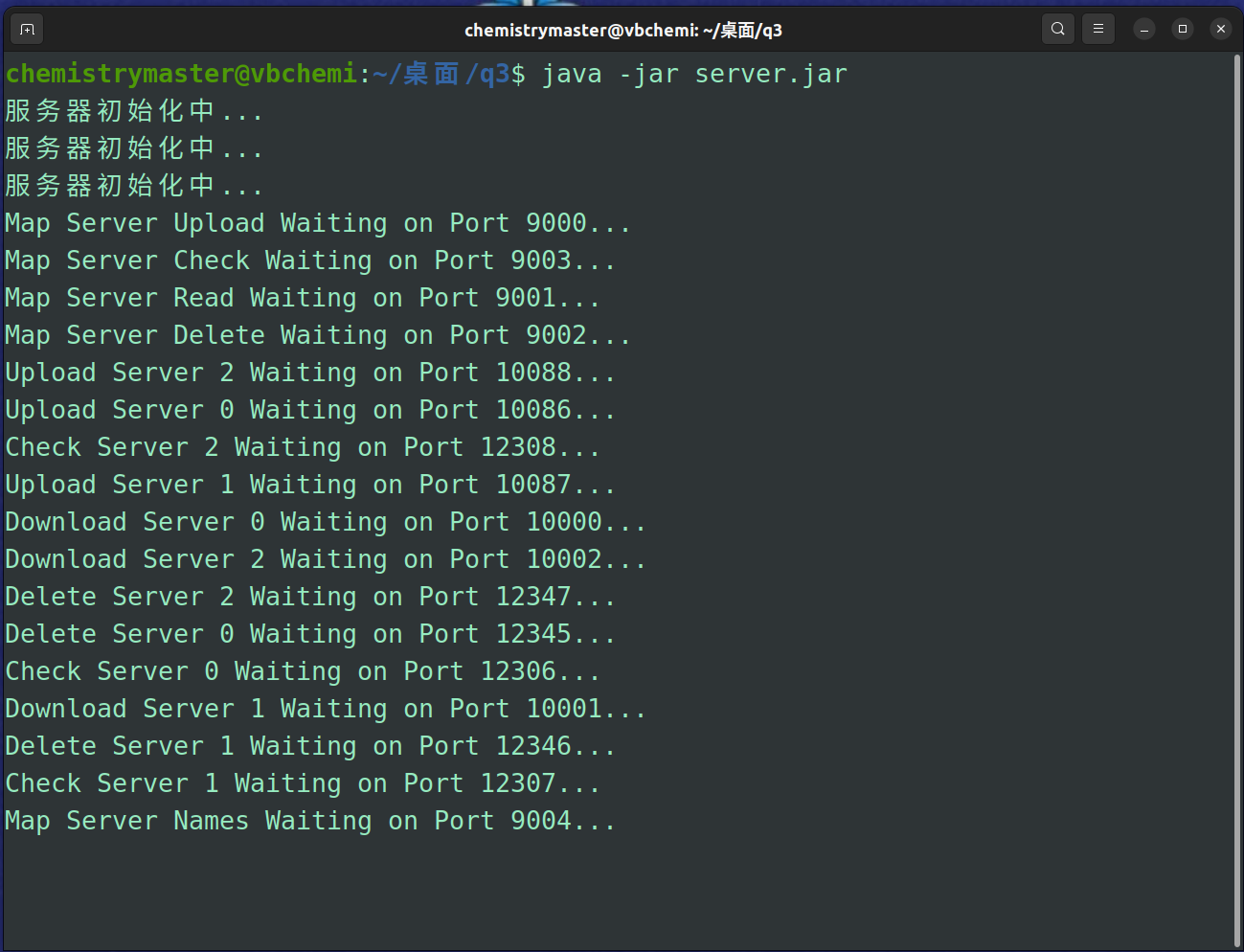
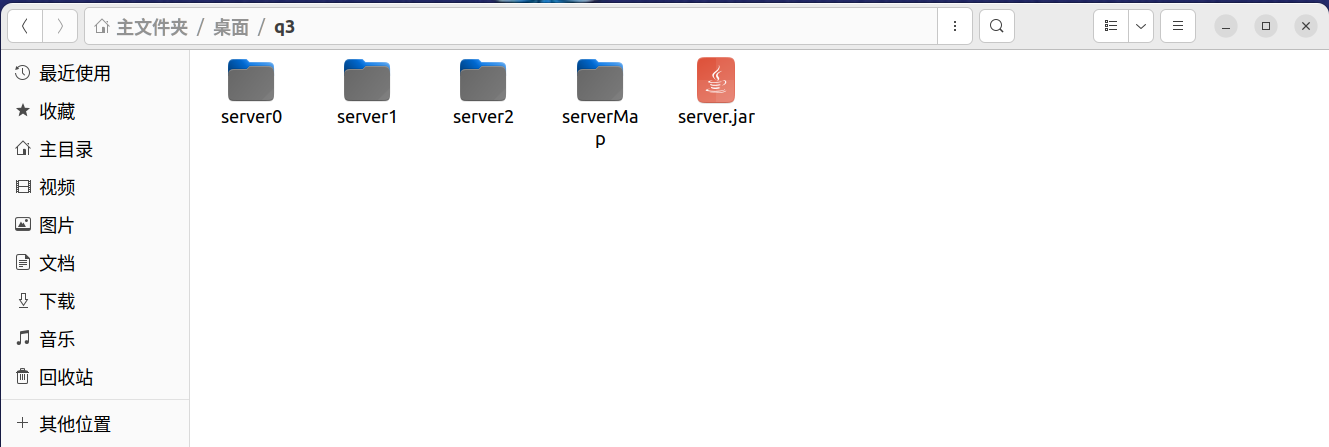
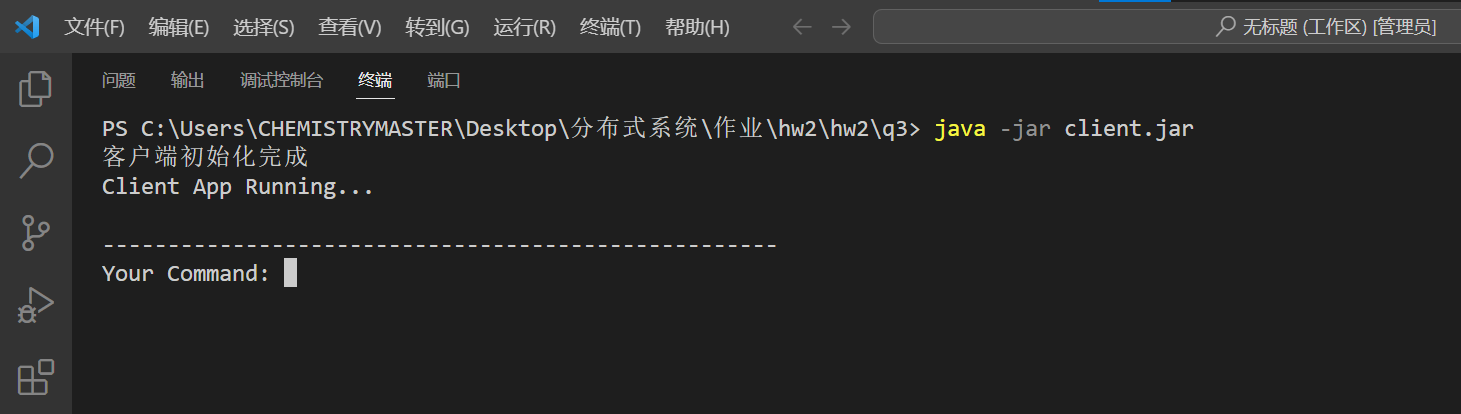
**准备工作：**

1.打开服务器，准备连接（在虚拟机上用多个端口模拟了四个服务器，其中三个存储块文件，一个存储映射表）。初始化时会在jar包相同目录下创建各个服务器的目录



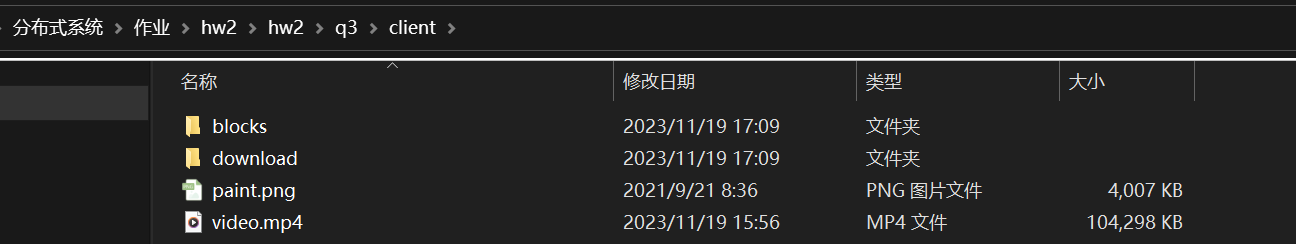


2.打开客户端程序（客户端有一个根目录client，里面有两个文件夹blocks（上传时暂存分块文件）和download（下载目录），要上传的文件需要放入client目录下才可以上传）





