目录

[一、 Redis服务器 1](#_Toc40465268)

[(一) 命令请求的执行过程 1](#_Toc40465269)

[1) 执行预备操作 2](#_Toc40465270)

[2) 调用命令的实现函数 3](#_Toc40465271)

[3) 执行后续工作 3](#_Toc40465272)

[(二) serverCron函数 4](#_Toc40465273)

[1) 更新服务器时间缓存 4](#_Toc40465274)

[2) 更新LRU时钟 4](#_Toc40465275)

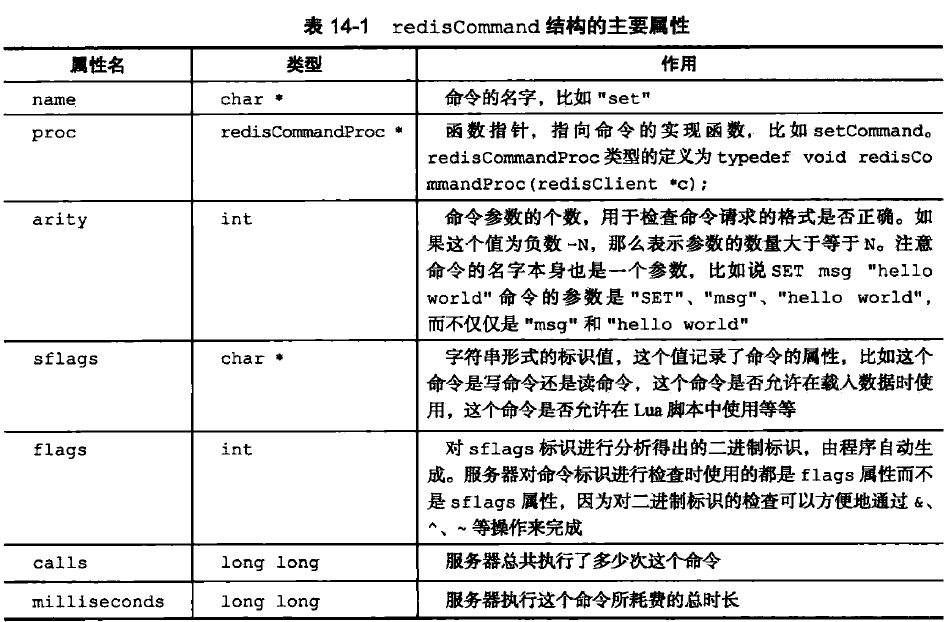
[3) 更新服务器每秒执行命令次数 5](#_Toc40465276)

[4) 更新服务器内存峰值记录 6](#_Toc40465277)

# Redis服务器

## 命令请求的执行过程

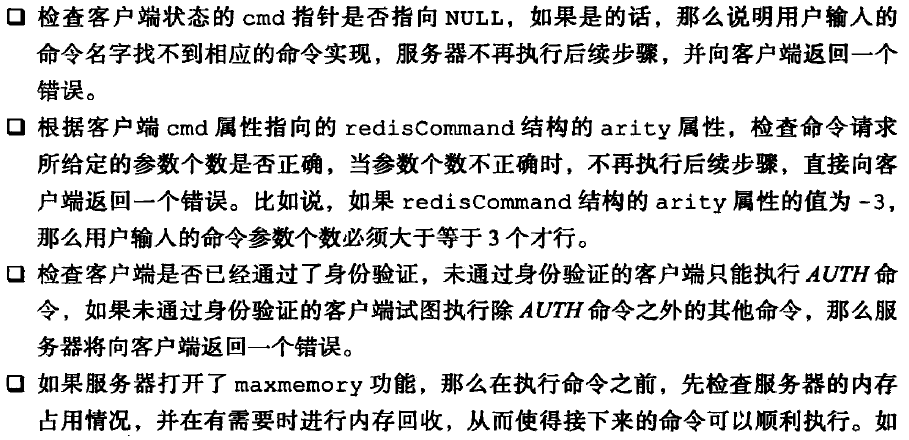
命令表是一个字典，字典的键是命令名字，值则是redisCommond结构。每一个redisCommond结构记录了一个让Redis命令的实现信息。

