Tedu.cn
达内教育

《分布式流媒体》实训项目

C/C++教学体系

Tedu.cn
达内教育

TNV DAY03

直播课

目录

报文宏和数据类型

实用工具

Tedu.cn
达内教育

报文宏和数据类型

一般报文和差错报文

知识讲解

- 一般报文
  - 包体长度字节数：BODYLEN_SIZE
  - 命令字节数：COMMAND_SIZE
  - 状态字节数：STATUS_SIZE
  - 包头长度：HEADLEN
- 差错报文
  - 错误号字节数：ERROR_NUMB_SIZE
  - 错误描述最大字节数(含结尾空字符)： ERROR_DESC_SIZE

包头			包体
包体长度	命令	状态	...
8	1	1	包体长度

包头			包体	
包体长度	命令	状态	错误号	错误描述
8	1	1	2	<=1024

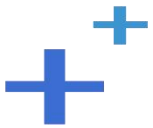
Tedu.cn
达内教育

ID报文、存储服务器加入包和心跳包

知识讲解

- ID报文
  - 应用ID最大字节数(含结尾空字符)：APPID_SIZE
  - 用户ID最大字节数(含结尾空字符)：USERID_SIZE
  - 文件ID最大字节数(含结尾空字符)：FILEID_SIZE
- 存储服务器加入包体
  - 版本、组名、主机名、端口号、启动时间、加入时间
- 存储服务器心跳包体
  - 组名、主机名

包头			包体		
包体长度	命令	状态	应用ID	用户ID	文件ID
8	1	1	16	256	128

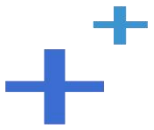


Tedu.cn
达内教育

跟踪服务器和ID服务器处理的命令

知识讲解

- 跟踪服务器处理的命令
  - 存储服务器向跟踪服务器发送加入包：CMD_TRACKER_JOIN
  - 存储服务器向跟踪服务器发送心跳包：CMD_TRACKER_BEAT
  - 客户机从跟踪服务器获取存储服务器地址列表：CMD_TRACKER_SADDRS
  - 客户机从跟踪服务器获取组列表：CMD_TRACKER_GROUPS
- ID服务器处理的命令
  - 存储服务器从ID服务器获取ID：CMD_ID_GET

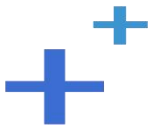


Tedu.cn
达内教育

存储服务器处理的命令和应答命令

知识讲解

- 存储服务器处理的命令
  - 客户机向存储服务器上传文件：CMD_STORAGE_UPLOAD
  - 客户机向存储服务器询问文件大小：CMD_STORAGE_FILESIZE
  - 客户机从存储服务器下载文件：CMD_STORAGE_DOWNLOAD
  - 客户机删除存储服务器上的文件：CMD_STORAGE_DELETE
- 应答命令
  - 跟踪服务器应答：CMD_TRACKER_REPLY
  - ID服务器应答：CMD_ID_REPLY
  - 存储服务器应答：CMD_STORAGE_REPLY



Tedu.cn
达内教育

实用工具

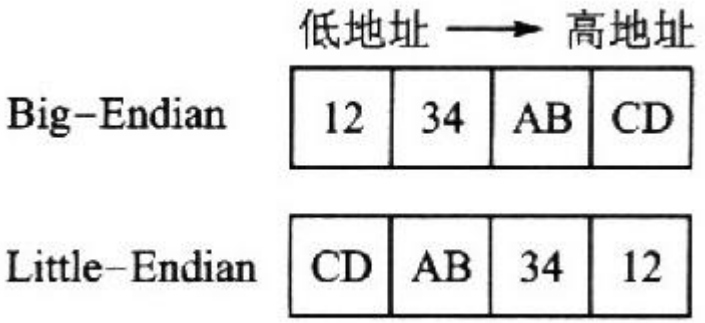
Tedu.cn
达内教育

字节序变换函数

知识讲解

- 主机序变网络(大端)序
  - 16位：ston
  - 32位：lton
  - 64位：llton
- 网络(大端)序变主机序
  - 16位：ntos
  - 32位：ntol
  - 64位：ntoll