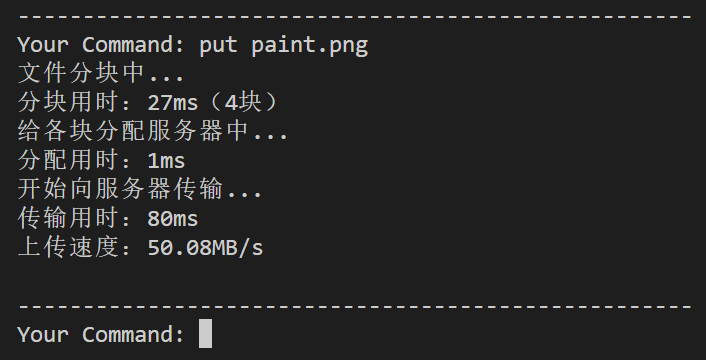
3.将待上传文件paint.png和video.mp4放入client目录中

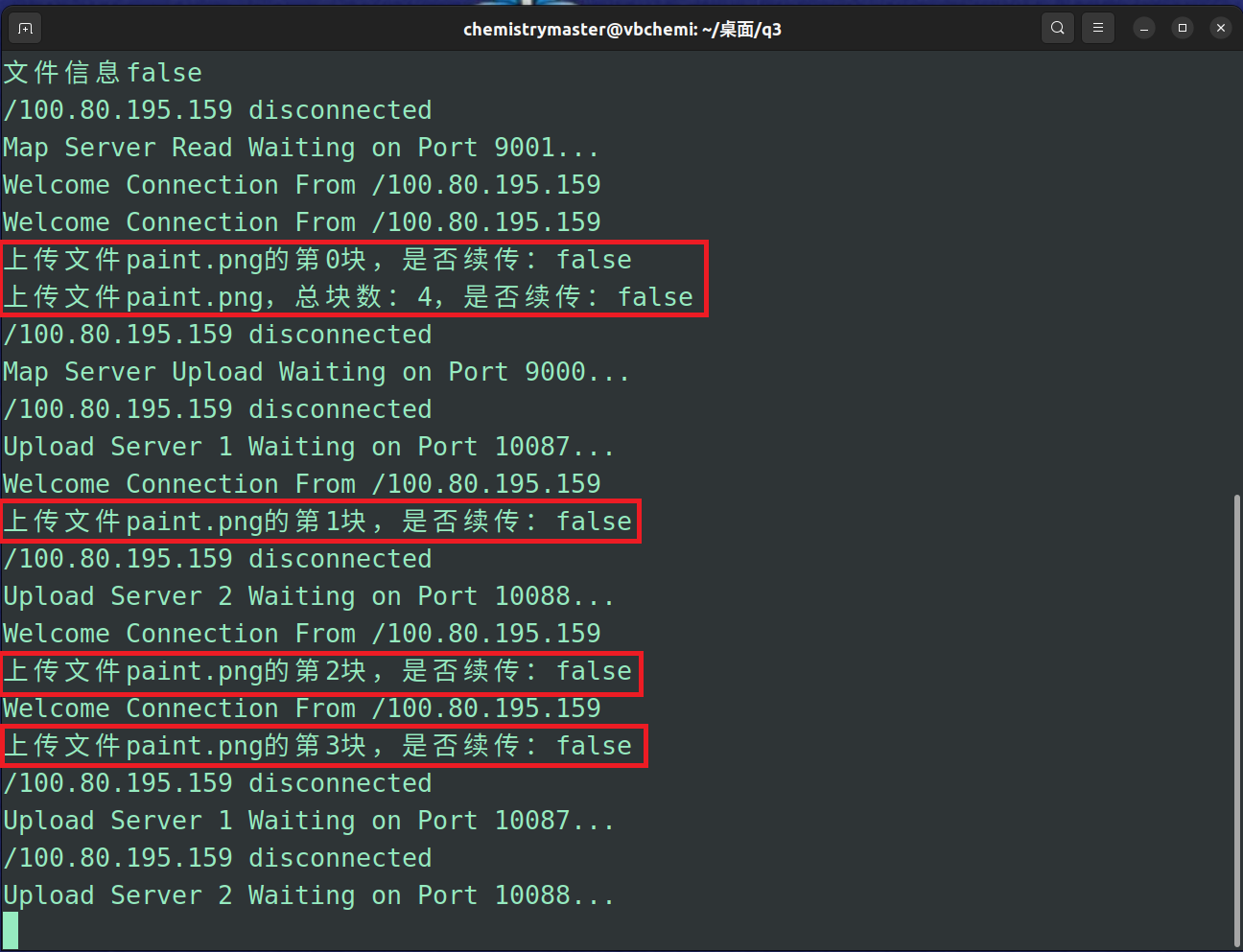


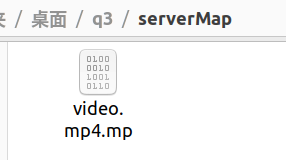
**基本功能**

**1.put**

1.1 将paint.png上传

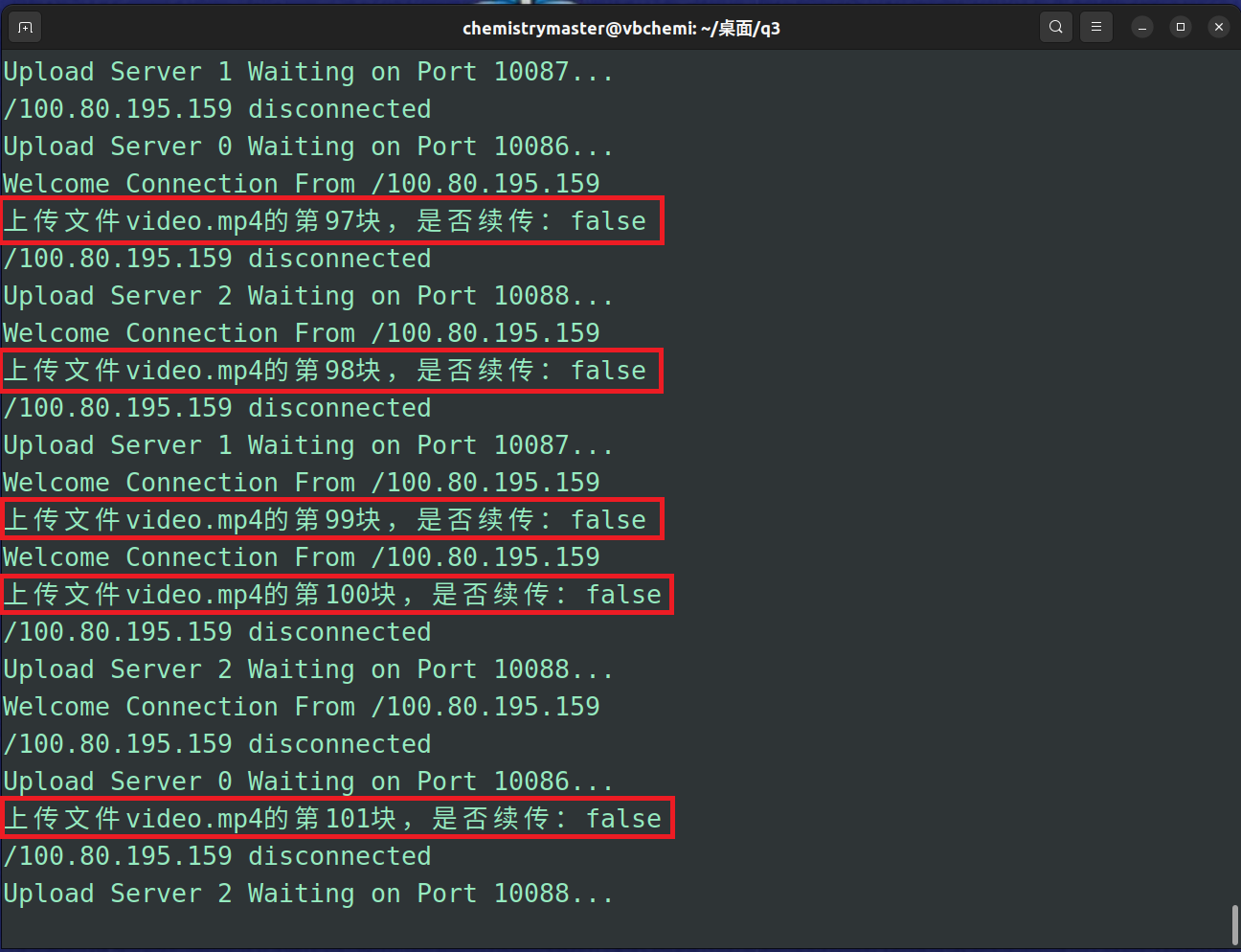


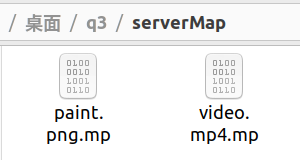
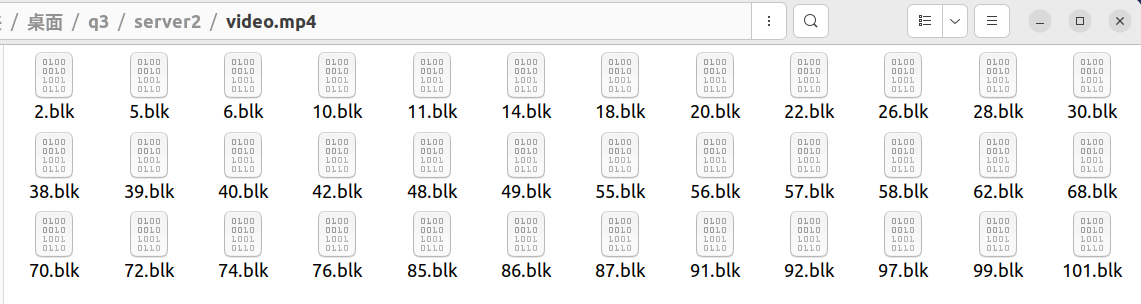
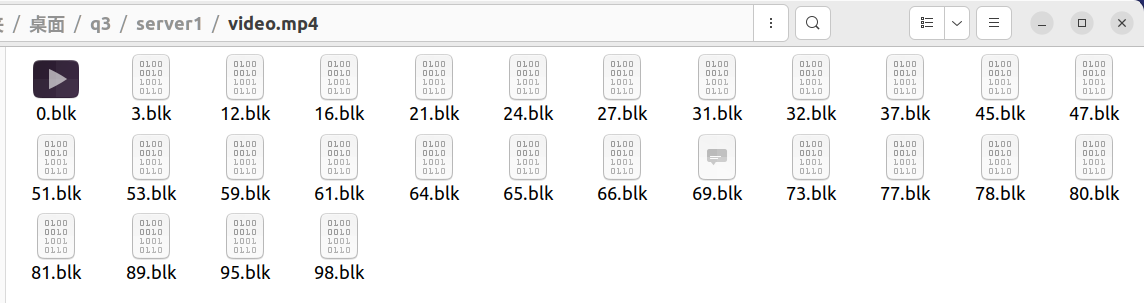
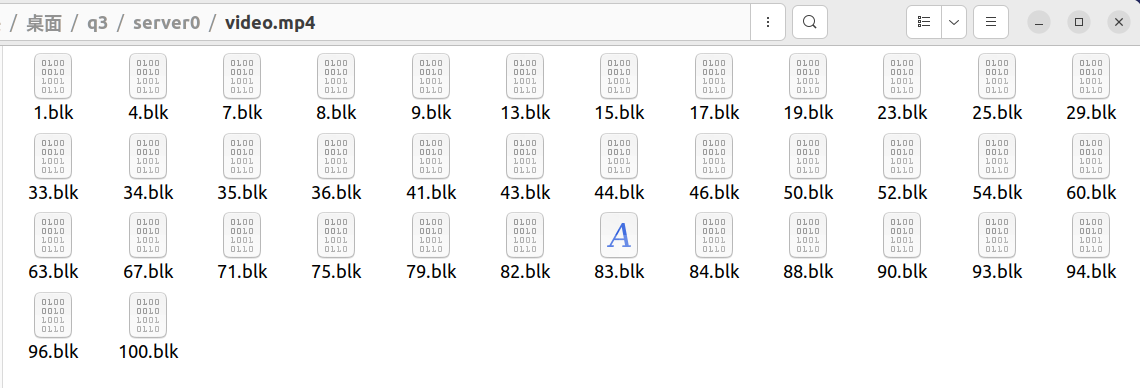




