收发文件大小测试

划分等价类（上传文件）

0mb~1mb 115kb 0s

1mb~3mb 1.12mb 2s

3mb~5mb 4.15mb 6s

5mb~7mb 5.92mb 16s

>7mb 13.7mb 42s

（可能是和socket有关系，从客户端上传的速度明显比从服务器下载的速度慢很多，就是send很快，但server的recv速度不知道为何非常慢，所以只能发送小一点的文件）

划分等价类（下载文件）

0mb~100mb 3.05mb 0s

100mb~500mb 166mb 2s

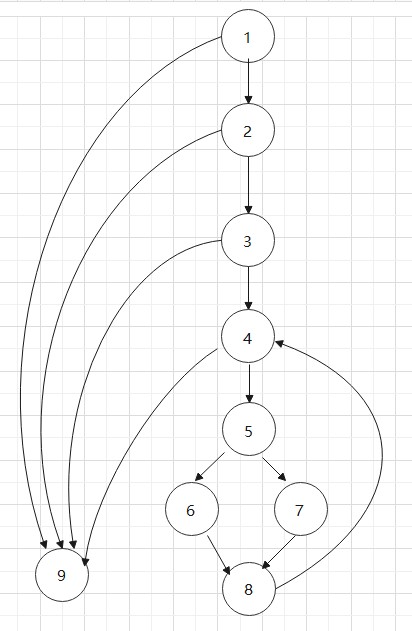
500mb~1G 731mb 23s

>1G 1.08G 46s

（建议大小是1G以下，文件过大的时候测试时有过一次电脑崩溃，且在大小超过500mb后传输耗时明显提升）

白盒测试（客户端下载文件）

Client：downloadFile()



环形复杂度为6

1：1-9

2：1-2-9

3：1-2-3-9

4：1-2-3-4-9

5：1-2-3-4-5-6-8-4-9

6：1-2-3-4-5-7-8-4-9

测试用例 输入 预期输出

路径1 不选择要下载的文件 点击download无反应

路径2 不选择存放文件夹 弹出警告窗口

路径3 没有得到有效文件名 弹出警告窗口

路径4 所选文件大小为0， 弹出完成窗口

不进入while循环

路径5 下载文件大小大于缓存区大小， 弹出完成窗口

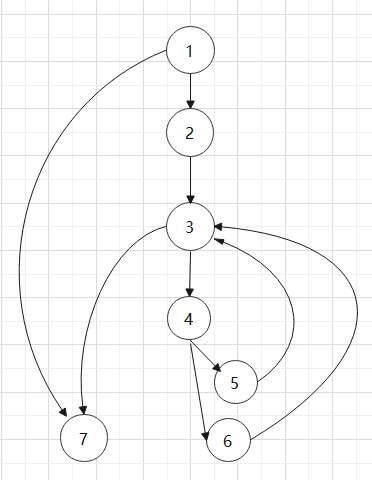
继续下载直到下载完所有数据

路径6 下载文件大小小于缓存区大小， 弹出完成窗口

经历一次while就下载完

白盒测试（客户端上传文件）

Client：uploadFile()



环形复杂度为4

1: 1-7

2: 1-2-3-7

3: 1-2-3-4-5-3-7

4：1-2-3-4-6-3-7

测试用例 输入 预期输出

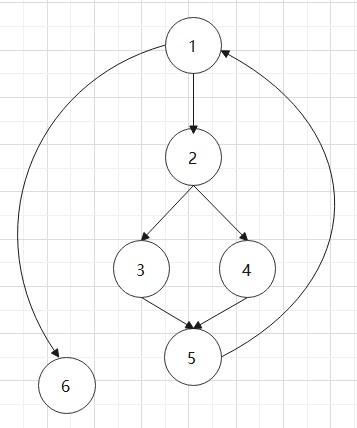
路径1 不选择要上传的文件 点击upload无反应

路径2 上传列表为空 弹出完成窗口

路径3 发送完文件收到EOF 回复  弹出完成窗口

路径4 发送完文件没收到EOF 回复  print报错并弹出完成窗口

Server：recvFile()



环形复杂度为3

1：1-6

2：1-2-3-5-1-6

3：1-2-4-5-1-6

测试用例 输入 预期输出

路径1 接收文件大小为0 print相关信息，发送EOF

路径2 文件大小小于缓存区大小， print相关信息，发送EOF

一次接收

路径3 文件大小大于缓存区大小， print相关信息，发送EOF

分批接收