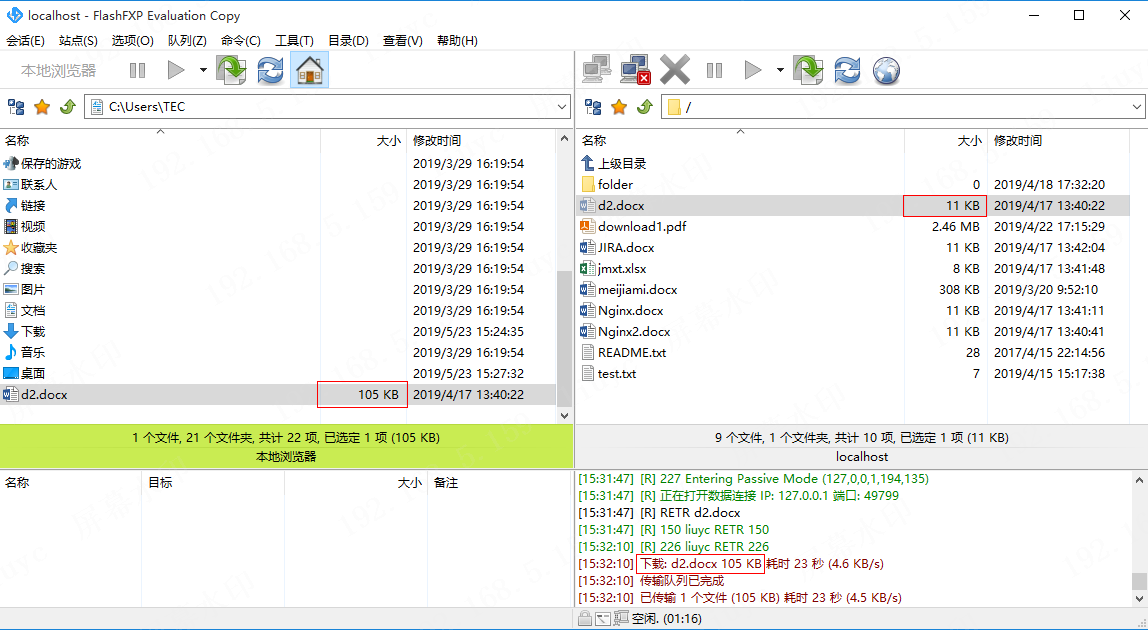
# 1、FlashFXP

## 1.1、直接下载，正常



但它显现的进度是



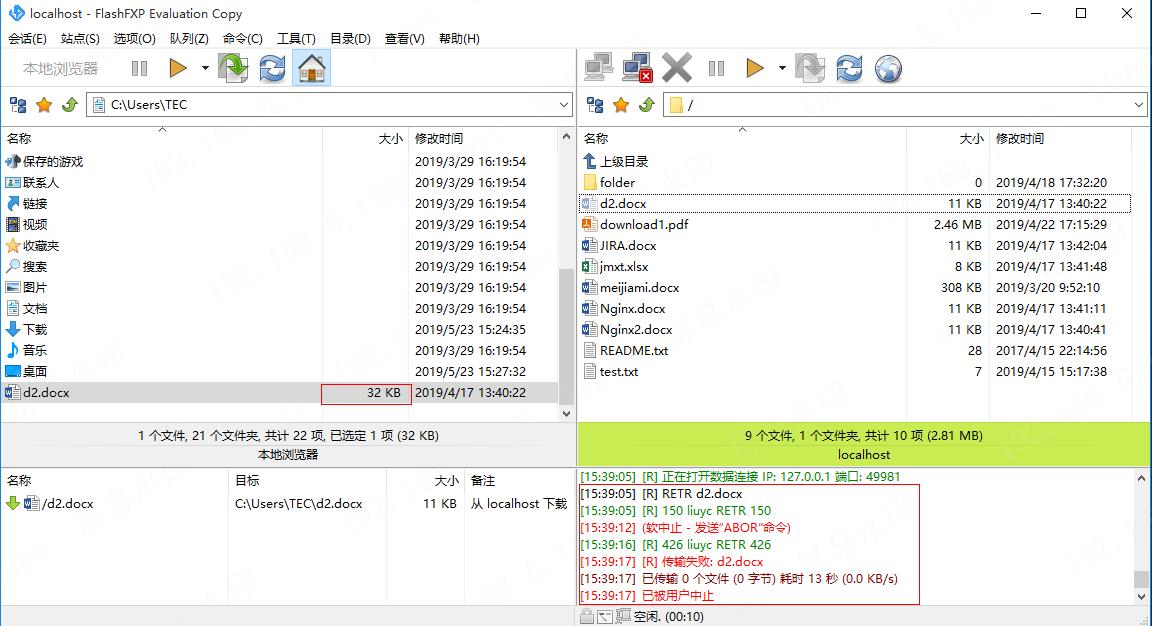
一开始能有进度条，当进度超出11KB（源文件大小）之后，进度条消失



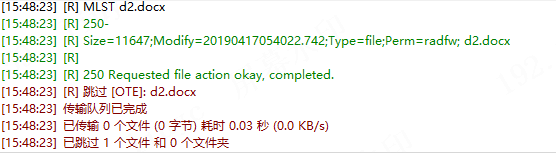
