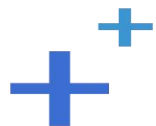


欢迎大家来到第五阶段课程

《分布式流媒体》实训项目



TNV DAY09

预习课

预习
内容

客户机 (7)

客户机 (7)



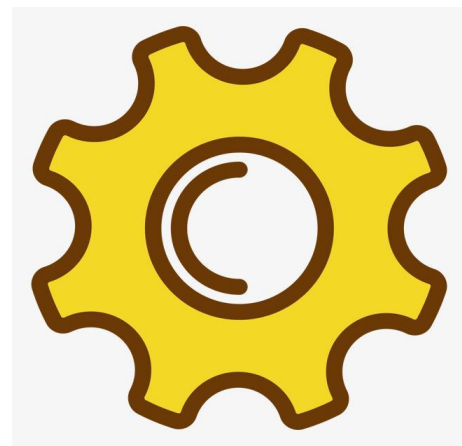
连接池类(pool_c)的属性和构造

- 成员变量
 - 连接超时: m_ctimeout
 - 读写超时: m_rtimeout
 - 空闲超时: m_itimeout
- 构造函数
 - 调用基类的构造函数
 - 初始化成员变量



连接池类(pool_c)的一般方法

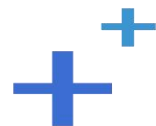
- 设置超时: timeouts
 - 设置连接超时
 - 设置读写超时
 - 设置空闲超时
- 获取连接: peek
 - 空闲检测
 - 调用基类的同名方法



连接池类(pool_c)的基类方法

- 创建连接: create_connect
 - 实例化连接类(conn_c)的对象

```
// 创建连接
acl::connect_client* pool_c::create_connect(void) {
    return new conn_c(addr_, m_ctimeout, m_rtimeout);
}
```



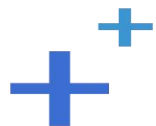
连接池管理器类(mngr_c)的基类方法

- 创建连接池: create_pool
 - 实例化连接池类(pool_c)的对象

```
// 创建连接池
acl::connect_pool* mngr_c::create_pool(
    char const* destaddr, size_t count, size_t index) {
    return new pool_c(destaddr, count, index);
}
```



附录：程序清单



TNV/src/05_client/03_pool.h

```
// 客户机
// 声明连接池类
//
#pragma once

#include <lib_acl.hpp>
//
// 连接池类
//
class pool_c: public acl::connect_pool {
public:
    // 构造函数
    pool_c(char const* destaddr, int count, size_t index = 0);

    // 设置超时
```



TNV/src/05_client/03_pool.h

```
void timeouts(int ctimeout = 30, int rtimeout = 60,  
             int itimeout = 90);  
// 获取连接  
acl::connect_client* peek(void);
```

protected:

```
// 创建连接  
acl::connect_client* create_connect(void);
```

private:

```
int m_ctimeout; // 连接超时  
int m_rtimeout; // 读写超时  
int m_itimeout; // 空闲超时
```

```
};
```



TNV/src/05_client/04_pool.cpp

```
// 客户机
// 实现连接池类
//
#include "01_conn.h"
#include "03_pool.h"

// 构造函数
pool_c::pool_c(char const* destaddr, int count, size_t index /* = 0 */):
    connect_pool(destaddr, count, index),
    m_ctimeout(30), m_rtimeout(60), m_itymeout(90) {

// 设置超时
void pool_c::timeouts(int ctimeout /* = 30 */, int rtimeout /* = 60 */,
    int itimeout /* = 90 */) {
```



TNV/src/05_client/04_pool.cpp

```
    m_ctimeout = ctimeout;
    m_rtimeout = rtimeout;
    m_itymeout = itimeout;
}

// 获取连接
acl::connect_client* pool_c::peek(void) {
    connect_pool::check_idle(m_itymeout);
    return connect_pool::peek();
}

// 创建连接
acl::connect_client* pool_c::create_connect(void) {
    return new conn_c(addr_, m_ctimeout, m_rtimeout);
}
```



TNV/src/05_client/05_mngr.h

```
// 客户机
// 声明连接池管理器类
//
#pragma once

#include <lib_acl.hpp>
//
// 连接池管理器类
//
class mngr_c: public acl::connect_manager {
protected:
    // 创建连接池
    acl::connect_pool* create_pool(
        char const* destaddr, size_t count, size_t index);
};
```



TNV/src/05_client/06_mngr.cpp

```
// 客户机
// 实现连接池管理器类
//
#include "03_pool.h"
#include "05_mngr.h"

// 创建连接池
acl::connect_pool* mngr_c::create_pool(
    char const* destaddr, size_t count, size_t index) {
    return new pool_c(destaddr, count, index);
}
```



直播课见