Memcached&&Redis

**Memcached介绍和安装**

Memcache(内存，缓存) ： 是一个高性能的分布式的内存对象缓存系统。通过在内存里维护一个巨大的hash表。（key=value）

|  |  |
| --- | --- |
| Hash表 | |
| key（键） | value（值） |
| var1 | "123456" |
| var2 | array('ab','cd') |
| var3 | object值 |



注解：-d:以守护进程的方式开启

-l:监听哪些服务器地址，默认本地 -l%允许远程开启

-m:分配多大的内存

-p:监听的端口号：11211

**安装：**

**[root@localhost tmp]# yum install memcached**

**[root@localhost tmp]# main memcached 查询memcached相关的命令**

**任何参数都不带启动：**

**[root@localhost ~]# memcached**

can't run as root without the -u switch

说明：如果没有带 -u root 的话就会报：  
can't run as root without the -u switch  
解决：带-u root就行！

**[root@localhost ~]# sudo memcached -uroot**

**查看服务是否已经开启：(另外开启一个终端输入命令)**

**[root@localhost ~]# ps -ef | grep memcached**

**root 6340 2049 0 08:23 pts/0 00:00:00 sudo memcached -uroot**

**root 6341 6340 0 08:23 pts/0 00:00:00 memcached -uroot**

**root 6349 5923 0 08:24 pts/1 00:00:00 grep --color=auto memcached**

**服务已经成功开启！！！！！！**

补充讲解：

**在linux一般使用netstat 来查看系统端口使用情况步。**  
  
      netstat命令是一个监控TCP/IP网络的非常有用的工具，它可以显示路由表、实际的网络连接以及每一个网络接口设备的  
  
  
      netstat命令的功能是显示网络连接、路由表和网络接口信息，可以让用户得知目前都有哪些网络连接正在运作。  
  
  
      该命令的一般格式为：  
  
  
      netstat [选项]  
  
  
      命令中各选项的含义如下：  
  
  
      -a 显示所有socket，包括正在监听的。  
  
  
      -c 每隔1秒就重新显示一遍，直到用户中断它。  
  
  
      -i 显示所有网络接口的信息，格式同“ifconfig -e”。  
  
  
      -n 以网络IP地址代替名称，显示出网络连接情形。  
  
  
      -r 显示核心路由表，格式同“route -e”。  
  
  
      -t 显示TCP协议的连接情况。  
  
  
      -u 显示UDP协议的连接情况。  
  
  
      -v 显示正在进行的工作。

**执行命令：**

**[root@localhost ~]# ps -ef | grep memcached**

tcp 0 0 0.0.0.0:11211 0.0.0.0:\* LISTEN 6341/memcached

tcp6 0 0 :::11211 :::\* LISTEN 6341/memcached

udp 0 0 0.0.0.0:11211 0.0.0.0:\* 6341/memcached

udp6 0 0 :::11211 :::\* 6341/memcached

绿色的代表的就是端口号。

**使用telnet命令判断某个网络端口是不是通的：**

**[root@localhost ~]# telnet 127.0.0.1 80**

-bash: telnet: command not found

[root@localhost ~]# yum install telnet.\* #安装telnet服务

**[root@localhost ~]# telnet 127.0.0.1 80**

Trying 127.0.0.1...

telnet: connect to address 127.0.0.1: Connection refused (此端口被关闭的)

[root@localhost ~]# service nginx start #开启80端口服务

**[root@localhost ~]# telnet 127.0.0.1 80**

Trying 127.0.0.1...

Connected to 127.0.0.1. #此时80端口就是通的

执行qui退出连接

**quit**

Connection closed by foreign host.

**memcached的增删改查**

1. **连接端口，进入memcached的命令终端**

**[root@localhost ~]# telnet 127.0.0.1 11211**

**Window平台也是支持的：在xshell中不连接linux系统：{**

**[c:\~]$ telnet 192.168.254.131 11211 (不连接linux系统的情况下)**

Connecting to 192.168.254.131:11211...