0x1234ABCD

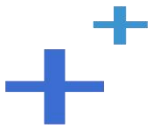


Tedu.cn
达内教育

字符串处理函数

知识讲解

- 字符串验证：valid
  - 字符串是否合法，即是否只包含26个英文字母大小写和0到9十个阿拉伯数字
- 字符串拆分：split
  - 以分号为分隔符将一个字符串拆分为多个子字符串，以字符串向量形式输出



Tedu.cn
达内教育

附录：程序清单

Tedu.cn
达内教育

TNV/src/01_common/02_proto.h

知识讲解

```
//公共模块
//定义与报文规约有关的宏和数据类型
//
#pragma once

#include "01_types.h"
//
//|包体长度|命令|状态| 包体 |
//|  8 | 1| 1|包体长度|
//
#define BODYLEN_SIZE 8 //包体长度字节数
#define COMMAND_SIZE 1 //命令字节数
#define STATUS_SIZE  1 //状态字节数
#define HEADLEN (BODYLEN_SIZE + COMMAND_SIZE + STATUS_SIZE) //包头长度
//
```

TNV/src/01_common/02_proto.h

知识讲解

```
// |包体长度|命令|状态|错误号|错误描述|
// |    8   |  1 |  1 |   2  | <=1024 |
//
#define ERROR_NUMB_SIZE     2 //错误号字节数
#define ERROR_DESC_SIZE 1024 //错误描述最大字节数(含结尾空字符)
//
// |包体长度|命令|状态|应用ID|用户ID|文件ID|
// |    8   |  1 |  1 |  16  |  256 |  128 |
//
#define APPID_SIZE    16 //应用ID最大字节数(含结尾空字符)
#define USERID_SIZE 256 //用户ID最大字节数(含结尾空字符)
#define FILEID_SIZE 128 //文件ID最大字节数(含结尾空字符)
//
//存储服务器加入包和心跳包
//
```

Tedu.cn
达内教育

TNV/src/01_common/02_proto.h

知识讲解

```
typedef struct storage_join_body {
        char sjb_version[STORAGE_VERSION_MAX+1];      //版本
        char sjb_groupname[STORAGE_GROUPNAME_MAX+1]; //组名
        char sjb_hostname[STORAGE_HOSTNAME_MAX+1];    //主机名
        char sjb_port[sizeof(in_port_t)];             //端口号
        char sjb_stime[sizeof(time_t)];               //启动时间
        char sjb_jtime[sizeof(time_t)];               //加入时间
}       storage_join_body_t; //存储服务器加入包体

typedef struct storage_beat_body {
        char sbb_groupname[STORAGE_GROUPNAME_MAX+1]; //组名
        char sbb_hostname[STORAGE_HOSTNAME_MAX+1];    //主机名
}       storage_beat_body_t; //存储服务器心跳包体
//
//命令
```

TNV/src/01_common/02_proto.h

知识讲解

```
//
#define CMD_TRACKER_JOIN    10 //存储服务器向跟踪服务器发送加入包
#define CMD_TRACKER_BEAT    11 //存储服务器向跟踪服务器发送心跳包
#define CMD_TRACKER_SADDRS  12 //客户机从跟踪服务器获取存储服务器地址列表
#define CMD_TRACKER_GROUPS  13 //客户机从跟踪服务器获取组列表

#define CMD_ID_GET 40 //存储服务器从ID服务器获取ID

#define CMD_STORAGE_UPLOAD    70 //客户机向存储服务器上传文件
#define CMD_STORAGE_FILESIZE  71 //客户机向存储服务器询问文件大小
#define CMD_STORAGE_DOWNLOAD  72 //客户机从存储服务器下载文件
#define CMD_STORAGE_DELETE    73 //客户机删除存储服务器上的文件

#define CMD_TRACKER_REPLY 100 //跟踪服务器应答
#define CMD_ID_REPLY      101 //ID服务器应答
#define CMD_STORAGE_REPLY 102 //存储服务器应答
```

Tedu.cn
达内教育

TNV/src/01_common/03_util.h

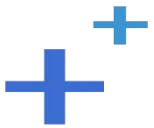
知识讲解

```
// 公共模块
// 声明几个实用函数
//
#pragma once

#include <string>
#include <vector>

// long long类型整数主机序转网络序
void llton(long long ll, char* n);
// long long类型整数网络序转主机序
long long ntoll(char const* n);

// long类型整数主机序转网络序
void lton(long l, char* n);
```



