解）

Redis中的Hash类型可以看成具有String Key和String Value的map容器。所以该类型非常适合于存储值对象的信息。如Username、Password和Age等。如果Hash中包含很少的字段，那么该类型的数据也将仅占用很少的磁盘空间。每一个Hash可以存储4294967295个键值对。

Hash ----🡪 {username:”张三”,age:”18”,sex:”man”}------javaBean

Hash特点：占用的磁盘空间极少

赋值：

* hset key field value：为指定的key设定field/value对（键值对）。
* **hmset key field value [field2 value2 …]**：设置key中的多个filed/value

取值：

* hmget key filed1 filed2 ….：获取key中的多个filed的值
* hgetall key：获取key中的所有filed-vaule

删除：

* **del key** ：删除整个hash

扩展：

* hkeys key ：获得所有的字段
* hvals key：获得所有的value

keys \* 查询所有的key

## 6、Jedis（java操作redis数据库技术）【重点】

Redis有什么命令，Jedis就有什么方法。

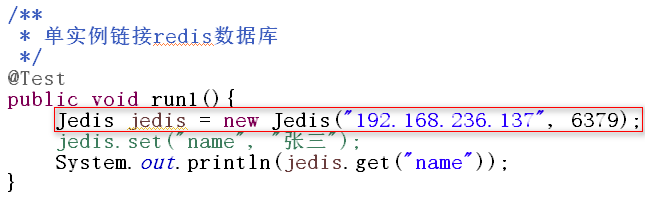
防火墙设置：

开放6379端口

/sbin/iptables -I INPUT -p tcp --dport 6379 -j ACCEPT

保存配置

/etc/rc.d/init.d/iptables save



Jedis连接池：



抽取连接池工具：

JedisUtils

