Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。

redis提供五种数据类型：[string](http://baike.baidu.com/view/1005388.htm)，hash，list，set及zset(sorted set)。

Redis C接口hiredis：

基本函数如下：

redisContext \*redisConnect(const char \*ip,int port);

此函数用来连接数据库，使用ip地址和port端口(一般为6379)

Void redisCommand(redisContext \*c,const char \*format,…);

函数执行命令，如同MySQL中的SQL语句

void freeReplyObject(void \*reply);

释放redisCommand执行后返回redisReply所占的内存

void redisFree(redisContext \*c);

释放redisConnect()所产生的连接

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <stddef.h>
4. #include <stdarg.h>
5. #include <string.h>
6. #include <assert.h>
7. #include <hiredis/hiredis.h>
9. **void** doTest()
10. {
11. //redis默认监听端口为6387 可以再配置文件中修改
12. redisContext\* c = redisConnect("127.0.0.1", 6379);
13. **if** ( c->err)
14. {
15. redisFree(c);
16. printf("Connect to redisServer faile\n");
17. **return** ;
18. }
19. printf("Connect to redisServer Success\n");
21. **const** **char**\* command1 = "set stest1 value1";
22. redisReply\* r = (redisReply\*)redisCommand(c, command1);
24. **if**( NULL == r)
25. {
26. printf("Execut command1 failure\n");
27. redisFree(c);
28. **return**;
29. }
30. **if**( !(r->type == REDIS\_REPLY\_STATUS && strcasecmp(r->str,"OK")==0))
31. {
32. printf("Failed to execute command[%s]\n",command1);
33. freeReplyObject(r);
34. redisFree(c);
35. **return**;
36. }
37. freeReplyObject(r);
38. printf("Succeed to execute command[%s]\n", command1);
40. **const** **char**\* command2 = "strlen stest1";
41. r = (redisReply\*)redisCommand(c, command2);
42. **if** ( r->type != REDIS\_REPLY\_INTEGER)
43. {
44. printf("Failed to execute command[%s]\n",command2);
45. freeReplyObject(r);
46. redisFree(c);
47. **return**;
48. }
49. **int** length =  r->integer;
50. freeReplyObject(r);
51. printf("The length of 'stest1' is %d.\n", length);
52. printf("Succeed to execute command[%s]\n", command2);

55. **const** **char**\* command3 = "get stest1";
56. r = (redisReply\*)redisCommand(c, command3);
57. **if** ( r->type != REDIS\_REPLY\_STRING)
58. {
59. printf("Failed to execute command[%s]\n",command3);
60. freeReplyObject(r);
61. redisFree(c);
62. **return**;
63. }
64. printf("The value of 'stest1' is %s\n", r->str);
65. freeReplyObject(r);
66. printf("Succeed to execute command[%s]\n", command3);
68. **const** **char**\* command4 = "get stest2";
69. r = (redisReply\*)redisCommand(c, command4);
70. **if** ( r->type != REDIS\_REPLY\_NIL)
71. {
72. printf("Failed to execute command[%s]\n",command4);
73. freeReplyObject(r);
74. redisFree(c);
75. **return**;
76. }
77. freeReplyObject(r);
78. printf("Succeed to execute command[%s]\n", command4);

81. redisFree(c);
83. }
85. **int** main()
86. {
87. doTest();
88. **return** 0;