# 整体结构

文件发送时被分为多个段，FileClient从FileServer获取分段大小。

# 文件下载

1. FileClient发送Request.DOWNLOAD消息

Request req=new Request();

req.setType(Request.DOWNLOAD);

req.setFile(remoteFile); //请求的文件

channel.writeAndFlush(req);

1. FileServer将Request.DOWNLOAD消息路由给ServerDownLoadProcessor. processDownload实例方法处理，返回Response.RESDOWNLOAD消息，该消息包含文件的分段大小，以及总的段数

Response response=new Response();

response.setType(Response.RESDOWNLOAD);

//请求文件存在

response.setRetCode(Response.SUCESS);

response.setBufferSize(bufferSize); //文件分割后每段的大小

response.setSumSeg(getFileSegs(raf.length())); //总的分段数

//请求文件不存在

response.setRetCode(Response.FILENOTFOUND);

ctx.writeAndFlush(response);

1. FileClient将Response.RESDOWNLOAD消息路由给ClientDownLoadProcessor.processDownLoad实例方法处理，发送Request.REQFILESEGMENT消息，请求文件的每一个段

Request req=new Request();

req.setType(Request.REQFILESEGMENT);

req.setCurSeg(curSeg);

ctx.writeAndFlush(req);

1. FileServer将消息路由给ServerDownLoadProcessor.processSegment实例方法处理，按段发送文件内容，消息类型Response.DOWNLOADCONTENT

Response res=new Response();

res.setType(Response.DOWNLOADCONTENT);

res.setCurSeg(request.getCurSeg());

res.setBuffer(buffer);

ctx.writeAndFlush(res);

1. FileClient将消息路由给ClientDownLoadProcessor.processDownLoadContent实例方法处理，将当前段保存下来，请求下一个段，循环3-5，直到文件下载完成。

# 文件上传

1. FileClient发送Request.UPLOAD消息

Request req=new Request();

req.setType(Request.UPLOAD);

req.setFile(remoteFile);

req.setFileSize(fileSize);

channel.writeAndFlush(req);

1. FileServer将消息路由给ServerUploadProcessor. processUpload实例方法

Response res=new Response();

res.setType(Response.RESUPLOAD);

//server上文件存在

res.setRetCode(Response.FILEEXIST);

ctx.writeAndFlush(res);

ctx.close();

//文件

long fileSize=request.getFileSize();

sumSeg=getFileSegs(fileSize);

res.setRetCode(Response.SUCESS);

res.setBufferSize(bufferSize);

res.setSumSeg(sumSeg);

ctx.writeAndFlush(res);

//请求第一个分段

Response firstRes=new Response();

firstRes.setType(Response.REQUPLOADSEG);

firstRes.setCurSeg(curSeg);

ctx.writeAndFlush(firstRes);

1. FileClient接收到两个消息，Response.RESUPLOAD被路由到ClientUploadProcessor.processUpload实例方法，做数据传送的初始化工作；Response.REQUPLOADSEG被路由到ClientUploadProcessor.processSegment实例方法，开始发送分段

Request req=new Request();

req.setType(Request.UPLOADCONTENT);

req.setCurSeg(curSeg);

req.setBuffer(buffer);

ctx.writeAndFlush(req);

1. FileServer将消息路由到ServerUploadProcessor .processUpLoadContent，将分段内容写入文件，请求下一个分段，重复请求，直到文件传送完毕。