1创建并传递zmq\_msg\_t对象，而非数据块

2要读取消息，可使用zmq\_msg\_init()来创建一个空消息，然后传递到zmq\_msg\_recv()

3要把新数据写入消息，则使用zmq\_msg\_init\_size()来创建消息，并在同一时间分配某个大小的数据块。

char \* recv\_str = s\_recv(void \*socket);

....

free(recv\_str);

int s\_send(void \*socket,char \*string)

if(s\_send())

free(string);

如果你想多次发送相同的消息，可以创建第二个消息，并使用zmq\_msg\_init()初始化它，然后使用zmq\_msg\_copy() 来创建第一个消息的副本。这不复制数据，而是引用它，你就可以将消息发送两次（或者更多，如果你创建多个副本），消息将仅在最后一个副本被发送或者关闭时最终被销毁//zmq 云page40

zmq\_msg\_t task;

Zmq\_msg\_init(&task);

If(zmq\_msg\_recv(&task,receiver,ZMQ\_DONTWAIT) != -1){

// 处理任务

}

zmq\_msg\_close(&task)

/////////////zmq 云page42

论 zmq\_msg\_close() 重要性 /////////page43

一个客户端套接字可以连接到多个服务器套机字

static char \*s\_recv(void \*socket){

zmq\_msg\_t msg;

zmq\_msg\_init(&msg);

int size = zmq\_msg\_recv(&msg,socket,0);

if(size == -1)

Return NULL;

char \*string = malloc(size + 1);

memcpy(string,zmq\_msg\_data(&msg),size);

Zmq\_msg\_close(&msg);

return(stirng);

}