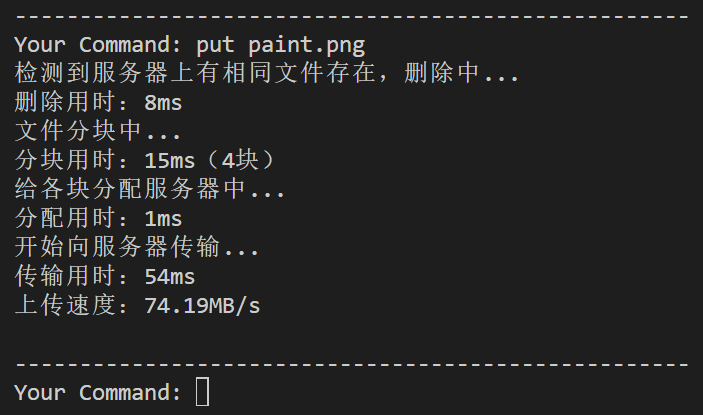
1.2 将video.mp4上传

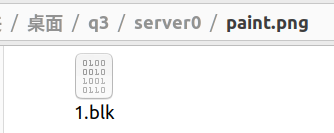




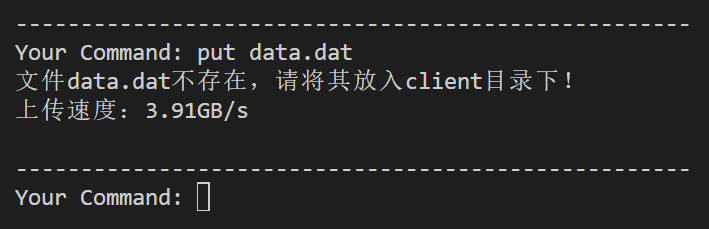


1.3 重复上传paint.png（会将原有文件删除，然后重新上传）



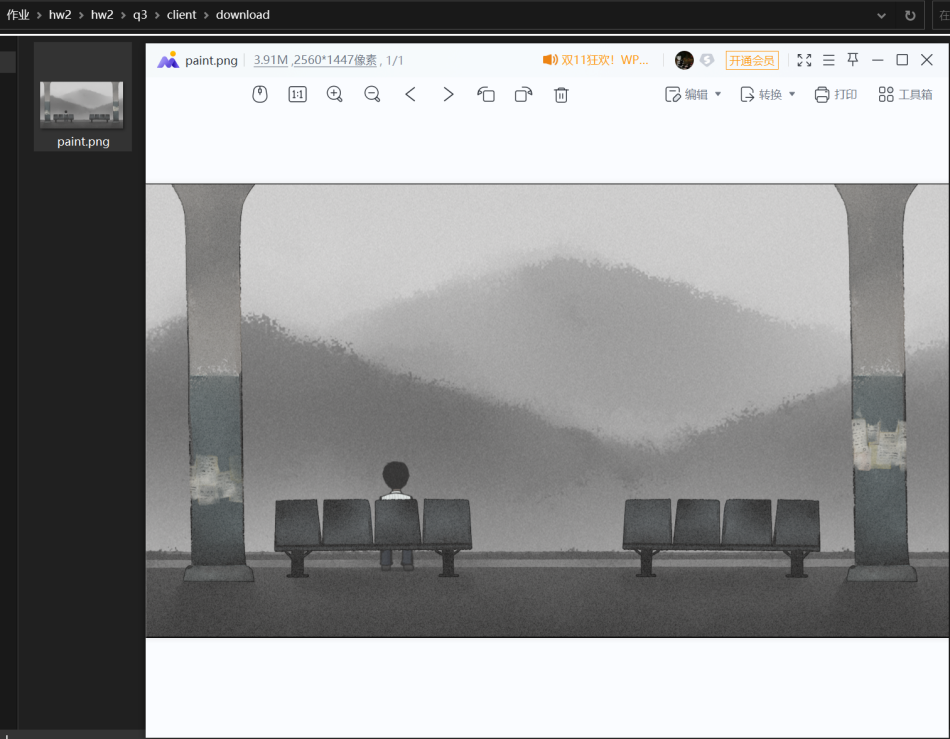