Connection established.

To escape to local shell, press 'Ctrl+Alt+]'.

ERROR

set hello 0 60 10

hello word

STORED

get hello

VALUE hello 0 10

hello word

END

delete hello

DELETED

**}**

**2.增删改查（连接linu系统）**

**set a 0 60 5 ##0:标志位 60: 存储时间 5：数据大小（大数据）**

**hello**

**STORED**

**get a**

**VALUE a 0 5**

**hello**

**END**

**[root@localhost ~]# telnet -d 127.0.0.1 11211 (以后台守护进程启动)**

**3.增删改查**

**set hello 0 60 10**

**hello word**

**STORED**

**get hello**

**VALUE hello 0 10**

**hello word**

**END**

**delete hello**

**DELETED**

**Memcached中set和add的区别：**

|  |  |  |
| --- | --- | --- |
| 方法 | 当key存在 | 当key不存在 |
| add | false | true |
| replace | 替换（true） | false |
| set | 替换（true） | true |

memcached的操作：

|  |  |  |
| --- | --- | --- |
| Command | Description | Example |
| get | Reads a value | get mykey |
| set | Set a key unconditionally | set mykey 0 60 5 |
| add | Add a new key | add newkey 0 60 5 |
| replace | Overwrite existing key | replace key 0 60 5 |
| append | Append data to existing key | append key 0 60 15 |
| prepend | Prepend data to existing key | prepend key 0 60 15 |
| incr | Increments numerical key value by given number | incr mykey 2 |
| decr | Decrements numerical key value by given number | decr mykey 5 |
| delete | Deletes an existing key | delete mykey |
| flush\_all | Invalidate specific items immediately | flush\_all |
| Invalidate all items in n seconds | flush\_all 900 |
| stats | Prints general statistics | stats |
| Prints memory statistics | stats slabs |
| Prints memory statistics | stats malloc |
| Print higher level allocation statistics | stats items |
|  | stats detail |
|  | stats sizes |
| Resets statistics | stats reset |
| version | Prints server version. | version |
| verbosity | Increases log level | verbosity |
| quit | Terminate telnet session | quit |

**杀死进程，结束服务**

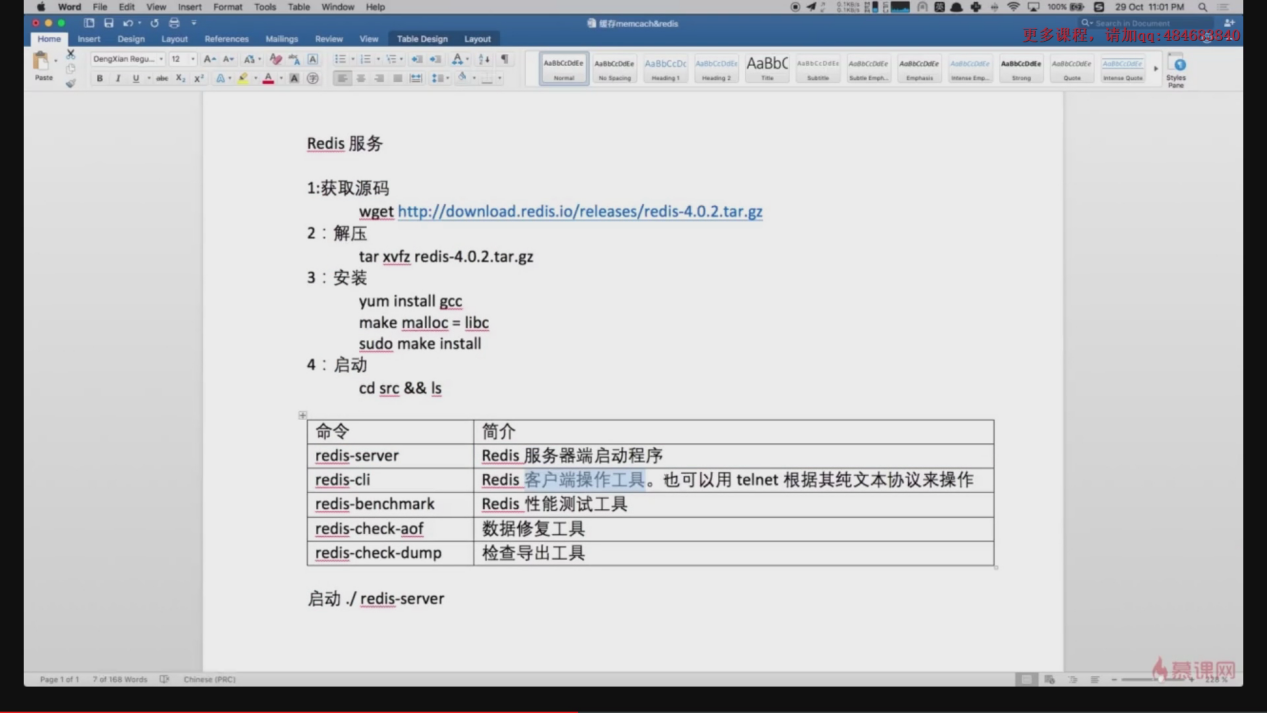
**[root@localhost ~]# ps -ef | grep mem**