## 1.2、断点续传，会出现跳过

### （1）第一种情况

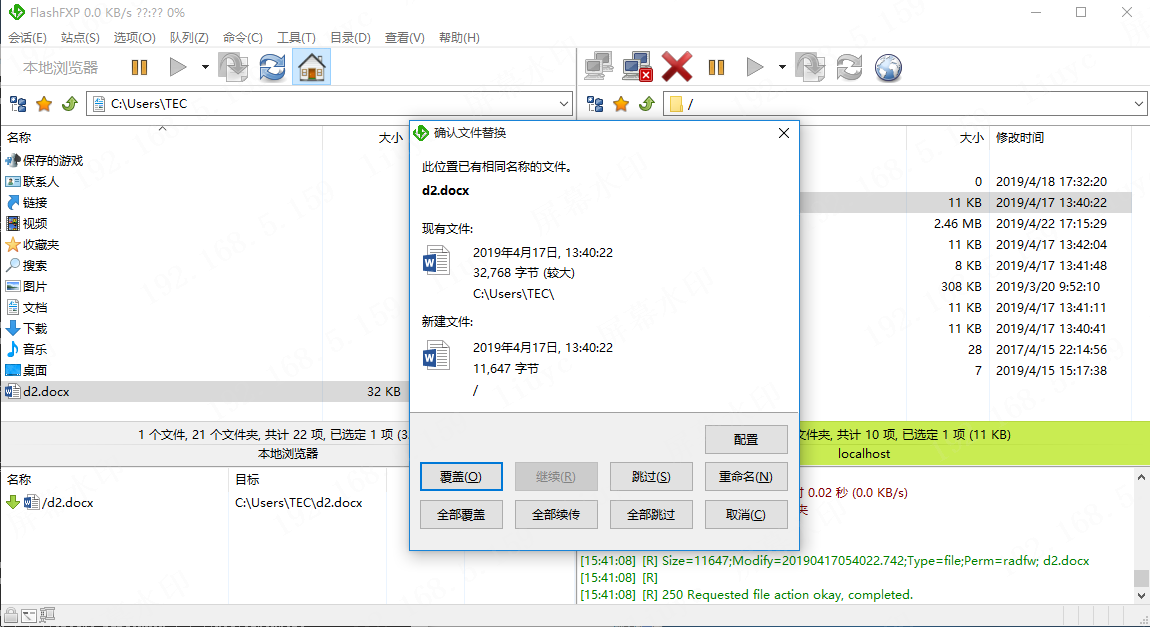
在下载过程中中止：



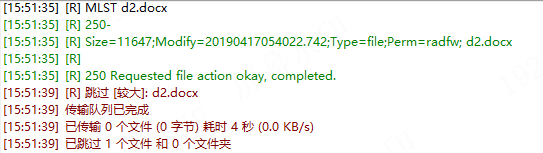
点击开启传输队列，跳过了文件，无法继续传输：



接着点击“传输指定项”，无法续传：

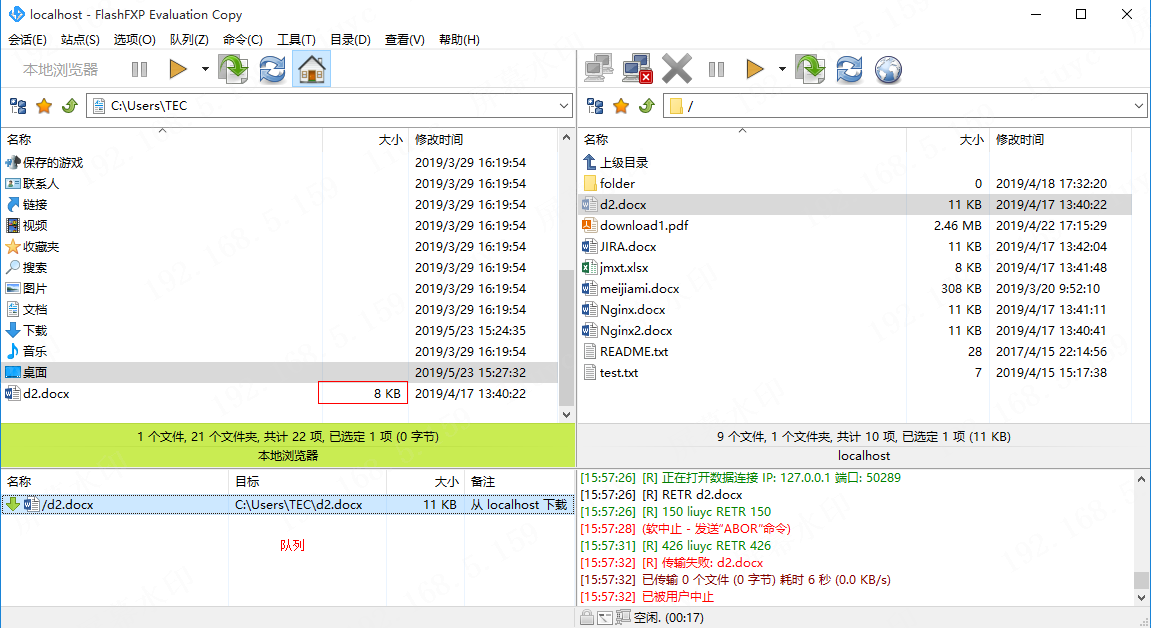