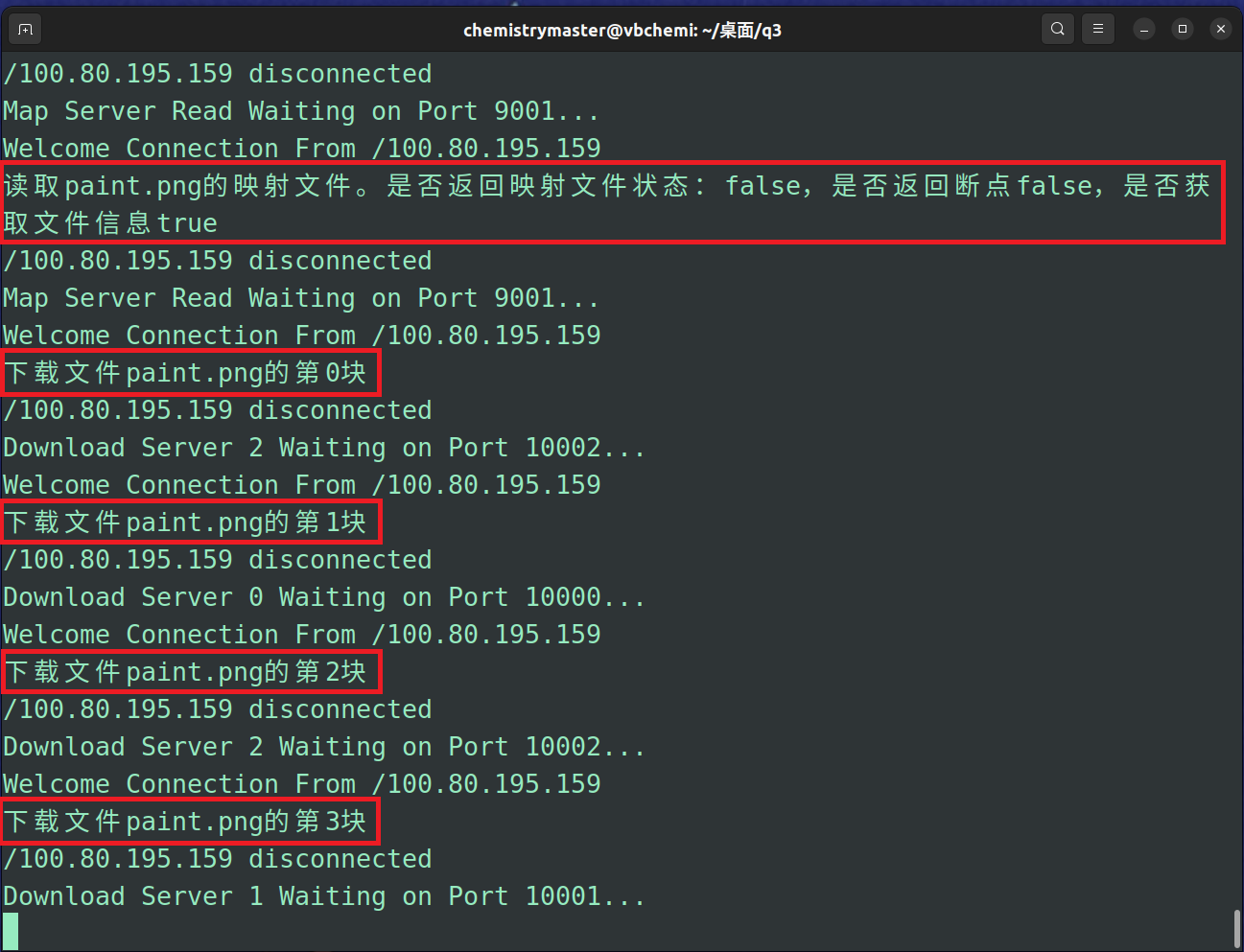
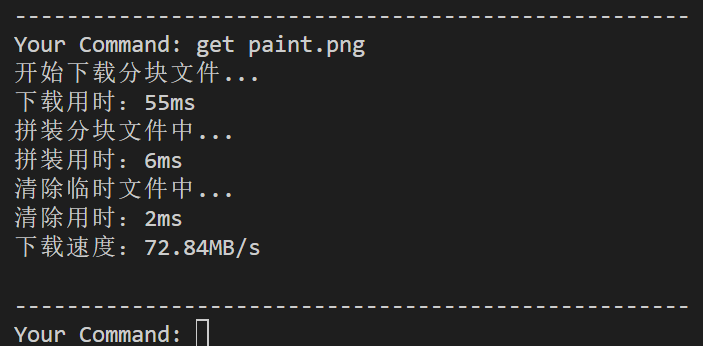


1.4 上传一个不存在的文件（会打印提示信息）

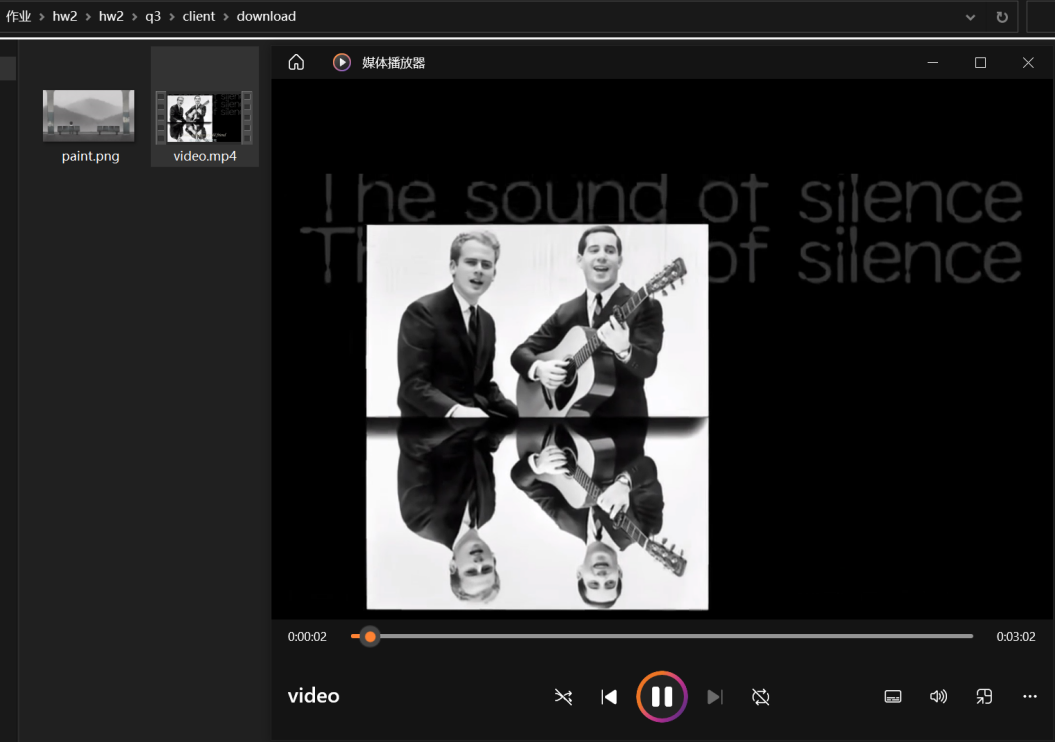
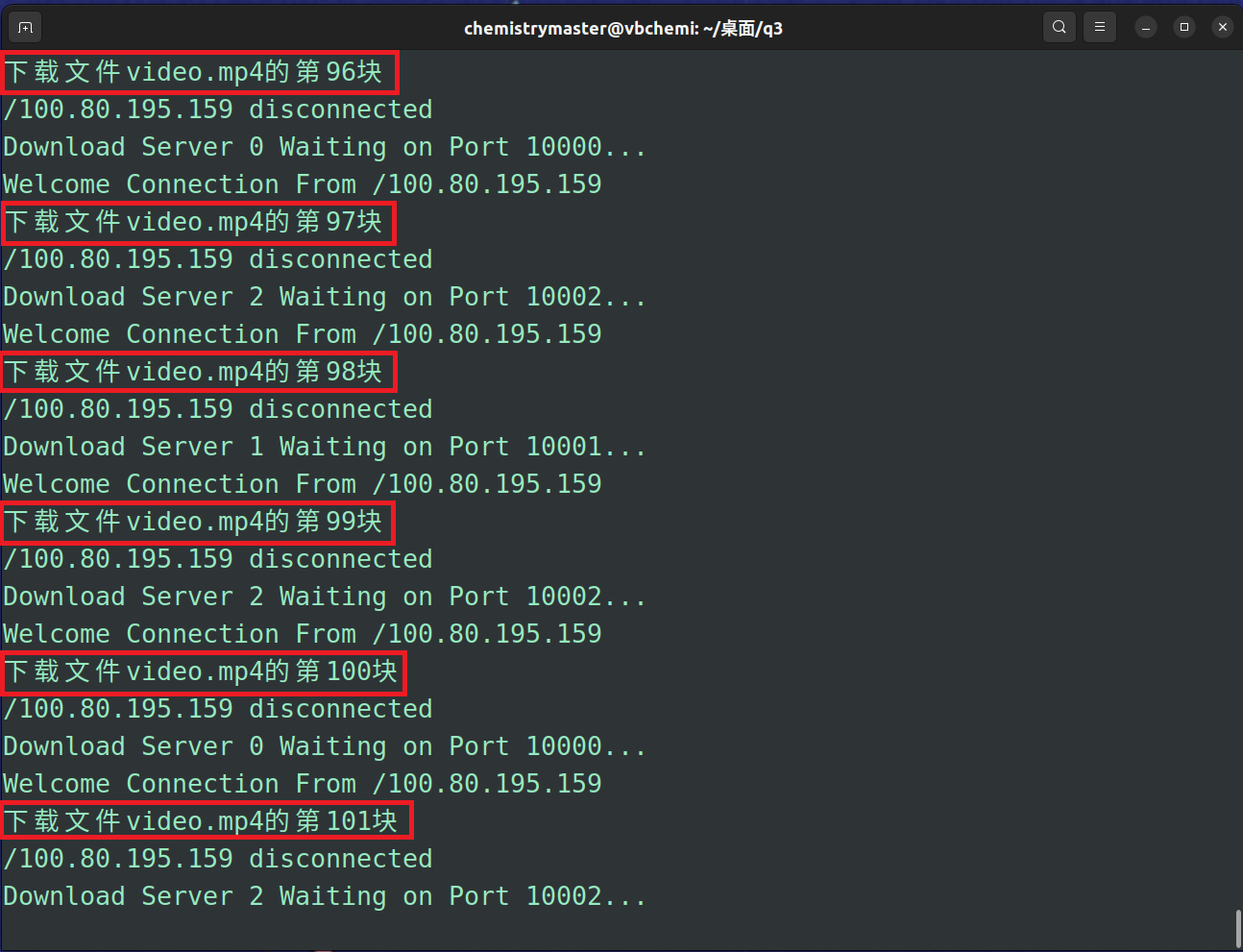
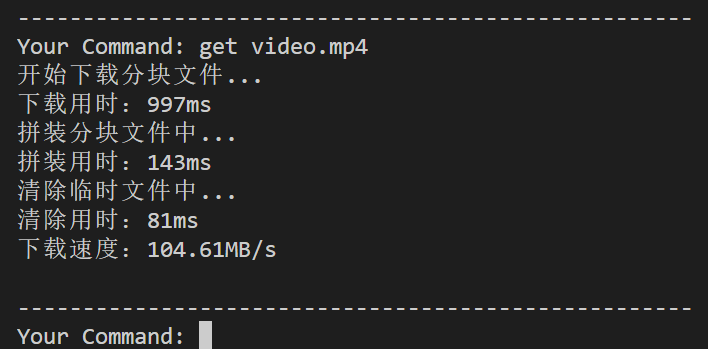


