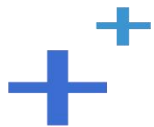


欢迎大家来到第五阶段课程

《分布式流媒体》实训项目



TNV DAY06

复习课

预习
内容

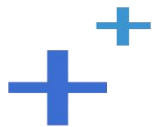
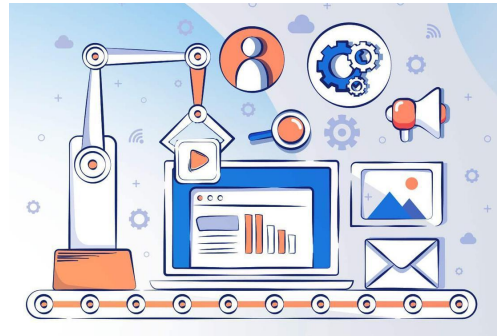
客户机 (2)

客户机 (2)



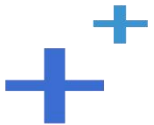
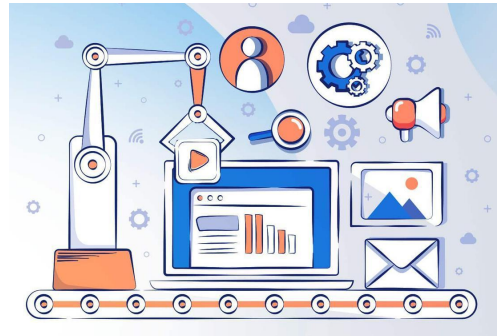
连接类(conn_c)的一级方法

- 从跟踪服务器获取存储服务器地址列表: saddrs
 - 构造请求
 - 发送请求
 - 接收包体
 - 解析包体
 - 成功
 - 输出存储服务器地址列表
 - 服务器状态异常
 - 获取错误号和错误描述
 - 释放包体
 - 返回处理结果



连接类(conn_c)的一级方法

- 从跟踪服务器获取组列表: groups
 - 构造请求
 - 发送请求
 - 接收包体
 - 解析包体
 - 成功
 - 输出组列表
 - 服务器状态异常
 - 获取错误号和错误描述
 - 释放包体
 - 返回处理结果



附录：程序清单



TNV/src/05_client/02_conn.cpp

// 从跟踪服务器获取存储服务器地址列表

```
int conn_c::saddrs(char const* appid, char const* userid,
    char const* fileid, std::string& saddrs) {
    // |包体长度|命令|状态|应用ID|用户ID|文件ID|
    // | 8 | 1 | 1 | 16 | 256 | 128 |
    // 构造请求
    long long bodylen = APPID_SIZE + USERID_SIZE + FILEID_SIZE;
    long long requlen = HEADLEN + bodylen;
    char requ[requlen];
    if (makerequ(CMD_TRACKER_SADDRS,
        appid, userid, fileid, requ) != OK)
        return ERROR;
    hton(bodylen, requ);

    if (!open())
```



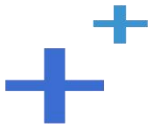
TNV/src/05_client/02_conn.cpp

```
return SOCKET_ERROR;
```

```
// 发送请求
```

```
if (m_conn->write(requ, reqlen) < 0) {  
    logger_error("write fail: %s, reqlen: %lld, to: %s",  
                acl::last_serror(), reqlen, m_conn->get_peer());  
    m_errnumb = -1;  
    m_errdesc.format("write fail: %s, reqlen: %lld, to: %s",  
                    acl::last_serror(), reqlen, m_conn->get_peer());  
    close();  
    return SOCKET_ERROR;  
}
```

```
char* body = NULL; // 包体指针
```



TNV/src/05_client/02_conn.cpp

// 接收包体

```
int result = recvbody(&body, &bodylen);
```

// 解析包体

```
if (result == OK)
```

```
    // |包体长度|命令|状态|组名|存储服务器地址列表|
```

```
    // | 8 | 1 | 1 |16+1| 包体长度-(16+1) |
```

```
    saddrs = body + STORAGE_GROUPNAME_MAX + 1;
```

```
else if (result == STATUS_ERROR) {
```

```
    // |包体长度|命令|状态|错误号|错误描述|
```

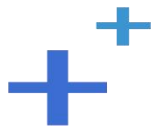
```
    // | 8 | 1 | 1 | 2 | <=1024 |
```

```
    m_errnumb = ntos(body);
```

```
    m_errdesc = bodylen > ERROR_NUMB_SIZE ?
```

```
        body + ERROR_NUMB_SIZE : "";
```

```
}
```

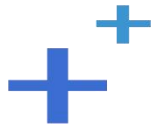


TNV/src/05_client/02_conn.cpp

```
// 释放包体
if (body) {
    free(body);
    body = NULL;
}

return result;
}

// 从跟踪服务器获取组列表
int conn_c::groups(std::string& groups) {
    // |包体长度|命令|状态|
    // | 8 | 1 | 1 |
    // 构造请求
    long long bodylen = 0;
```

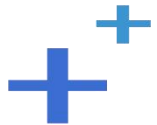


TNV/src/05_client/02_conn.cpp

```
long long reqlen = HEADLEN + bodylen;
char requ[reqlen] = {};
htonl(bodylen, requ);
requ[BODYLEN_SIZE] = CMD_TRACKER_GROUPS;
requ[BODYLEN_SIZE+COMMAND_SIZE] = 0;

if (!open())
    return SOCKET_ERROR;

// 发送请求
if (m_conn->write(requ, reqlen) < 0) {
    logger_error("write fail: %s, reqlen: %lld, to: %s",
                acl::last_serror(), reqlen, m_conn->get_peer());
    m_errnumb = -1;
    m_errdesc.format("write fail: %s, reqlen: %lld, to: %s",
```



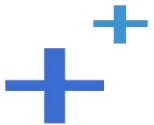
TNV/src/05_client/02_conn.cpp

```
        acl::last_serror(), reqlen, m_conn->get_peer());
    close();
    return SOCKET_ERROR;
}

char* body = NULL; // 包体指针

// 接收包体
int result = recvbody(&body, &bodylen);

// 解析包体
if (result == OK)
    // |包体长度|命令|状态|组列表|
    // | 8 | 1 | 1 |包体长度|
    groups = body;
```



TNV/src/05_client/02_conn.cpp

```
else if (result == STATUS_ERROR) {  
    // |包体长度|命令|状态|错误号|错误描述|  
    // | 8 | 1 | 1 | 2 | <=1024 |  
    m_errnumb = ntos(body);  
    m_errdesc = bodylen > ERROR_NUMB_SIZE ?  
                body + ERROR_NUMB_SIZE : "";  
}  
  
// 释放包体  
if (body) {  
    free(body);  
    body = NULL;  
}  
  
return result;
```



下节课见