WSADATA wsaData;

WSAStartup(MAKEWORD(2,1), &wsaData);

redisContext \*c;

const char \*hostname = "127.0.0.1";

int port =6379;

timeval tv ; // 1.5 seconds

tv.tv\_sec = 1;

tv.tv\_usec =500000;

c = redisConnectWithTimeout(hostname, port, tv);

if (c == NULL || c->err) {

if (c) {

printf("Connection error: %s\n", c->errstr);

redisFree(c);

}

else {

printf("Connection error: can't allocate redis context\n");

}

}

//自定义getCacheValue 命令

int getCacheValue(redisContext\* c,string &key,float fea[])

if (c == NULL)

{

return -3;

}

string command = "get " + key;

redisReply \*r;

r = (redisReply \*)redisCommand(c, command.c\_str());

if (r->type == REDIS\_REPLY\_NIL)

{

printf("Failed to execute command[%s]\n", command.c\_str());

freeReplyObject(r);

// redisFree(c);

return -1;

}

//判断返回的是否是字符串类型

if (r->type != REDIS\_REPLY\_STRING)

{

printf("Failed to execute command[%s]\n", command.c\_str());

freeReplyObject(r);

// redisFree(c);

return -2;

}

printf("The value of 'stest1' is %s\n", r->str);

memcpy(reinterpret\_cast<char\*>(fea), r->str, r->len);//这里要先判断r->len/sizeof(float)==fea.size()

freeReplyObject(r);

//redisFree(c);

printf("Succeed to execute command[%s]\n", command.c\_str());

string file\_key = "faceroom:R01:datamap:\*";

string command = "KEYS " + file\_key;//注意KEYS后面的空格

redisReply \*file\_r;

file\_r = (redisReply \*)redisCommand(c, command.c\_str());

vector<vector < float> > feas;

//遍历KEY值读取所有的fea值

for (int i = 0; i < file\_r->elements;i++)

{

redisReply\* childReply = file\_r->element[i];

string child\_key = childReply->str;

string child\_command = "get " + child\_key;

redisReply\* child\_r;

child\_r = (redisReply \*)redisCommand(c, child\_command.c\_str());

vector <float> fea(1024);

fea.resize(child\_r->len);

memcpy(reinterpret\_cast<char\*>(&fea[0]), child\_r->str, child\_r->len);//这里要先判断r->len/sizeof(float)==fea.size()

feas.push\_back(fea);

}