**2.get**

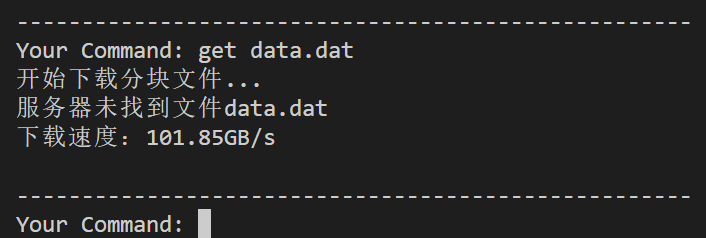
2.1 下载paint.png



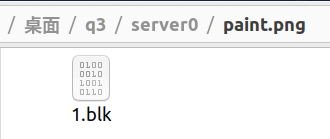
2.2 下载video.png

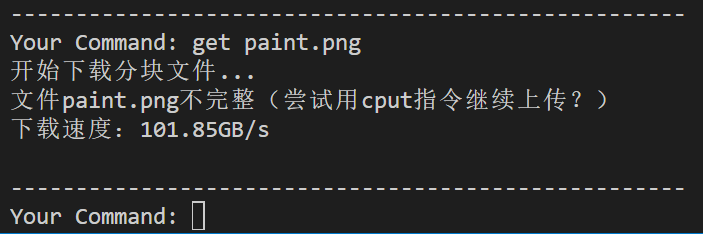


2.3 下载不存在的文件（会打印提示信息）



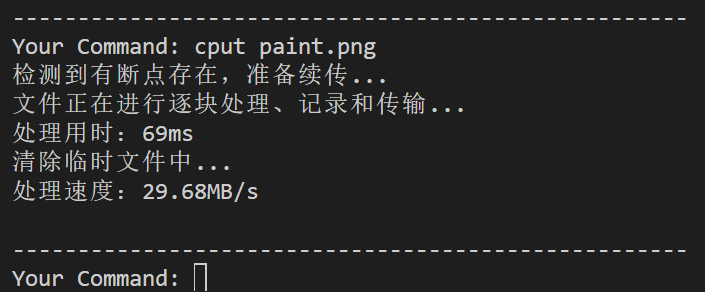
2.4 下载不完整的文件（此时paint.png缺失第2块和第3块，且映射文件中也没有这两块的存储信息，会打印提示信息）