点击全部续传也不行，仍会跳过：

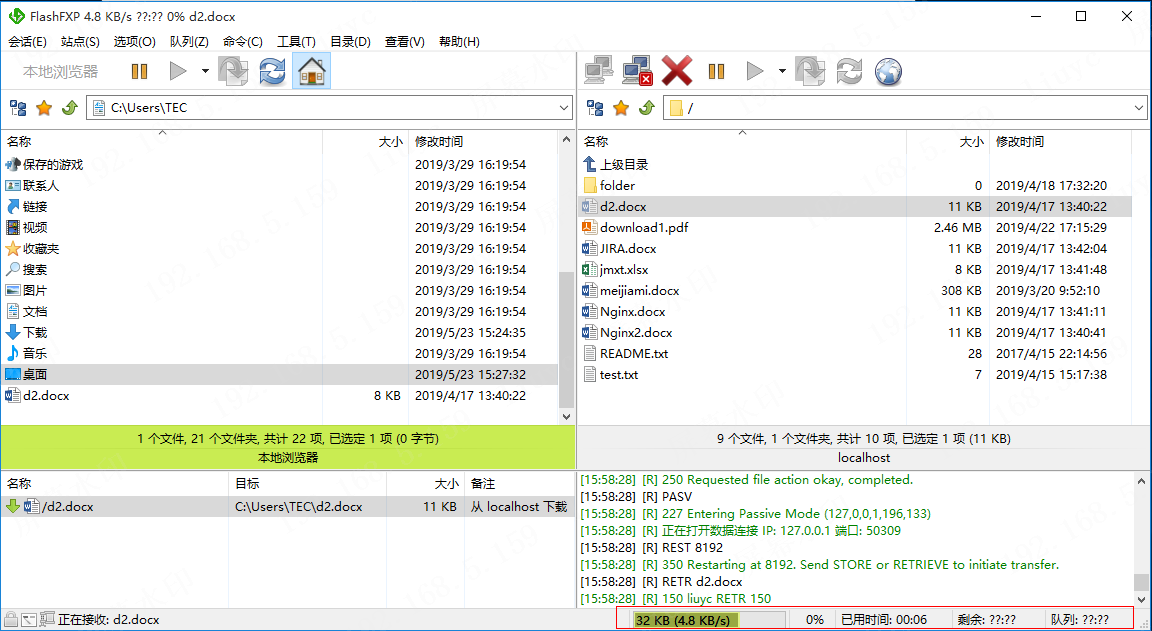


### （2）第二种情况

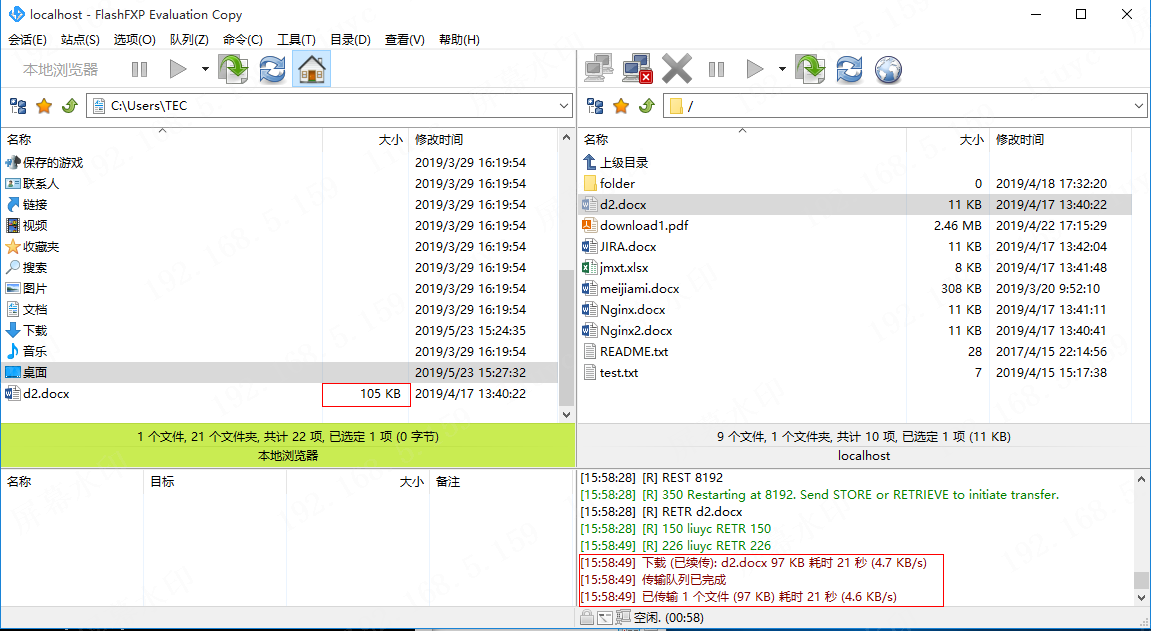
上面的例子中，在中止的时候传输大小超过了源文件的大小，现在在传输大小小于源文件大小的时候中止。



点击开启传输队列，可以继续传，但进度条是停留在上次中止的位置：



结果：



### （3）补充说明

这里再一次说明一下当前的apache ftpserver处理的情况，当我下载一个文件时（如d2.docx 11K），在ftpserver中我会根据当前用户的工作目录，找出该用户的加密文件目录（当前是固定映射目录），然后在加密文件目录下寻找相同文件名的文件（d2.docx）来下载，如果没找到相同文件名的文件，就对源文件进行加密操作（模拟的处理），并把处理后的文件保存在加密文件目录，并下载新的文件。