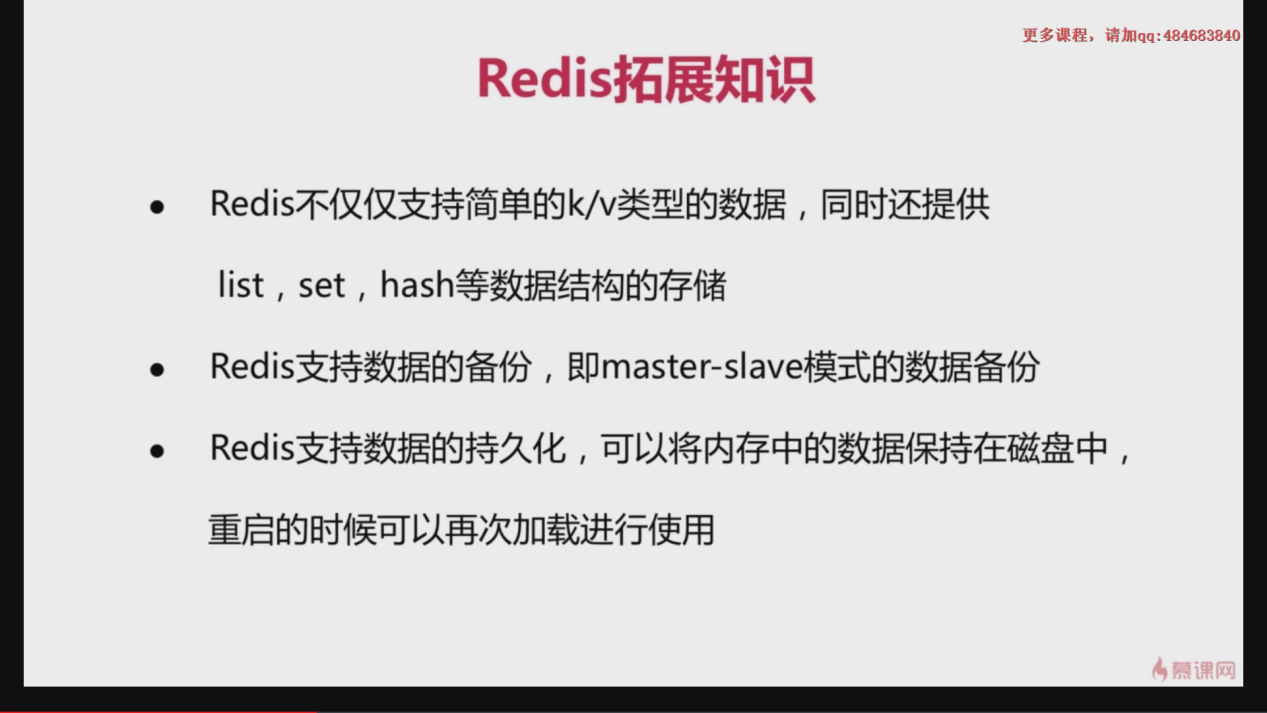
root 1152 1 0 20:40 ? 00:00:00 memcached -d -uroot

