

《分布式流媒体》实训项目

C/C++教学体系

TNV DAY02

直播课



目录

公共模块详细设计

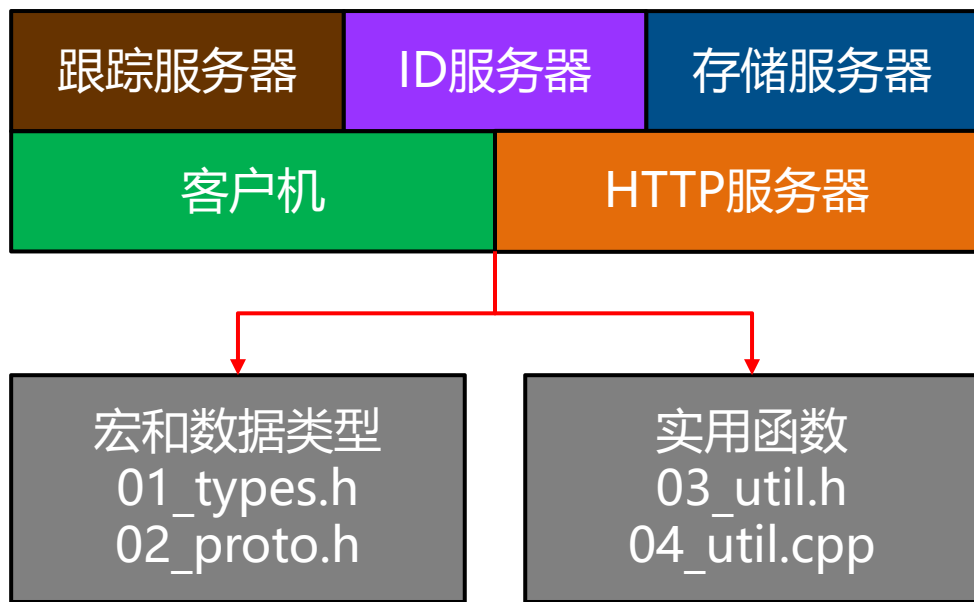
通用宏和数据类型

公共模块详细设计



组织结构

- 01_common



开发计划

序号	内容	文档/代码	时间
6	定义所有模块都会用到的宏和数据类型	01_common/01_types.h	6小时
7	定义与报文规约有关的宏和数据类型	01_common/02_proto.h	
8	声明几个实用函数	01_common/03_util.h	
9	定义几个实用函数	01_common/04_util.cpp	



通用宏和数据类型



函数返回值和缓存

- 函数返回值
 - 成功: OK
 - 本地错误: ERROR
 - 套接字通信错误: SOCKET_ERROR
 - 服务器状态异常: STATUS_ERROR
- 缓存
 - 跟踪服务器Redis前缀: TRACKER_REDIS_PREFIX
 - 存储服务器Redis前缀: STORAGE_REDIS_PREFIX



存储服务器状态和描述

- 存储服务器状态
 - 离线: STORAGE_STATUS_OFFLINE
 - 在线: STORAGE_STATUS_ONLINE
 - 活动: STORAGE_STATUS_ACTIVE
- 存储服务器描述
 - 版本最大字符数: STORAGE_VERSION_MAX
 - 组名最大字符数: STORAGE_GROUPNAME_MAX
 - 主机名最大字符数: STORAGE_HOSTNAME_MAX
 - 地址最大字符数: STORAGE_ADDR_MAX



存储服务器加入、信息、ID键值对和文件缓冲区

- 存储服务器加入
 - 版本、组名、主机名、端口号、启动时间、加入时间
- 存储服务器信息
 - 版本、主机名、IP地址、端口号、启动时间、加入时间、心跳时间、状态
- ID键值对
 - 键最大字符数：ID_KEY_MAX
 - ID键值对：键、值、偏移
- 存储服务器读写磁盘文件缓冲区
 - 接收写入缓冲区字节数：STORAGE_RCVWR_SIZE
 - 读取发送缓冲区字节数：STORAGE_RDSND_SIZE



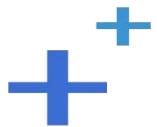
附录：程序清单



TNV/src/01_common/01_types.h

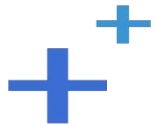
```
// 公共模块
// 定义所有模块都会用到的宏和数据类型
//
#pragma once

#include <netinet/in.h>
//
// 函数返回值
//
#define OK          0 // 成功
#define ERROR      -1 // 本地错误
#define SOCKET_ERROR -2 // 套接字通信错误
#define STATUS_ERROR -3 // 服务器状态异常
//
// 缓存
```



TNV/src/01_common/01_types.h

```
//  
#define TRACKER_REDIS_PREFIX "tracker" // 跟踪服务器Redis前缀  
#define STORAGE_REDIS_PREFIX "storage" // 存储服务器Redis前缀  
//  
// 存储服务器状态  
//  
typedef enum storage_status {  
    STORAGE_STATUS_OFFLINE, // 存储服务器离线  
    STORAGE_STATUS_ONLINE, // 存储服务器在线  
    STORAGE_STATUS_ACTIVE // 存储服务器活动  
} storage_status_t; // 存储服务器状态  
//  
// 存储服务器加入和信息  
//  
#define STORAGE_VERSION_MAX 6 // 版本最大字符数
```



TNV/src/01_common/01_types.h

```
#define STORAGE_GROUPNAME_MAX 16 // 组名最大字符数
#define STORAGE_HOSTNAME_MAX 128 // 主机名最大字符数
#define STORAGE_ADDR_MAX 16 // 地址最大字符数

typedef struct storage_join {
    char    sj_version[STORAGE_VERSION_MAX+1]; // 版本
    char    sj_groupname[STORAGE_GROUPNAME_MAX+1]; // 组名
    char    sj_hostname[STORAGE_HOSTNAME_MAX+1]; // 主机名
    in_port_t sj_port; // 端口号
    time_t  sj_stime; // 启动时间
    time_t  sj_jtime; // 加入时间
} storage_join_t; // 存储服务器加入

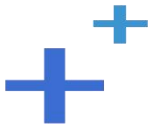
typedef struct storage_info {
    char    si_version[STORAGE_VERSION_MAX+1]; // 版本
```



TNV/src/01_common/01_types.h

```
char          si_hostname[STORAGE_HOSTNAME_MAX+1]; // 主机名
char          si_addr[STORAGE_ADDR_MAX+1];         // IP地址
in_port_t     si_port;                             // 端口号
time_t        si_stime;                             // 启动时间
time_t        si_jtime;                             // 加入时间
time_t        si_btime;                             // 心跳时间
storage_status_t si_status;                         // 状态
}
storage_info_t; // 存储服务器信息
//
// ID键值对
//
#define ID_KEY_MAX 64 // 键最大字符数

typedef struct id_pair {
    char id_key[ID_KEY_MAX+1]; // 键
```



TNV/src/01_common/01_types.h

```
    long id_value;           // 值
    int  id_offset;         // 偏移
}    id_pair_t;            // ID键值对
//
// 存储服务器读写磁盘文件缓冲区
//
#define STORAGE_RCVWR_SIZE (512 * 1024) // 接收写入缓冲区字节数
#define STORAGE_RDSND_SIZE (512 * 1024) // 读取发送缓冲区字节数
```



复习课见