1. Redis 安装（172.16.129.90/root/123456）
2. wget http://download.redis.io/releases/redis-2.8.19.tar.gz
3. $ tar xzf redis-2.8.19.tar.gz
4. $ cd redis-2.8.19
5. $ make
6. 启动命令：src/redis-server
7. Utils/redis\_init\_script通过初始化脚本启动redis，使Redis能随系统自动运行
8. src/redis-server redis.conf 按照配置文件中的设置启动redis：

daemonize: 默认情况下，redis不是在后台运行的，如果需要在后台运行，把该项的值更改为yes。

logfile /var/log/redis/redis.log 日志文件，默认为/dev/null

loglevel debug 日志级别

1. 进入命令界面：src/redis-cli
2. 关闭redis服务：src/redis-cli shutdown
3. Redis Command ：<http://redis.io/commands>
4. Redis 支持的数据类型包括字符串类型，散列表类型，列表类型，集合类型，有序集合类型

Redis 字典顺序：0<9<A<Z<a<z

1. 字符串： set key value / get key value
2. 散列表类型（对象存储）： Hset key field value/hget key field

Hmset key field value [field value …]/hmget key field [field …]

/Hgetall key

1. 列表类型（字符串列表）： lpush key value[value…]

Rpush key value[value…]

Lrange key start stop(0至-1显示全部)

1. 集合类型（字符串集合）：sadd key member [member…]

Smembers key

1. 事务（multi/exec/watch）, 过期时间（expire/ttl）,排序（sort/by/get/store），消息通知，队列实现（brpop key [key …]timeout，blpop key timeout）,发布/订阅模式（publish/subscribe/unsubscribe/psubscribe/punsubscribe）
2. Redis 内存数据结构：http://www.searchtb.com/2011/05/redis-storage.html

**typedef** **struct** redisObject {

    unsigned type:4;  // 类型

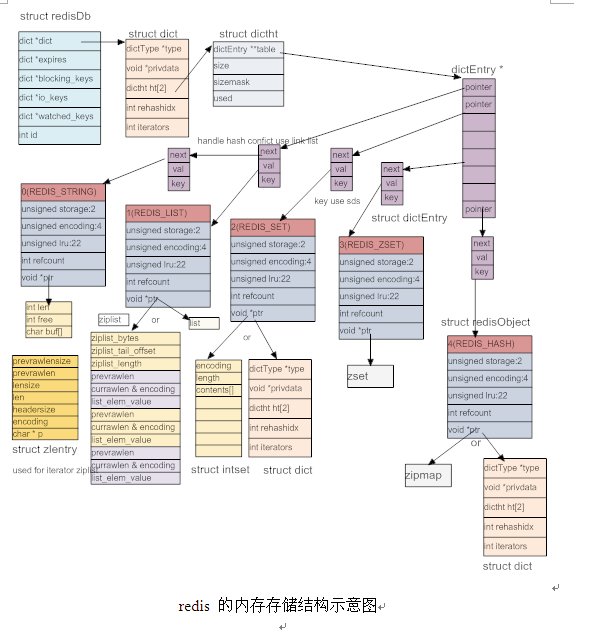
   unsigned encoding:4;    // 编码

  unsigned lru:REDIS\_LRU\_BITS; // 对象最后一次被访问的时间 /\* lru time (relative to server.lruclock) \*/

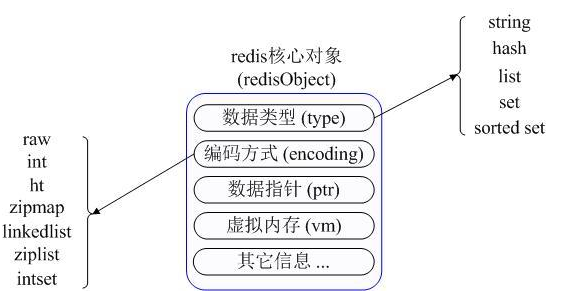
**int** refcount;    // 引用计数

**void** \*ptr;    // 指向实际值的指针

} robj;



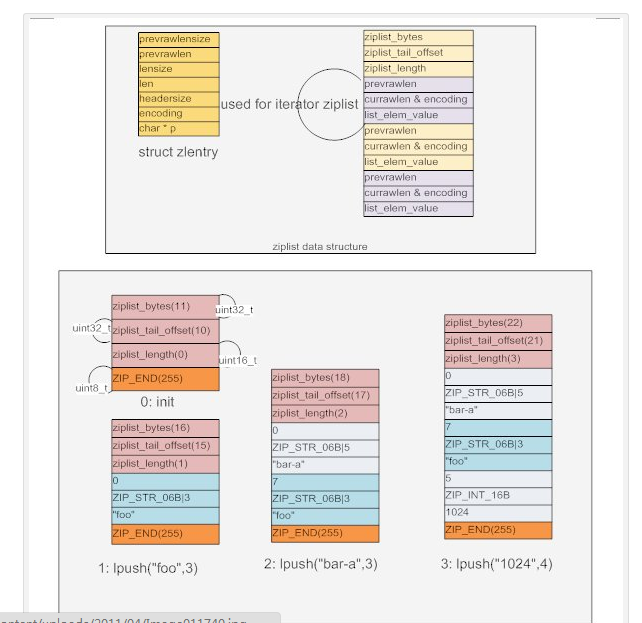
内部编码优化：http://blog.csdn.net/freebird\_lb/article/details/7733994



a. 字符串类型：sds(Simple Dynamic Strings)，使用sdshdr类型存储字符串，而redisObject的ptr字段指向的是该变量的地址，当健值内容可以用64位有符号整数表示时，redis会将键值转化为long类型来存储，redis启动后会预先建立10000个分别存储0到9999这些数字的redisObject类型变量作为共享对象。

struct sdshdr {   
　　　　int len; // buf 中已被使用的字符串空间数量   
　　　 int free; 　// buf 中预留字符串空间数量   
　　　　char buf[]; // 实际储存字符串的地方   
　　};

b. 散列类型： redis-encodeing-ziplist：



c. 集合类型

typedef struct intset {  
  uint32\_t encoding;  
  uint32\_t length;  
   int8\_t contents[];  
} intset;

encoding有三种:

#define INTSET\_ENC\_INT16 (sizeof(int16\_t))  
#define INTSET\_ENC\_INT32 (sizeof(int32\_t))  
#define INTSET\_ENC\_INT64 (sizeof(int64\_t))