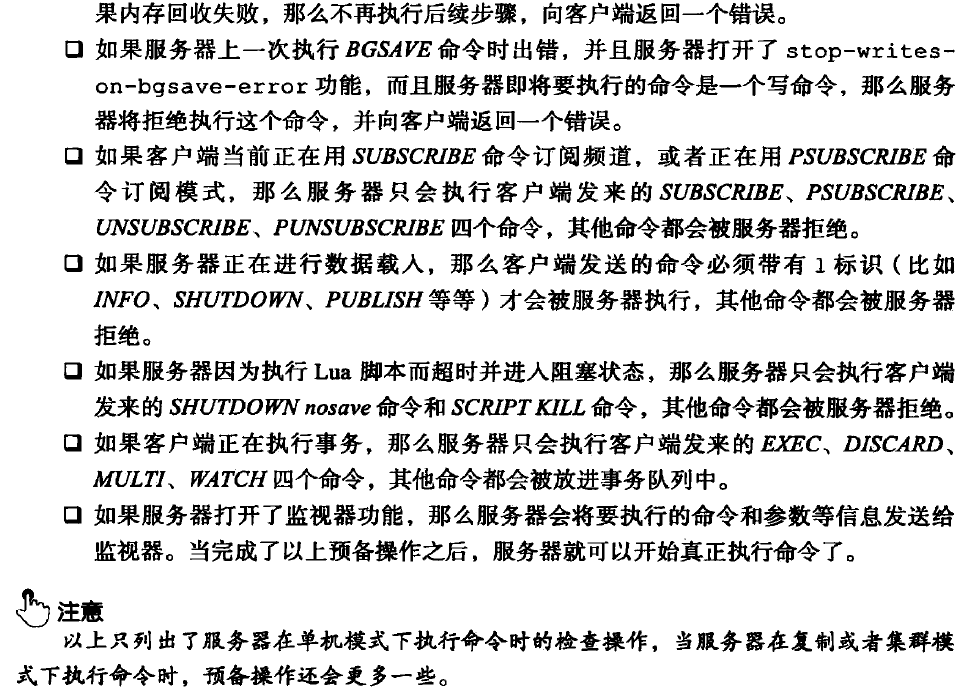




### 执行预备操作

为确保命令正确，顺利地被执行，这些操作包括：

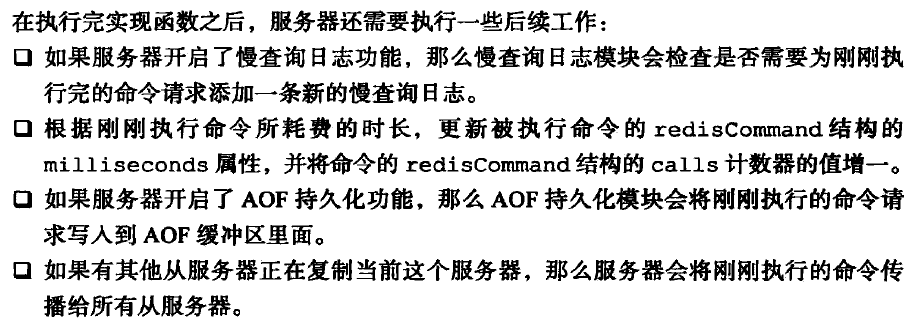




### 调用命令的实现函数

client->cmd->pro(client); //client是指向客户的指针

### 执行后续工作



## serverCron函数

Redis服务器中的serverCron函数默认每隔100ms执行一次，这个函数负责管理服务器的资源，并保持服务器的良好运转。

### 更新服务器时间缓存

Struct redisServer{

//…

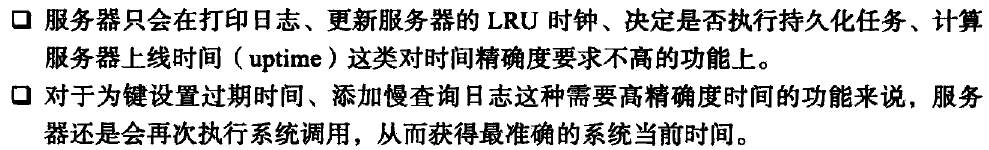
time\_t unixtime; //保存了秒级禁锢的系统当前UNIX时间戳

long long mstime; //保存了毫秒级精度的系统当前UNIX时间戳

//…

