Tedu.cn
达内教育

欢迎大家来到第五阶段课程

《分布式流媒体》实训项目

预习课

TNV DAY07

预习课

预习内容

客户机（3）

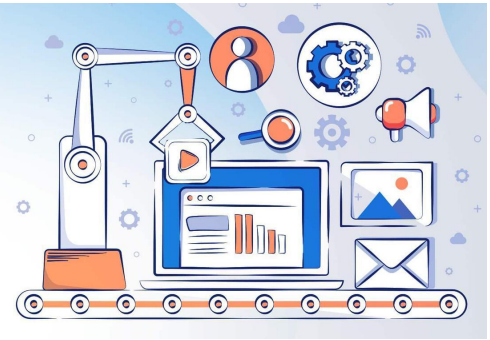
预习课

客户机（3）

连接类(conn_c)的一级方法

知识讲解

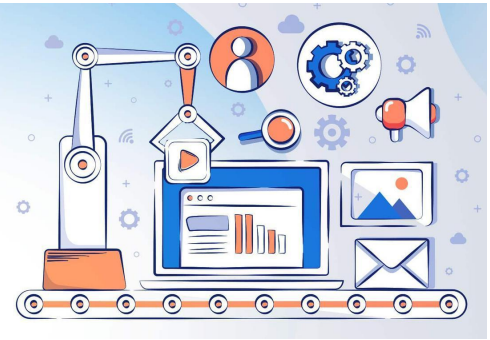
- 向存储服务器上传文件： upload
  - 构造请求
  - 发送请求
  - 逐块发送文件数据
    - 初始化未发送字节数为文件大小
    - 若还有未发送数据则循环执行
      - 发送数据
      - 未发递减
  - 接收包体
  - 解析包体
  - 释放包体
  - 返回处理结果



连接类(conn_c)的一级方法

- 向存储服务器上传文件：upload
  - 构造请求
  - 发送请求
  - 发送文件
  - 接收包体
  - 解析包体
    - 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

知识讲解



附录：程序清单

TNV/src/05_client/02_conn.cpp

知识讲解

```
//向存储服务器上传文件
int conn_c::upload(char const* appid, char const* userid,
        char const* fileid, char const* filepath) {
        //|包体长度|命令|状态|应用ID|用户ID|文件ID|文件大小|文件内容|
        //|  8  | 1| 1| 16 | 256| 128|  8  |文件大小|
        //构造请求
        long long bodylen = APPID_SIZE + USERID_SIZE + FILEID_SIZE +
                BODYLEN_SIZE;
        long long requlen = HEADLEN + bodylen;
        char requ[requlen];
        if (makerequ(CMD_STORAGE_UPLOAD,
                appid, userid, fileid, requ) != OK)
                return ERROR;
        acl::ifstream ifs;
        if (!ifs.open_read(filepath)) {
```

TNV/src/05_client/02_conn.cpp


```
            logger_error("open file fail, filepath: %s", filepath);
            return ERROR;
    }
    long long filesize = ifs.fsize();
    llton(filesize, requ + HEADLEN +
            APPID_SIZE + USERID_SIZE + FILEID_SIZE);
    bodylen += filesize;
    llton(bodylen, requ);

    if (!open())
            return SOCKET_ERROR;

    //发送请求
    if (m_conn->write(requ, requlen) < 0) {
            logger_error("write fail: %s, requlen: %lld, to: %s",
```

TNV/src/05_client/02_conn.cpp


```
                acl::last_serror(), requlen, m_conn->get_peer());
        m_errnumb = -1;
        m_errdesc.format("write fail: %s, requlen: %lld, to: %s",
                acl::last_serror(), requlen, m_conn->get_peer());
        close();
        return SOCKET_ERROR;
}

//发送文件
long long remain = filesize; //未发送字节数
off_t offset = 0; //文件读写位置
while (remain) { //还有未发送数据
        //发送数据
        long long bytes = std::min(remain,
                (long long)STORAGE_RCVWR_SIZE);
```

TNV/src/05_client/02_conn.cpp


```
long long count = sendfile(m_conn->sock_handle(),
        ifs.file_handle(), &offset, bytes);
if (count < 0) {
        logger_error(
                "send file fail, filesize: %lld, remain: %lld",
                filesize, remain);
        m_errnumb = -1;
        m_errdesc.format(
                "send file fail, filesize: %lld, remain: %lld",
                filesize, remain);
        close();
        return SOCKET_ERROR;
}
//未发递减
remain -= count;
```

TNV/src/05_client/02_conn.cpp


```
}
ifs.close();

char* body = NULL; //包体指针

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

//解析包体
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;
}

//向存储服务器上传文件
int conn_c::upload(char const* appid, char const* userid,
        char const* fileid, char const* filedata, long long filesize) {
        //|包体长度|命令|状态|应用ID|用户ID|文件ID|文件大小|文件内容|
```

TNV/src/05_client/02_conn.cpp

知识讲解

```
//|  8  | 1| 1| 16 | 256| 128|  8  |文件大小|
//构造请求
long long bodylen = APPID_SIZE + USERID_SIZE + FILEID_SIZE +
        BODYLEN_SIZE;
long long requlen = HEADLEN + bodylen;
char requ[requlen];
if (makerequ(CMD_STORAGE_UPLOAD,
                appid, userid, fileid, requ) != OK)
        return ERROR;
llton(filesize, requ + HEADLEN +
        APPID_SIZE + USERID_SIZE + FILEID_SIZE);
bodylen += filesize;
llton(bodylen, requ);

if (!open())
```

TNV/src/05_client/02_conn.cpp


```
        return SOCKET_ERROR;

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

    //发送文件
    if (m_conn->write(filedata, filesize) < 0) {
```

TNV/src/05_client/02_conn.cpp


```
        logger_error("write fail: %s, filesize: %lld, to: %s",
                acl::last_serror(), filesize, m_conn->get_peer());
        m_errnumb = -1;
        m_errdesc.format("write fail: %s, filesize: %lld, to: %s",
                acl::last_serror(), filesize, 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
```

TNV/src/05_client/02_conn.cpp

```
        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;
}
```

直播课见