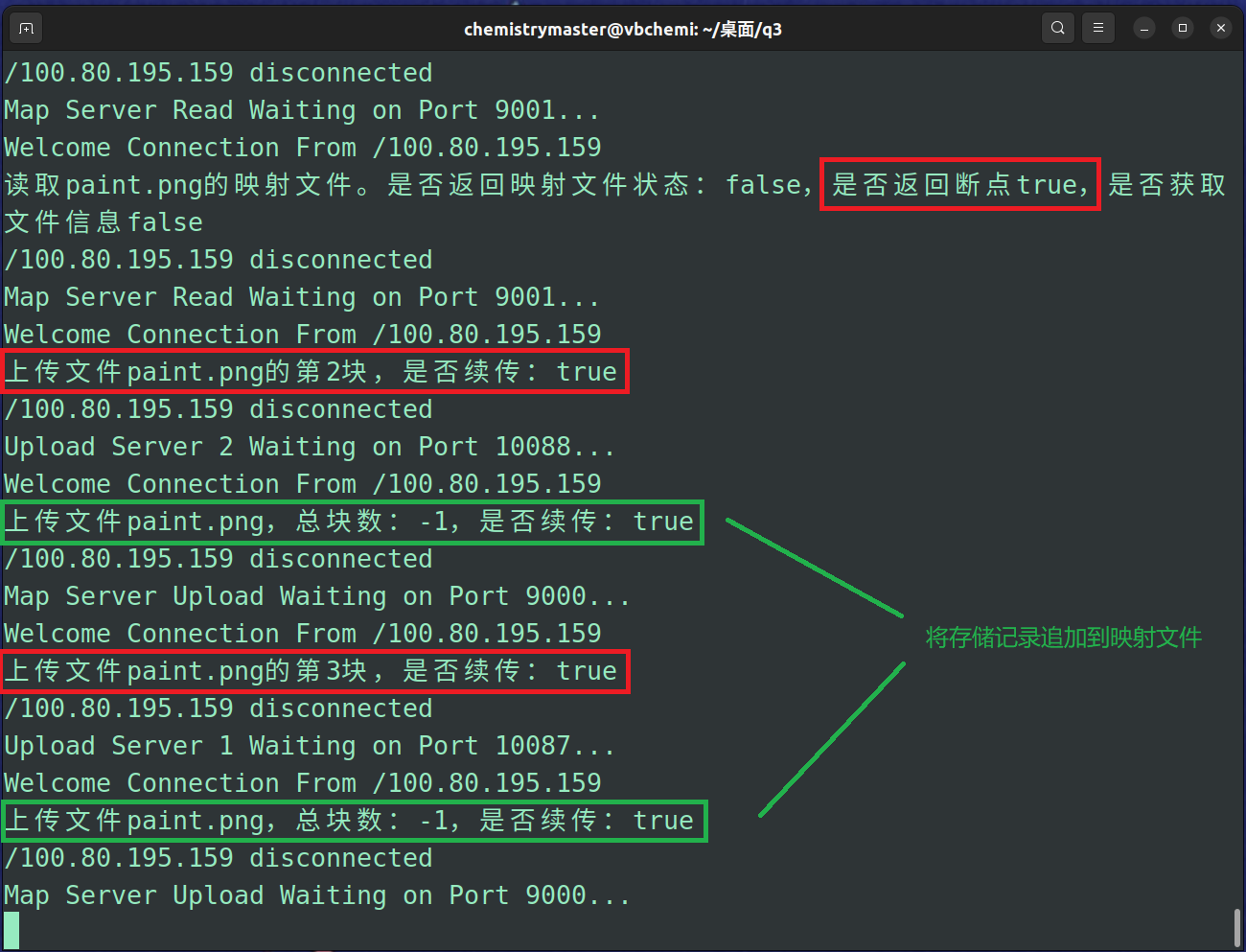




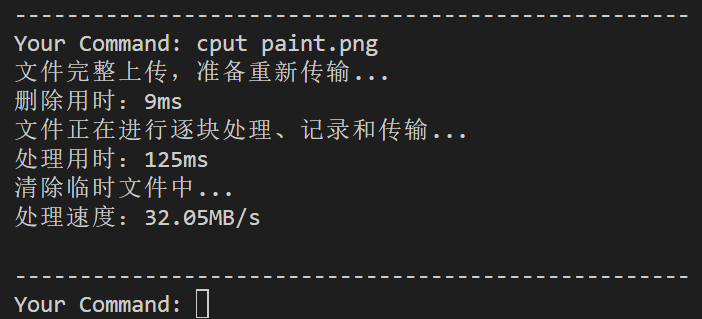
**3.cput**

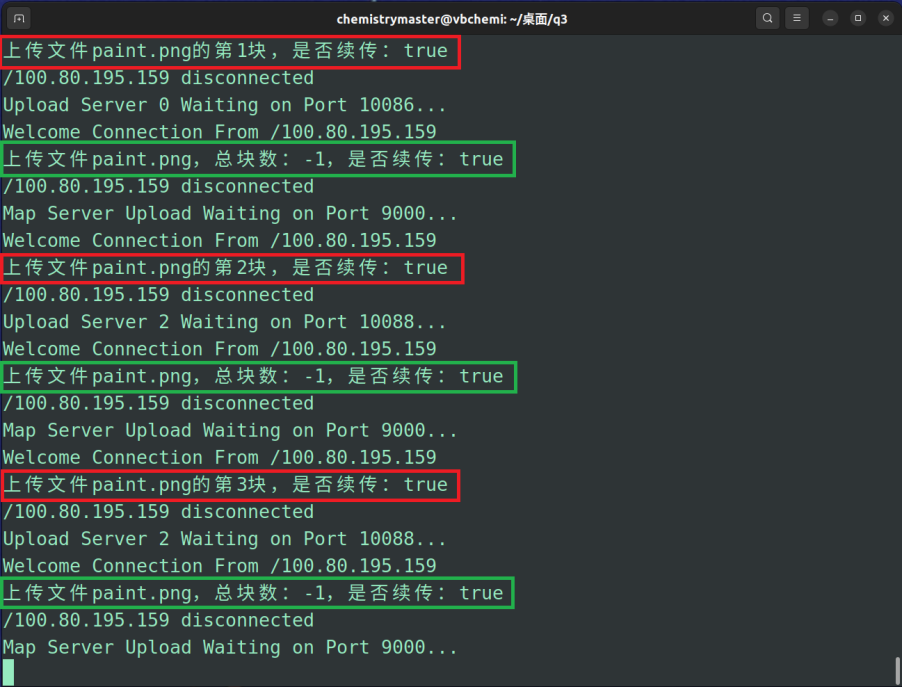
3.1 续传不完整的paint.png





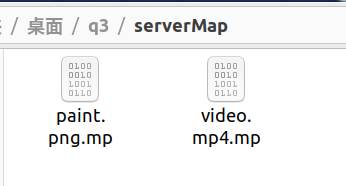
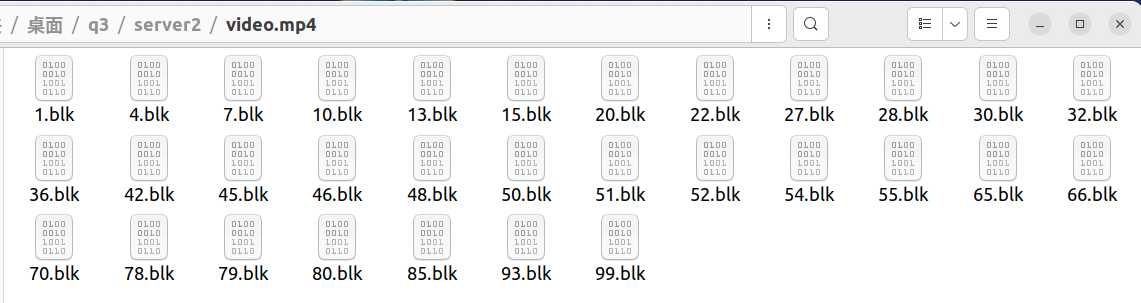
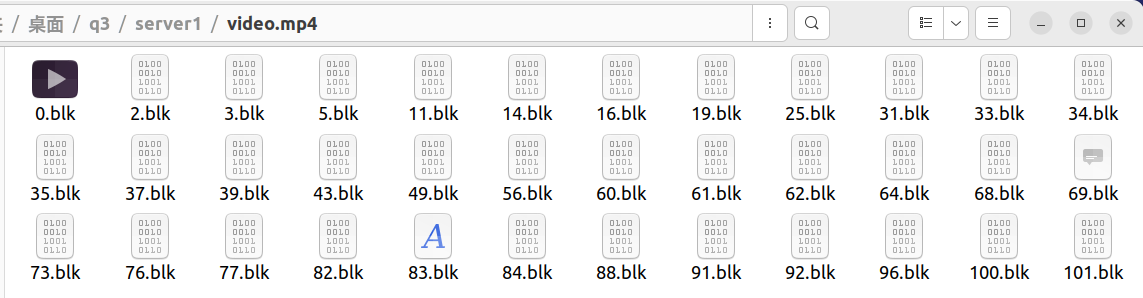
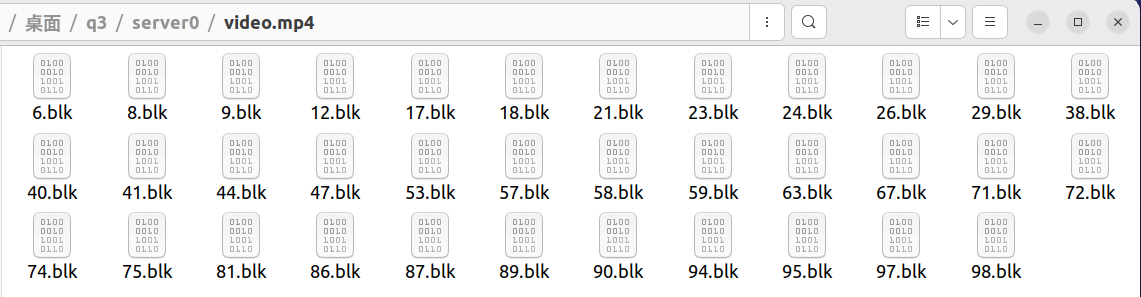
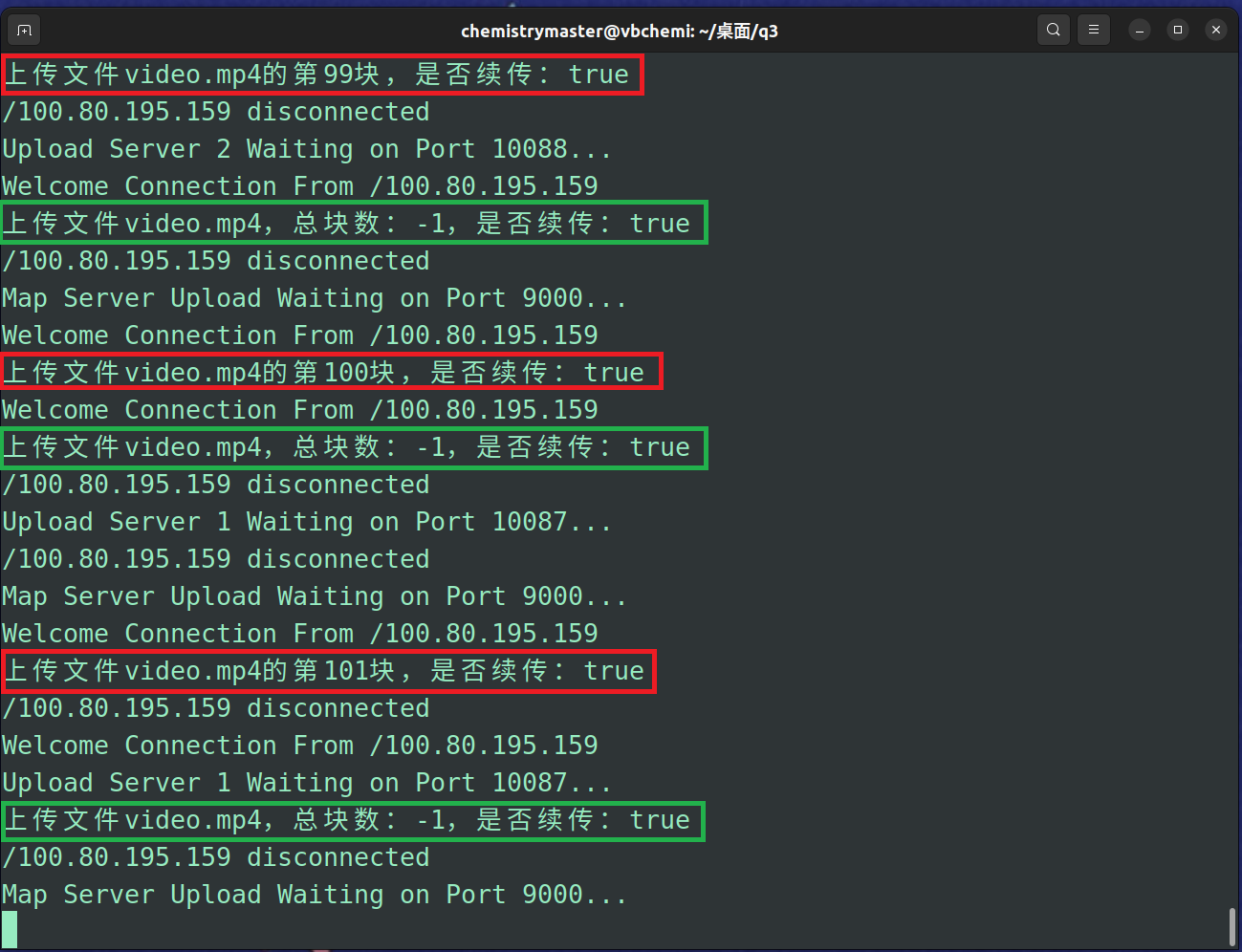
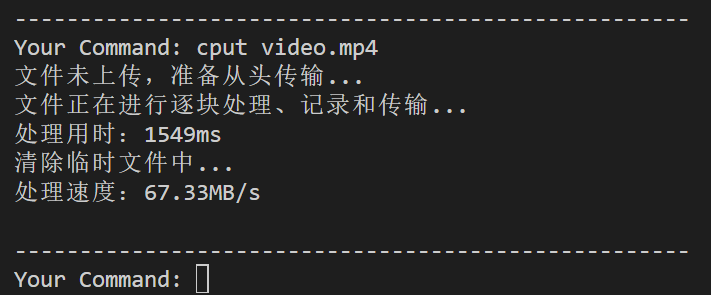
3.2 续传本就完整的文件（删除原有文件，并重新上传）