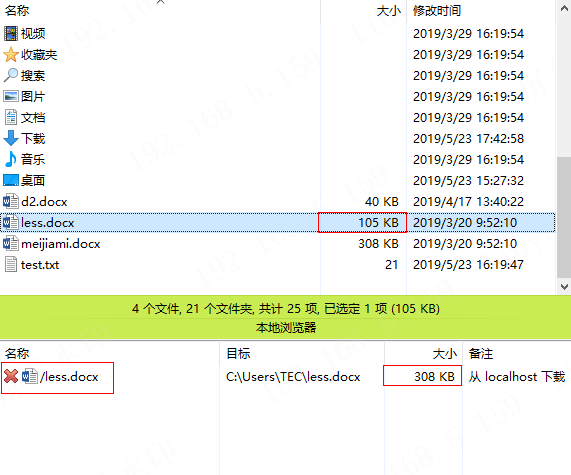
然后测试了下载上图中的meijiami.docx文件，在加密目录中也事先保存了一份没加密处理的文件meijiami.docx，依然是执行上面的过程（根据用户找加密文件目录下载同名文件）。正常处理下载和续传。

**额外测试：**

由于测试了 源文件大小<处理后文件大小 和 源文件大小=处理后文件大小，这里再测试 源文件大小（309K）>处理后文件大小（106K） 的情况（一般不考虑这种情况，只在FlashFXP中测试一次）。当下载到106K的时候，它会报传输失败，并以续传的方式重试：