root 1159 1078 0 20:40 pts/0 00:00:00 grep --color=auto mem

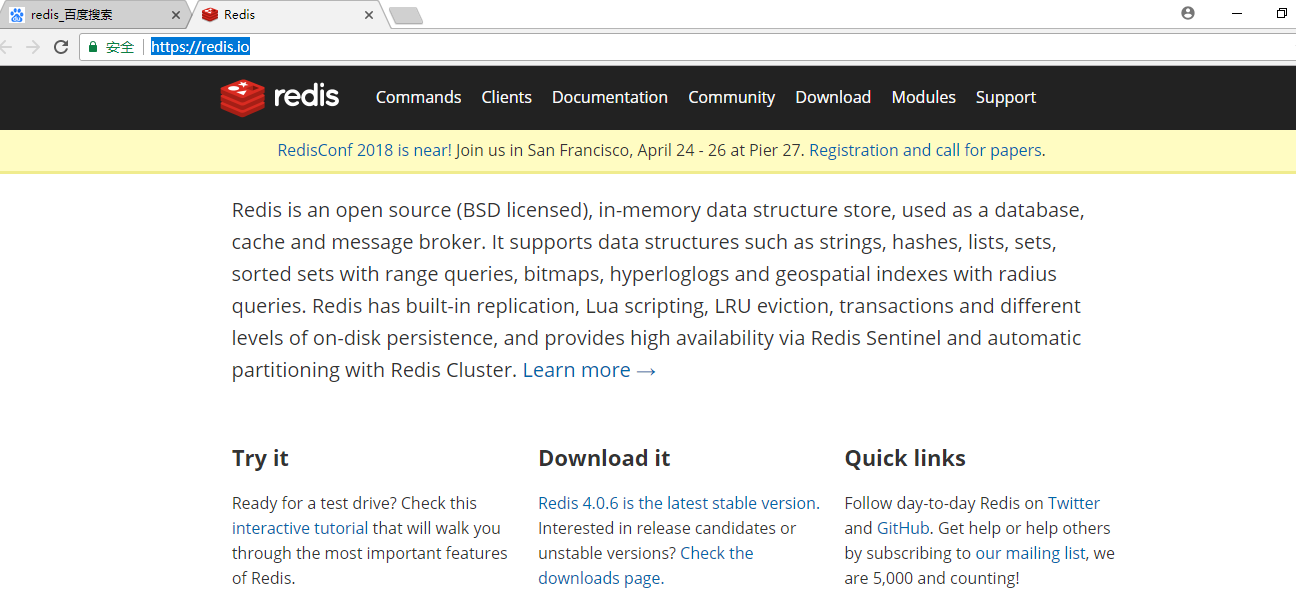
**[root@localhost ~]# kill 1152**







**Redis安装（源码安装）**



复制链接：

1. **[root@localhost tmp]# wget <http://download.redis.io/releases/redis-4.0.6.tar.gz> ##下载redis源码**
2. [root@localhost tmp]# tar -xzvf redis-4.0.6.tar.gz ##解压包