}

serverCron函数默认每隔100ms更新unixtime和mstime属性，因此这两个属性的精度并不高



### 更新LRU时钟

Struct redisServer{

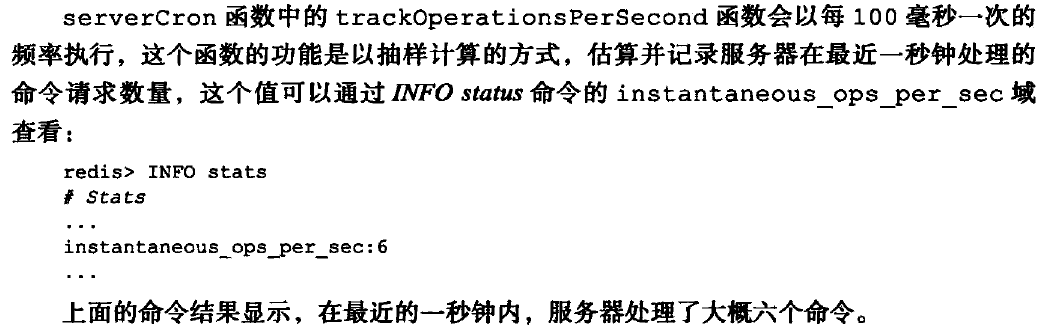
//…

Unsigned lruclock:22; //默认每10s更新一次，用于计算键的空转时长

//…

}

### 更新服务器每秒执行命令次数



Struct redisServer{

//…

Long long ops\_sec\_last\_sample\_time; //上次抽样的时间

Long long ops\_sec\_last\_sample\_ops; //上次抽样的数量

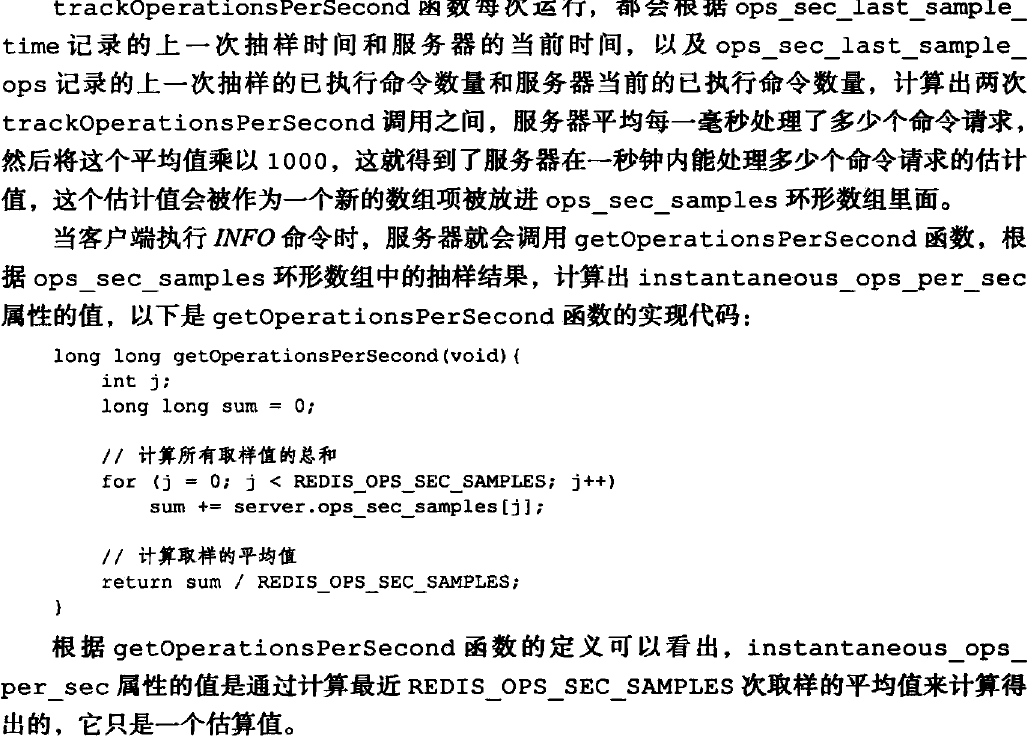
Long long ops\_sec\_samples[REDIS\_OPS\_SEC\_SAMPLES]; //记录每次抽样的结果

