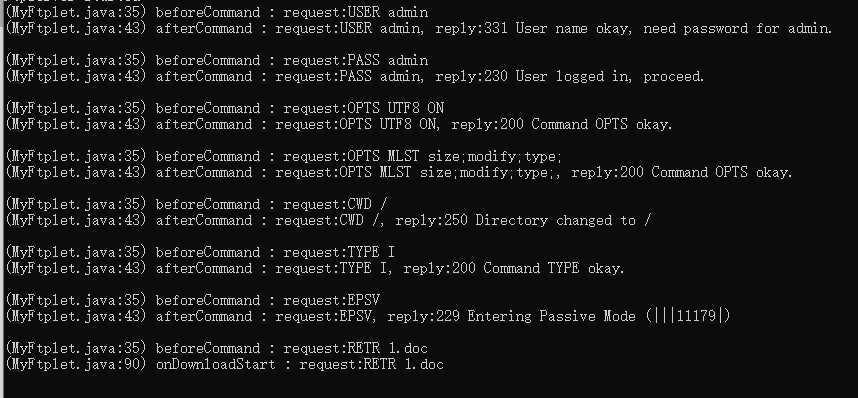
# 1、下载时触发的命令

FileZilla客户端，点击下载的时候：



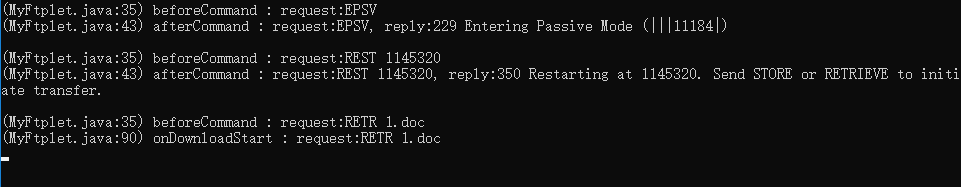
这么多的命令其实大概就是先刷新一下登录、然后刷新一下远程列表，然后设置目录，设置传输类型，设置被动模式，最后是下载命令。

然后中断：



捕捉到异常，并输出信息，这里输出的是我自定义的消息，在onDownloadStart中模拟下载的逻辑然后返回SKIP不执行服务器默认的RETR操作。

然后续传：



先设为被动模式，然后发送REST + 文件大小的命令（这里不是文件名，之前说错的，因为忘记了这个命令），session中就会记住了这个偏移，在下次的下载中就能获取这个偏移（session.getFileOffset();）。

如果REST后面不是跟随RETR或者STOR，会视为非法或者无效指令。

# 2、测试把max-threads和max-logins调大的结果

测试环境：

win10 64位

处理器Inter(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40GHz

16GB内存

apache ftpserver服务器

设置max-threads = 100，max-logins = 80

符合预期：有7个同时下载10个文件，第8个同时下载2个文件，即

7 \* (10 + 1) + (1\*2 +1) = 80

这时总线程数



设置max-threads = 500，max-logins = 480

这个时候由于程序太多，已经无法统计同时下载数量

在收到421登录限制的时候，看到占用的线程是505：



测试的时候，大概在第40个以上，访问速度就有点慢了，就是连接到服务器，不能马上就连上，会等一小段时间才收到欢迎。大概43-45个窗口的时候，就已经收到421回复了，主要是没有设置空闲时间，所以有一些超过2分钟然后退出了，导致无法统计，但估算一下，一个窗口正常情况占1个登录10个下载，44\*11=484，的确是符合预期的480，这里是没有超过threads的，如果超过是收不到回复的。

============================================

测试环境：

win7 64位

处理器Intel(R) Pentium(R) CPU G620 @ 2.60GHz 2.60GHz

8GB内存

apache ftpserver服务器

设置max-threads = 300，max-logins = 280

在第23个窗口的时候，FileZilla客户端在连接的时候客户端程序无响应了，然后等一段时间还是能继续下载的，不过后面的客户端在连接的时候都会卡到没响应，就没继续测试了

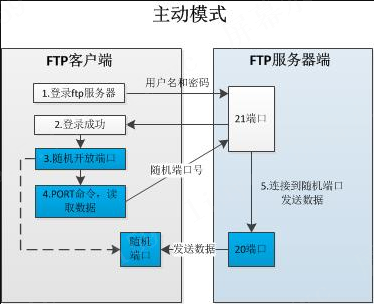


这个时候大概是22 \* (1 + 10) = 242，还没达到280，的确是能同时下载

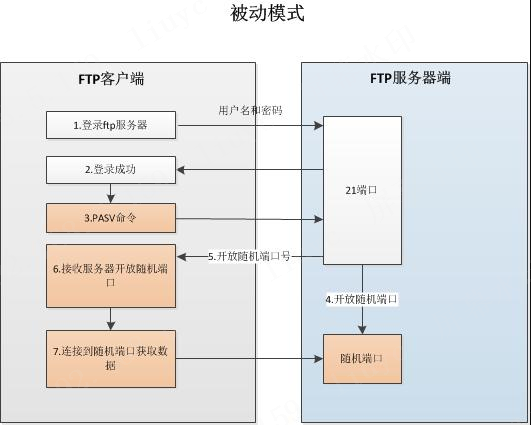
# 3、被动模式和主动模式

**FTP支持两种模式，一种方式叫做Standard (也就是 PORT方式，主动方式)，一种是 Passive (也就是PASV，被动方式)。 Standard模式 FTP的客户端发送 PORT 命令到FTP服务器。Passive模式FTP的客户端发送 PASV命令到 FTP Server。**

Port模式FTP 客户端首先和FTP服务器的TCP 21端口建立连接，通过这个通道发送命令，客户端需要接收数据的时候在这个通道上发送PORT命令。 PORT命令包含了客户端用什么端口接收数据。在传送数据的时候，服务器端通过自己的TCP 20端口连接至客户端的指定端口发送数据。 FTP server必须和客户端建立一个新的连接用来传送数据。



Passive模式在建立控制通道的时候和Standard模式类似，但建立连接后发送的不是Port命令，而是Pasv命令。FTP服务器收到Pasv命令后，随机打开一个临时端口（也叫自由端口，端口号大于1023小于65535）并且通知客户端在这个端口上传送数据的请求，客户端连接FTP服务器此端口，然后FTP服务器将通过这个端口进行数据的传送，这个时候FTP server不再需要建立一个新的和客户端之间的连接。



测试：

设置FileZilla客户端为主动连接

然后设置max-threads="5" max-logins="7"，还是只能下同时下载5个文件。

然后不设置max-threads，设置max-logins="7"，还是只能同时下载6个文件，因为登录占了1个单位

所以主动模式和被动模式与max-threads和max-logins配置之间没有发现差异