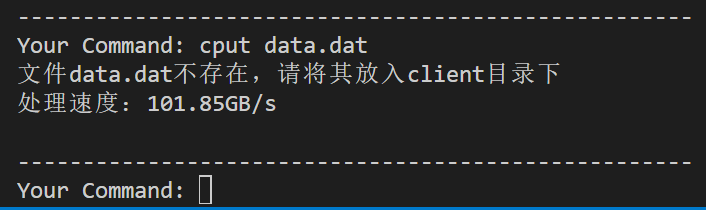




3.3 续传还没上传的文件（这里删除了原先上传的video.mp4，会正常上传）



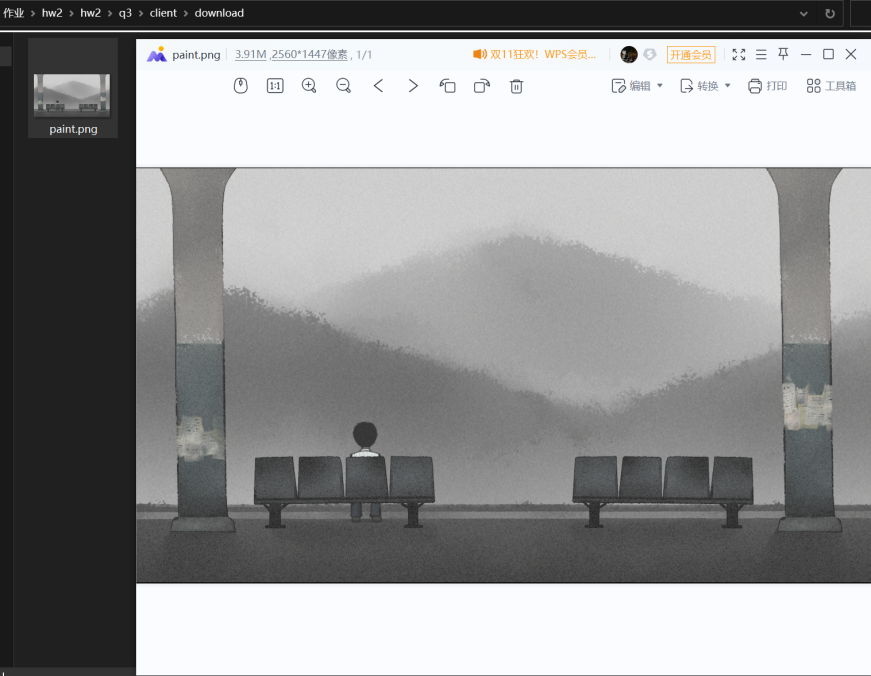
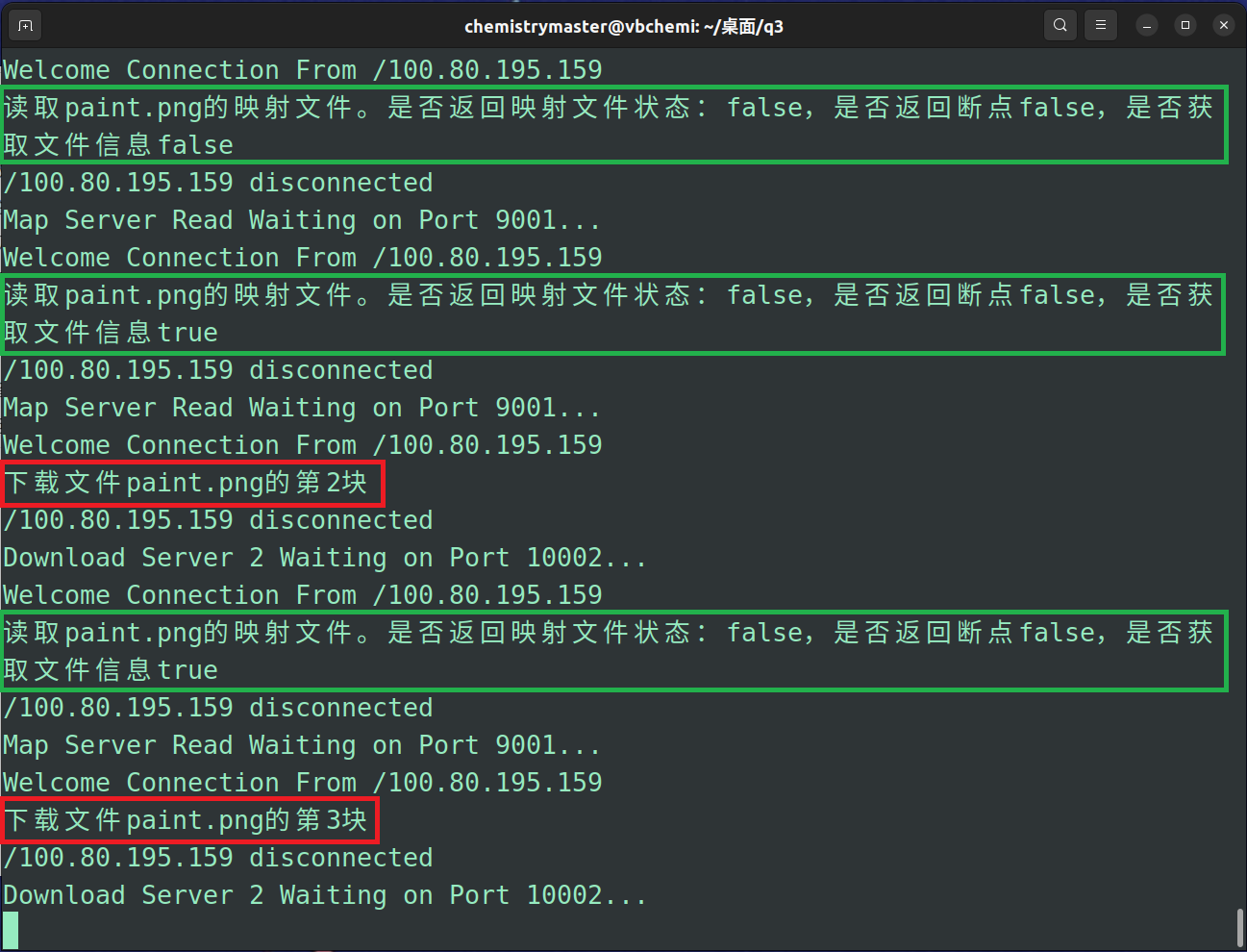
3.4 续传不存在的文件（会打印提示信息）