//REDIS\_OPS\_SEC\_SAMPLES默认为16

Int ops\_sec\_idx; //每次抽样后自增1，16时变回0

//…

}



### 更新服务器内存峰值记录

Struct redisServer{

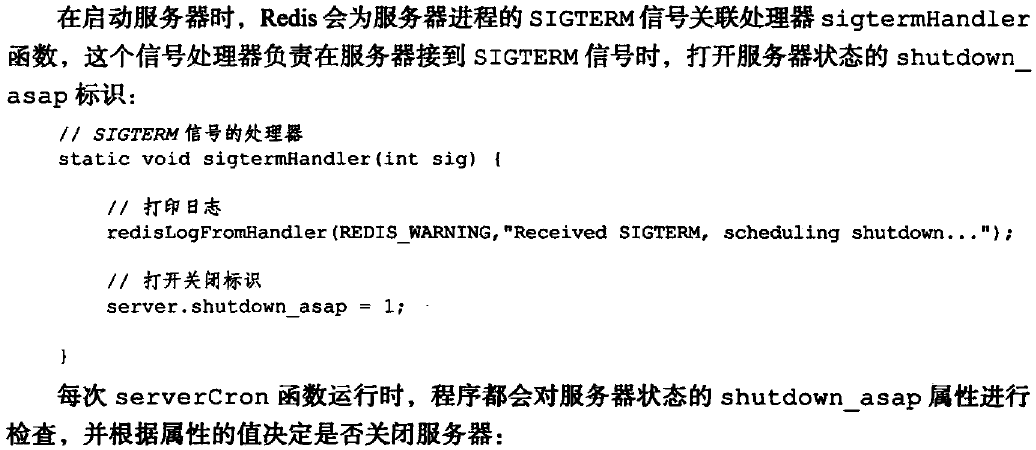
//…

Size\_t stat\_peak\_memory; //已使用内存峰值

//…

}

### 处理SIGTERM信号



Struct redisServer{

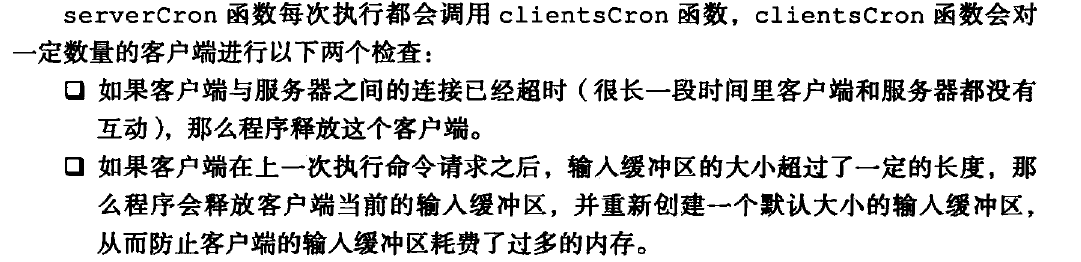
//…

Int shutdown\_asap; //1为关，0为不做动作

//…

}

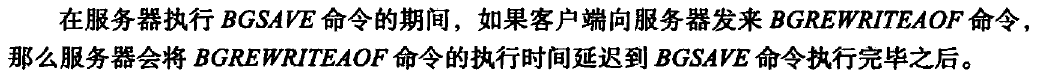
### 管理客户端资源



### 管理数据库资源

调用databaseCron函数，对服务器中一部分数据库进行检查，删除其中的过期键，并在有需要时对字典进行收缩操作。