Tedu.cn
达内教育

TNV/src/01_common/03_util.h

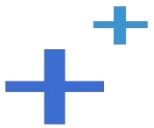
知识讲解

```
// long类型整数网络序转主机序
long ntol(char const* n);

// short类型整数主机序转网络序
void ston(short s, char* n);
// short类型整数网络序转主机序
short ntos(char const* n);

// 字符串是否合法，即是否只包含26个英文字母大小写和0到9十个阿拉伯数字
int valid(char const* str);

// 以分号为分隔符将一个字符串拆分为多个子字符串
int split(char const* str, std::vector<std::string>& substrs);
```



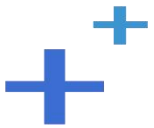
TNV/src/01_common/04_util.cpp

知识讲解

```
// 公共模块
// 定义几个实用函数
//
#include <string.h>
#include "01_types.h"
#include "03_util.h"

// long long类型整数主机序转网络序
void llton(long long ll, char* n) {
        for (size_t i = 0; i < sizeof(ll); ++i)
                n[i] = ll >> (sizeof(ll) - i - 1) * 8;
}

// long long类型整数网络序转主机序
long long ntoll(char const* n) {
```



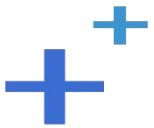
TNV/src/01_common/04_util.cpp

知识讲解

```
        long long ll = 0;
        for (size_t i = 0; i < sizeof(ll); ++i)
                ll |= (long long)(unsigned char)n[i] << (sizeof(ll) - i - 1) * 8;
        return ll;
}

// long类型整数主机序转网络序
void lton(long l, char* n) {
        for (size_t i = 0; i < sizeof(l); ++i)
                n[i] = l >> (sizeof(l) - i - 1) * 8;
}

// long类型整数网络序转主机序
long ntol(char const* n) {
        long l = 0;
```



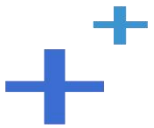
TNV/src/01_common/04_util.cpp

知识讲解

```
        for (size_t i = 0; i < sizeof(l); ++i)
                l |= (long)(unsigned char)n[i] << (sizeof(l) - i - 1) * 8;
        return l;
}

// short类型整数主机序转网络序
void ston(short s, char* n) {
        for (size_t i = 0; i < sizeof(s); ++i)
                n[i] = s >> (sizeof(s) - i - 1) * 8;
}

// short类型整数网络序转主机序
short ntos(char const* n) {
        short s = 0;
        for (size_t i = 0; i < sizeof(s); ++i)
```



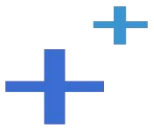
TNV/src/01_common/04_util.cpp


```
                s |= (short)(unsigned char)n[i] << (sizeof(s) - i - 1) * 8;
        return s;
}

//字符串是否合法，即是否只包含26个英文字母大小写和0到9十个阿拉伯数字
int valid(char const* str) {
        if (!str)
                return ERROR;

        size_t len = strlen(str);
        if (!len)
                return ERROR;

        for (size_t i = 0; i < len; ++i)
                if (!(('a' <= str[i] && str[i] <= 'z') ||
```



TNV/src/01_common/04_util.cpp

```
                ('A' <= str[i] && str[i] <= 'Z') ||
                ('0' <= str[i] && str[i] <= '9')))
            return ERROR;

    return OK;
}

//以分号为分隔符将一个字符串拆分为多个子字符串
int split(char const* str, std::vector<std::string>& substrs) {
    if (!str)
        return ERROR;

    size_t len = strlen(str);
    if (!len)
        return ERROR;
```

知识讲解

TNV/src/01_common/04_util.cpp

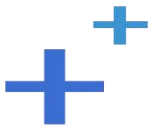
知识讲解

```
        char* buf = new char[len+1];
        strcpy(buf, str);

        char const* sep = ";";
        for (char* substr = strtok(buf, sep); substr;
                substr = strtok(NULL, sep))
                substrs.push_back(substr);

        delete[] buf;

        return OK;
}
```



Tedu.cn
达内教育

复习课见