像这种下载源码编译安装的，可以到解压文件中查看软件的信息，一般是在解压文件中的README.md文件中

1. **编译（make）**

**[root@localhost redis-4.0.6]# make**

报错：

make[3]: gcc: Command not found （错误）

/bin/sh: cc: command not found

解决：

安装gcc命令：[root@localhost redis-4.0.6]# yum install gcc

**[root@localhost redis-4.0.6]# make ##再次执行**

报错：

fatal error: jemalloc/jemalloc.h: No such file or directory

解决：

[root@localhost redis-4.0.6]# make MALLOC=libc

1. **安装**

**[root@localhost redis-4.0.6]# make install**

启动：

cd src&&ls

**[root@localhost redis-4.0.6]# cd src**

**[root@localhost redis-4.0.6]# ls**

**[root@localhost src]# ./redis-server ##在此目录下进行启动**

**\_.\_**

**\_.-``\_\_ ''-.\_**

**\_.-`` `. `\_. ''-.\_ Redis 4.0.6 (00000000/0) 64 bit**

**.-`` .-```. ```\/ \_.,\_ ''-.\_**

**( ' , .-` | `, ) Running in standalone mode**

**|`-.\_`-...-` \_\_...-.``-.\_|'` \_.-'| Port: 6379** ##**服务端口号**

**| `-.\_ `.\_ / \_.-' | PID: 2247 ##进程ID**

**`-.\_ `-.\_ `-./ \_.-' \_.-'**

**|`-.\_`-.\_ `-.\_\_.-' \_.-'\_.-'|**

**| `-.\_`-.\_ \_.-'\_.-' | http://redis.io**

**`-.\_ `-.\_`-.\_\_.-'\_.-' \_.-'**

**|`-.\_`-.\_ `-.\_\_.-' \_.-'\_.-'|**

**| `-.\_`-.\_ \_.-'\_.-' |**

**`-.\_ `-.\_`-.\_\_.-'\_.-' \_.-'**

**`-.\_ `-.\_\_.-' \_.-'**

**`-.\_ \_.-'**

**`-.\_\_.-'**

**在另外一个终端执行以下命令：**

**[root@localhost ~]# netstat -anpl | grep redis**

tcp 0 0 0.0.0.0:6379 0.0.0.0:\* LISTEN 2247/./redis-server

tcp6 0 0 :::6379 :::\* LISTEN 2247/./redis-se

**6379：该服务的端口号**

**2247： 该服务的进程**

**使用杀死进程的方式结束服务：**

**[root@localhost ~]# netstat -anpl | grep redis** ##**这个命令很好用**

tcp 0 0 0.0.0.0:6379 0.0.0.0:\* LISTEN 2247/./redis-server

tcp6 0 0 :::6379 :::\* LISTEN 2247/./redis-server

**[root@localhost ~]# kill 2247**

**[root@localhost ~]# netstat -anpl | grep redis**

无显示，进程杀死，redis服务停止

**《------------------------------------------------------------------------------》**

**Redis基本操作：（在/tmp/redis-4.0.6/src目录下执行命令）**

**在一个终端打开服务，一个终端进行基本操作：**

**[root@localhost src]# ./redis-server ##一个终端开启服务**

**[root@localhost src]# ./redis-cli ##在另外一个终端开启客户端进行key-value编辑）**

**127.0.0.1:6379> set hello word**

**OK**

**127.0.0.1:6379> get hello**

**"word"**

**127.0.0.1:6379> del hello**

**(integer) 1**

**127.0.0.1:6379> get hello**

**(nil)**

**要想和memcached一样进行看客户端连接（使用telnet进行连接）：**

**1.更改参数：**

**[root@localhost src]# ./redis-cli**

**127.0.0.1:6379> CONFIG SET protected-mode no**

**OK**

1. **在xshell不连接linux的情况下操作：**

**telnet 192.168.254.131 6379**

Connecting to 192.168.254.131:6379...

Connection established.

To escape to local shell, press 'Ctrl+Alt+]'.

连接成功

**-ERR unknown command ''**

**set hello word**

**+OK**

**get hello**

**$4**

**word**

**一样可以执行key-value操作（此种方式可以直接在连接的终端下操作（连接后Enter一下），不需要开启第二个终端操作）**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*telnet是个很神奇的协议**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*redis支持更多的数据结构，memcached只是支持hash

**对此的学习网址： <http://www.runoob.com/>redis/redis-commands.html**