### 执行被延迟的BGREWRITEAOF



Struct redisServer{

//…

Int aof\_rewrite\_scheduled; //如果为1则表示被延迟了

//…

}

### 检查持久化操作的运行状态

### 将AOF缓冲区中的内容写入AOF文件

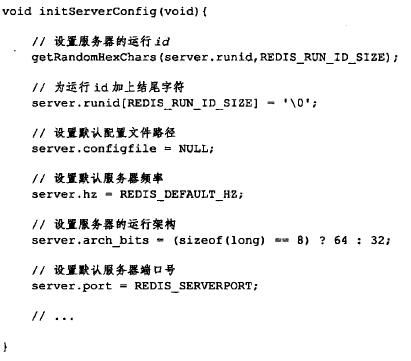
### 关闭异步客户端

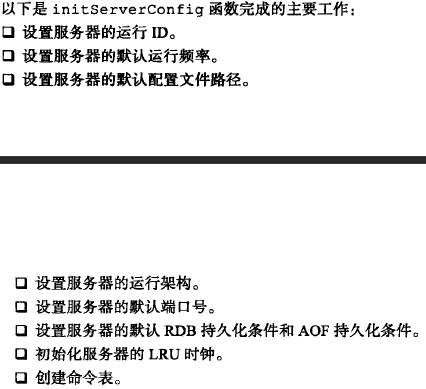
### 增加cronloops计数器的值

## 初始化服务器

### 创建redisServer实例

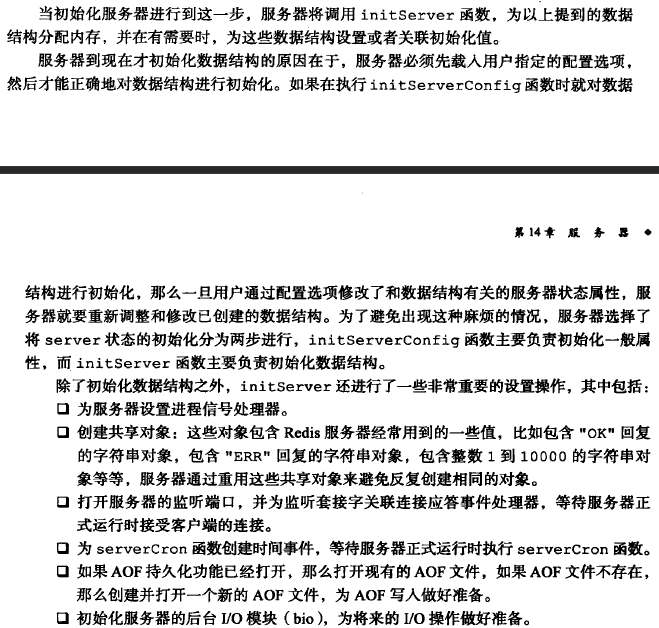
初始化服务器第一步就是创建一个redisServer类型的实例变量，并为各个属性设置默认值。该操作由initServerConfig函数完成。





### 载入配置选项

### 初始化服务器数据结构（为什么现在才初始化数据结构）



### 还原数据库状态

### 执行事件循环