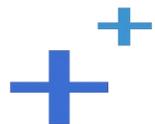


# 欢迎大家来到第五阶段课程

## 《分布式流媒体》实训项目



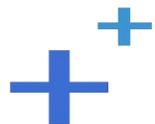
# TNV DAY14

预习课

预习  
内容

HTTP服务器 (5)

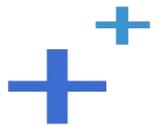
# HTTP服务器 (5)



# 服务器类(server\_c)

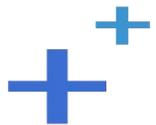
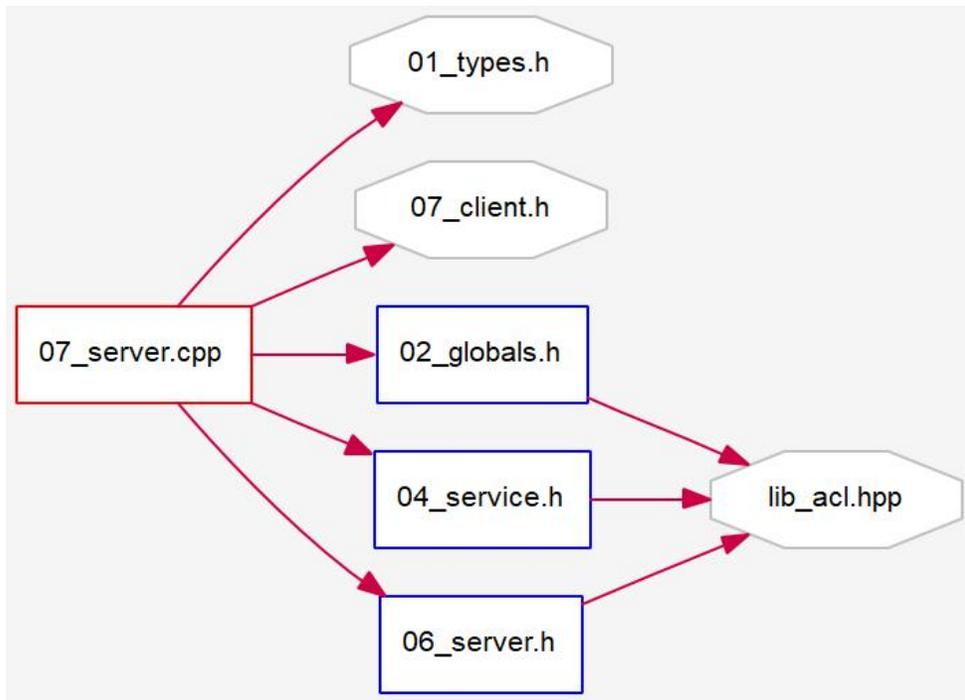
server\_c

```
+ thread_on_accept (conn: int *) bool  
+ thread_on_close (conn: int *) void  
+ thread_on_read (conn: int *) bool  
+ thread_on_timeout (conn: int *) bool  
# proc_exit_timer (nclients: size_t , nthreads: size_t ) bool  
# proc_on_exit () void  
# proc_on_init () void
```

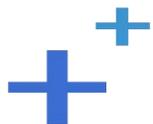


# 服务器类(server\_c)的属性

- 成员变量
  - Redis集群: m\_redis



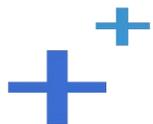
# 附录：程序清单



# TNV/src/06\_http/06\_server.h

```
// HTTP服务器
// 声明服务器类
//
#pragma once

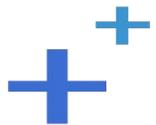
#include <lib_acl.hpp>
//
// 服务器类
//
class server_c: public acl::master_threads {
protected:
    // 进程切换用户后被调用
    void proc_on_init(void);
    // 子进程意图退出时被调用
    // 返回true, 子进程立即退出, 否则
```



# TNV/src/06\_http/06\_server.h

```
// 若配置项ioctl_quick_abort非0, 子进程立即退出, 否则
// 待所有客户机连接都关闭后, 子进程再退出
bool proc_exit_timer(size_t nclients, size_t nthreads);
// 进程退出前被调用
void proc_on_exit(void);

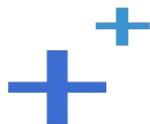
// 线程获得连接时被调用
// 返回true, 连接将被用于后续通信, 否则
// 函数返回后即关闭连接
bool thread_on_accept(acl::socket_stream* conn);
// 与线程绑定的连接可读时被调用
// 返回true, 保持长连接, 否则
// 函数返回后即关闭连接
bool thread_on_read(acl::socket_stream* conn);
// 线程读写连接超时时被调用
```



# TNV/src/06\_http/06\_server.h

```
// 返回true, 继续等待下一次读写, 否则
// 函数返回后即关闭连接
bool thread_on_timeout(acl::socket_stream* conn);
// 与线程绑定的连接关闭时被调用
void thread_on_close(acl::socket_stream* conn);

private:
    acl::redis_client_cluster* m_redis; // Redis集群
};
```



# 直播课见