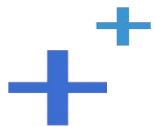


欢迎大家来到第五阶段课程

《分布式流媒体》实训项目



TNV DAY07

复习课

预习
内容

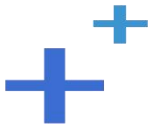
客户机 (4)

客户机 (4)



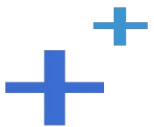
连接类(conn_c)的一级方法

- 向存储服务器询问文件大小：filesize
 - 构造请求
 - 发送请求
 - 接收包体
 - 解析包体
 - 成功
 - 输出文件大小
 - 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

- 从存储服务器下载文件：download
 - 构造请求
 - 发送请求
 - 接收包体
 - 解析包体
 - 成功
 - 输出文件数据和大小
 - 返回处理结果
 - 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

// 向存储服务询问文件大小

```
int conn_c::filesize(char const* appid, char const* userid,
    char const* fileid, long long* filesize) {
    // |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 reqlen = HEADLEN + bodylen;
    char requ[reqlen];
    if (makerequ(CMD_STORAGE_FILESIZE,
        appid, userid, fileid, requ) != OK)
        return ERROR;
    hton(bodylen, requ);

    if (!open())
```



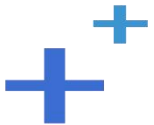
TNV/src/05_client/02_conn.cpp

```
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|文件大小|
```

```
    // | 8 | 1 | 1 | 8 |
```

```
    *filesize = ntoll(body);
```

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

```
    // |包体长度|命令|状态|错误号|错误描述|
```

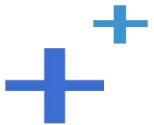
```
    // | 8 | 1 | 1 | 2 | <=1024 |
```

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

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

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

```
}
```

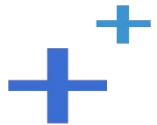


TNV/src/05_client/02_conn.cpp

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

return result;
}

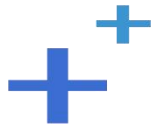
// 从存储服务器下载文件
int conn_c::download(char const* appid, char const* userid,
    char const* fileid, long long offset, long long size,
    char** filedata, long long* filesize) {
    // |包体长度|命令|状态|应用ID|用户ID|文件ID|偏移|大小|
    // | 8 | 1 | 1 | 16 | 256 | 128 | 8 | 8 |
```



TNV/src/05_client/02_conn.cpp

// 构造请求

```
long long bodylen = APPID_SIZE + USERID_SIZE + FILEID_SIZE +  
    BODYLEN_SIZE + BODYLEN_SIZE;  
long long requlen = HEADLEN + bodylen;  
char requ[requlen];  
if (makerequ(CMD_STORAGE_DOWNLOAD,  
    appid, userid, fileid, requ) != OK)  
    return ERROR;  
llton(bodylen, requ);  
llton(offset, requ + HEADLEN +  
    APPID_SIZE + USERID_SIZE + FILEID_SIZE);  
llton(size, requ + HEADLEN +  
    APPID_SIZE + USERID_SIZE + FILEID_SIZE + BODYLEN_SIZE);  
  
if (!open())
```



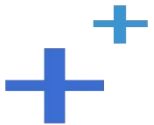
TNV/src/05_client/02_conn.cpp

```
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;  
}
```

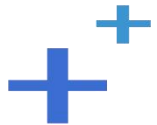
```
char* body = NULL; // 包体指针
```



TNV/src/05_client/02_conn.cpp

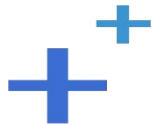
```
// 接收包体
int result = recvbody(&body, &bodylen);

// 解析包体
if (result == OK) {
    // |包体长度|命令|状态|文件内容|
    // | 8 | 1 | 1 |内容大小|
    *filedata = body;
    *filesize = bodylen;
    return result;
}
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);
}
```



TNV/src/05_client/02_conn.cpp

```
        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