但实际上已经完成接收了所有的数据，所以文件能正常打开：



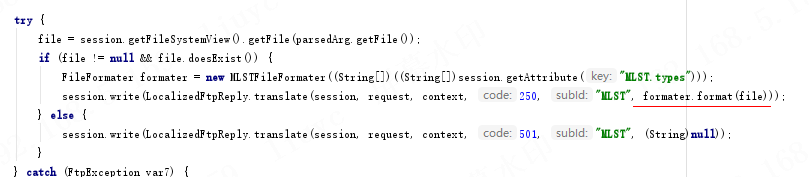
### （4）解决：

在服务端收到MLST命令，该命令会返回文件大小给客户端，猜测是该命令导致客户端实现了跳过的逻辑。重新实现了MLST命令，在返回之前，对内容进行字符串的替换：

250 Size=11647;Modify=20190417054022.742;Type=file;Perm=radfw; d2.docx

250 Size=107718;Modify=20190417054022.742;Type=file;Perm=radfw; d2.docx

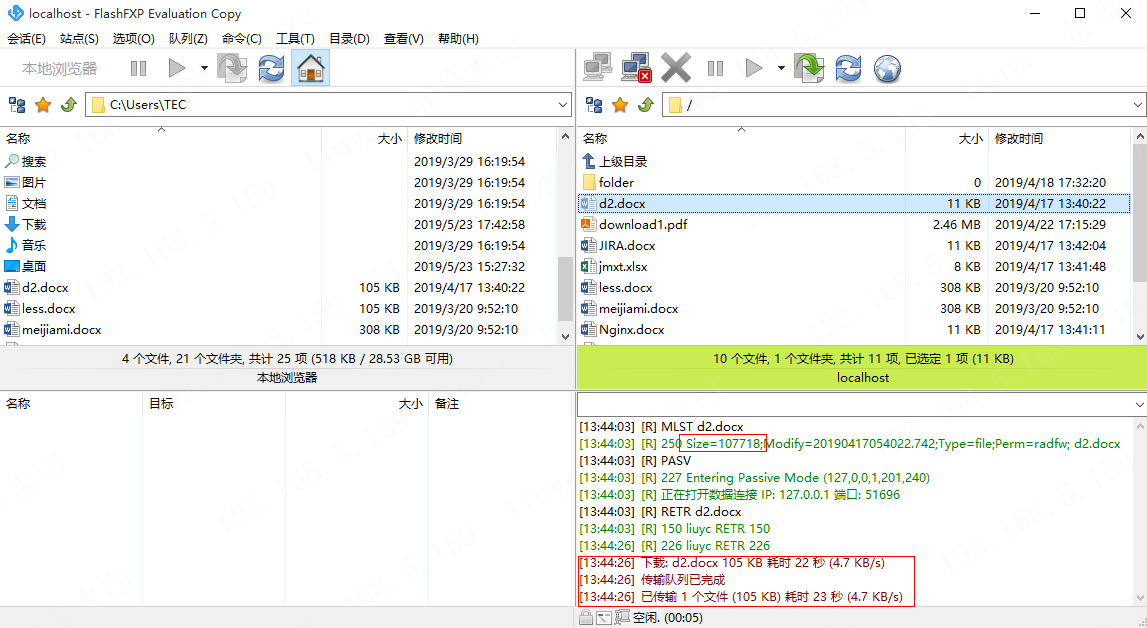
源码：



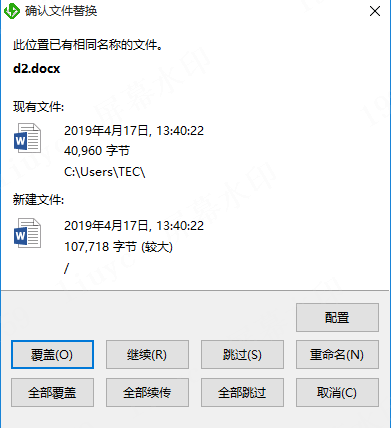
修改后：



MLST修改文件大小之后，之前的[1.1](#_1.1、直接下载，正常)的情况能正常显示进度条，[1.2中的第三点](#_（3）补充说明)中的源文件大小（309K）>处理后文件大小（106K）也没有报错了，进度也是正常，大概就可以确定该软件是以MLST命令返回的文件大小作为百分比分母。



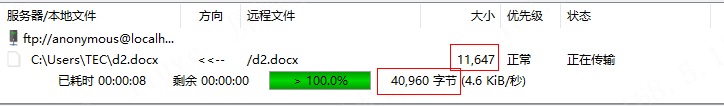
并且[1.2（1）](#_（1）第一种情况)的续传也变正常，进度也是正常显示。



通过这里的修改，回看8uftp，发现8uftp发送的是REST命令，然后该命令我是没有实现过的，它返回350 Restarting at 0. Send STORE or RETRIEVE to initiate transfer. 正常是应该收到Restarting at 文件偏移量.。 如果真的把REST返回值改为文件的偏移量，那8uftp客户端可能又会无法实现断点续传（如FlashFXP比较大小更大就跳过了的情况，这里我没测试过）。至于8uftp是通过什么来决定进度的分母，仍未知。但修改了MLST命令，并没有影响8uftp和FileZilla的运行，而且FlashFXP也变正常，并且比8uftp、FileZilla多一个进度条正常的显示。

# 2、FileZilla

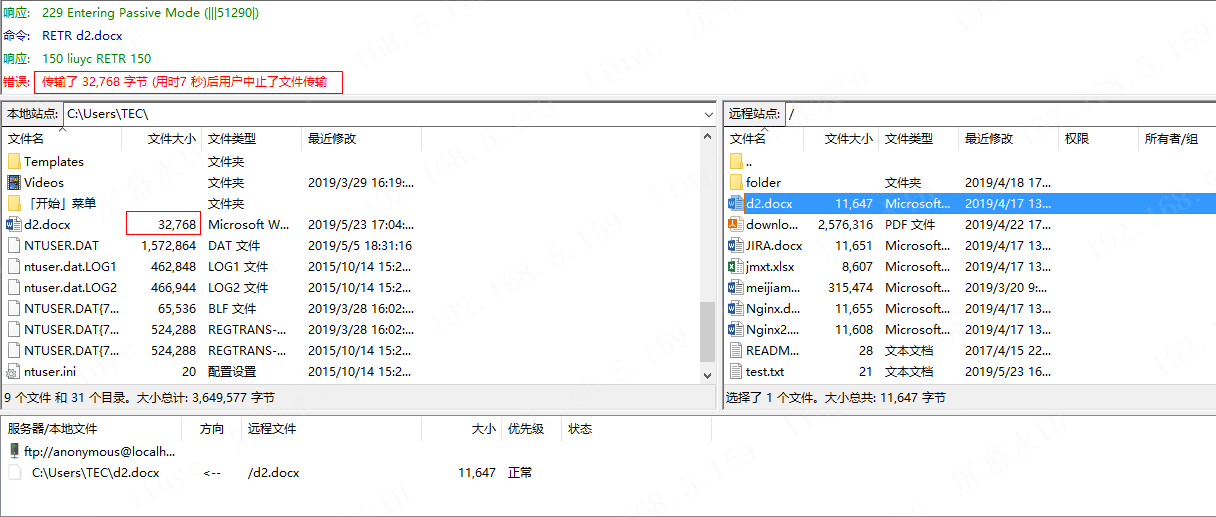
## 2.1、直接下载，正常



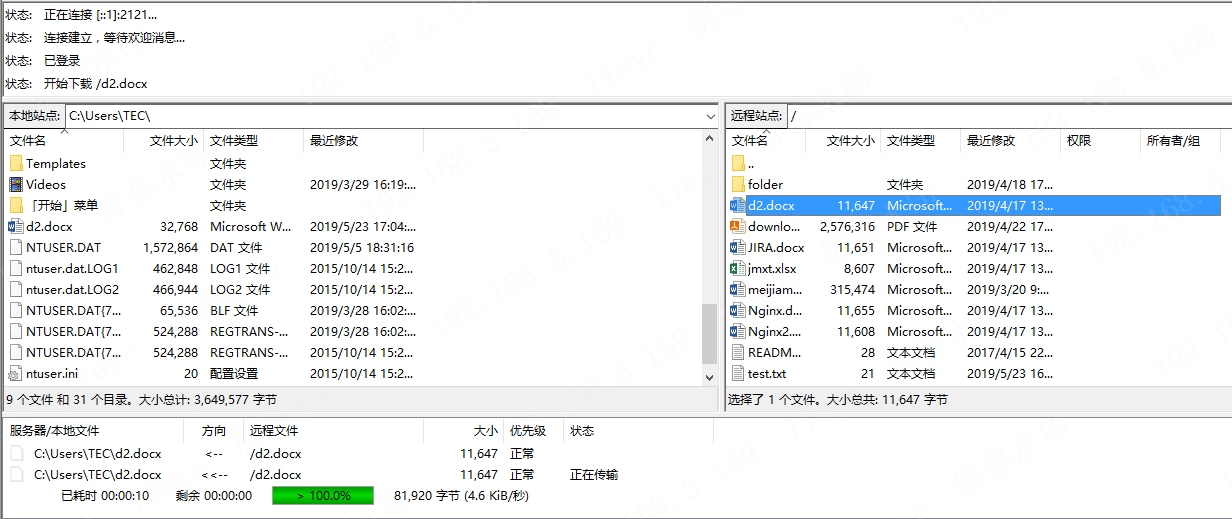
效果与8UFTP.exe相同，具体情况查看之前的《apache ftpserver文档.docx》

## 2.2、断点续传，正常

在传输过程中中止：



能正常续传：



效果也是和8UFTP.exe相同，同样可以参考之前的文档。

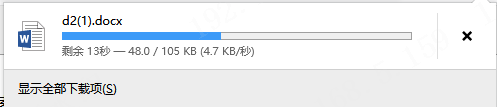
# 3、IE浏览器

能显示正常进度，即以105KB作为100%（源文件为11K）



# 4、firefox

也是正常下载



下载，命令 onDownloadStart -- RETR -- onDownloadEnd

然后我在onDownloadStart中实现自定义的RETR命令，最后返回SKIP，不让它执行默认的RETR。

下面把自定义解析为在ftplet api中实现自定义功能的意思。

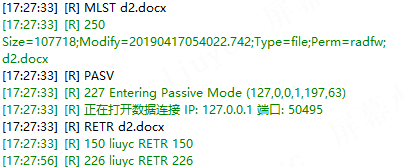
默认就是执行默认的ftplet api，没有实现额外的处理。

# 8uftp

自定义的和默认的下载效果相同。只是自定义的进度条仍是不对的。

# FlashFXP

左边是自定义下载的，右边是默认下载的



比默认的多发送了一个MLST命令，每次都是。未清楚原因。但经过自定义的MLST处理后，都能正常下载和显示正确的进度条。

# FileZilla

表现和8uftp一样

续传：自定义的报了个错Socket exception during data transferSocket closed

正常的应该返回426 Data connection error.