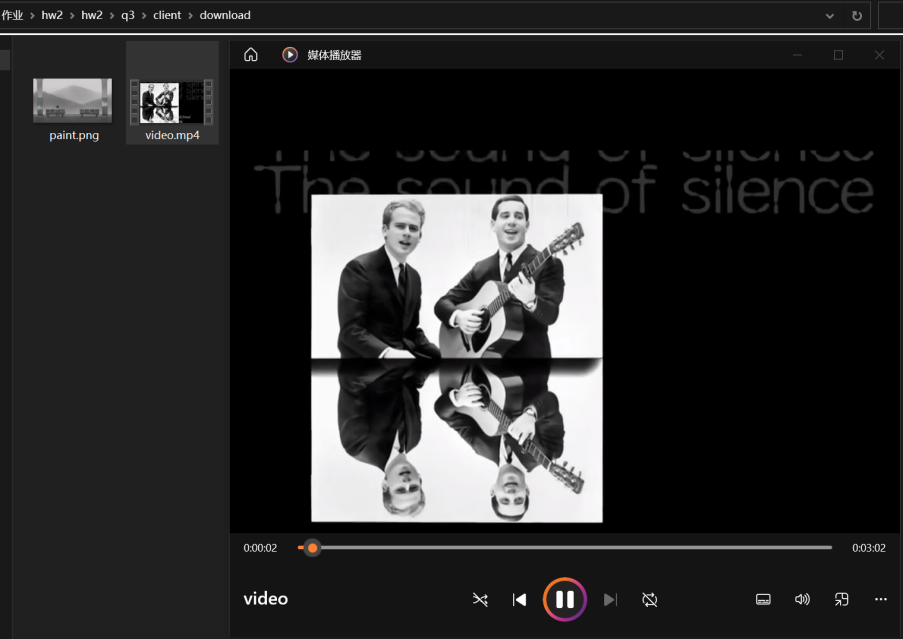
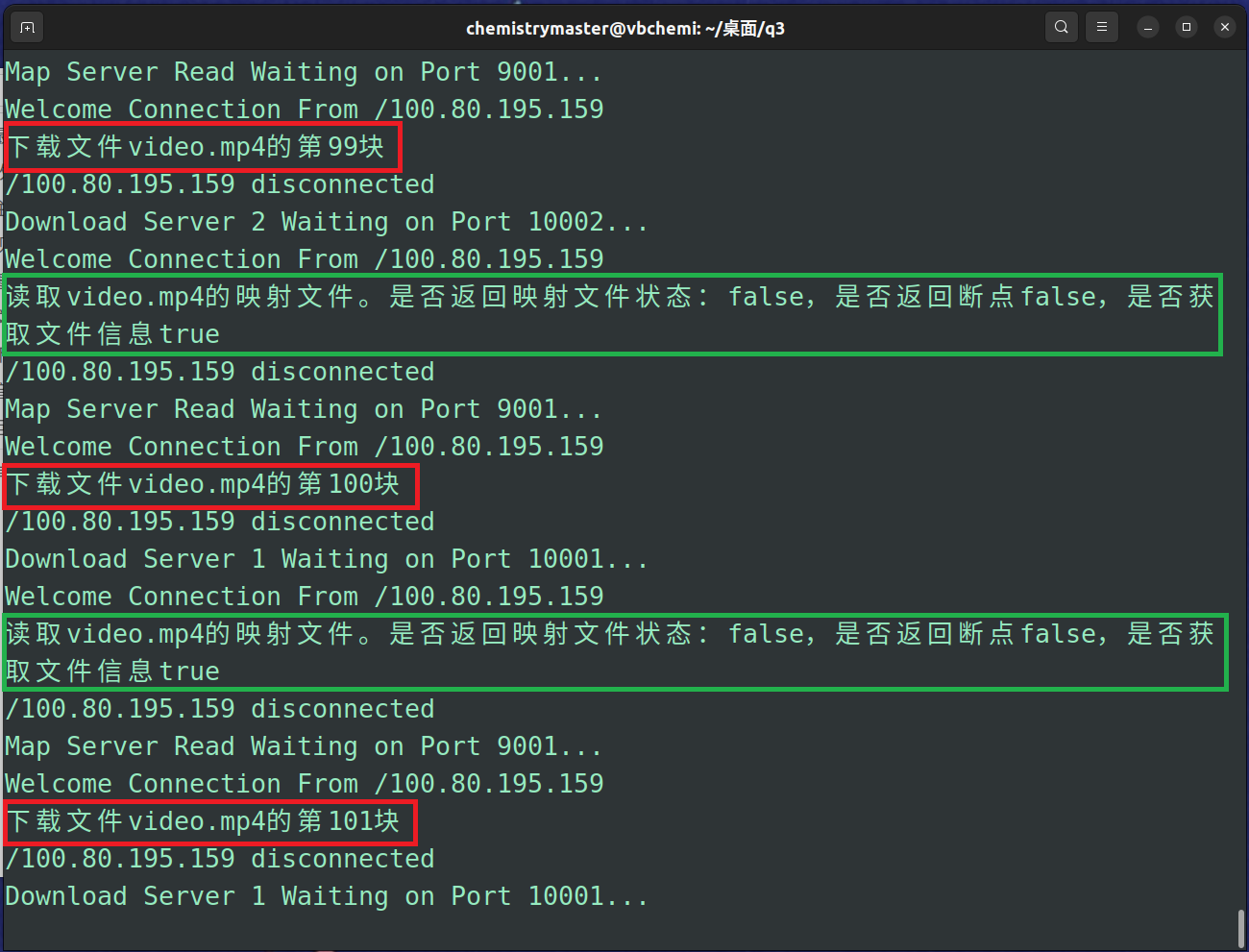
**4.cget**

4.1 续传不完整的文件（下载时，客户端download目录下会生成一个文件名.temp文件夹，里面存放下载的块文件，等下载完毕后拼装成原文件，再将这个临时目录及里面的内容删除。此时客户端只有paint.png的第0块和第1块）

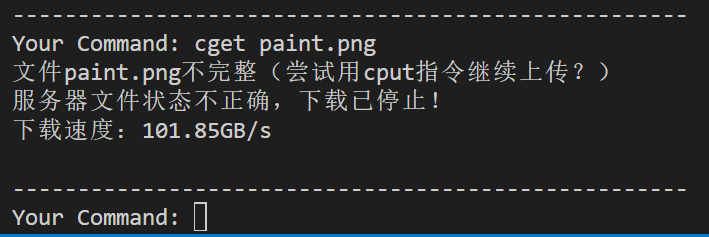




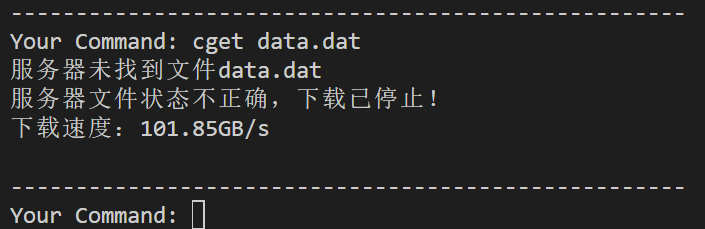
4.2 续传一个未下载的文件（会从头下载）



4.3 续传一个不完整的文件（此时paint.png缺失第2块和第3块，且映射文件中也没有这两块的存储信息，会打印提示信息）

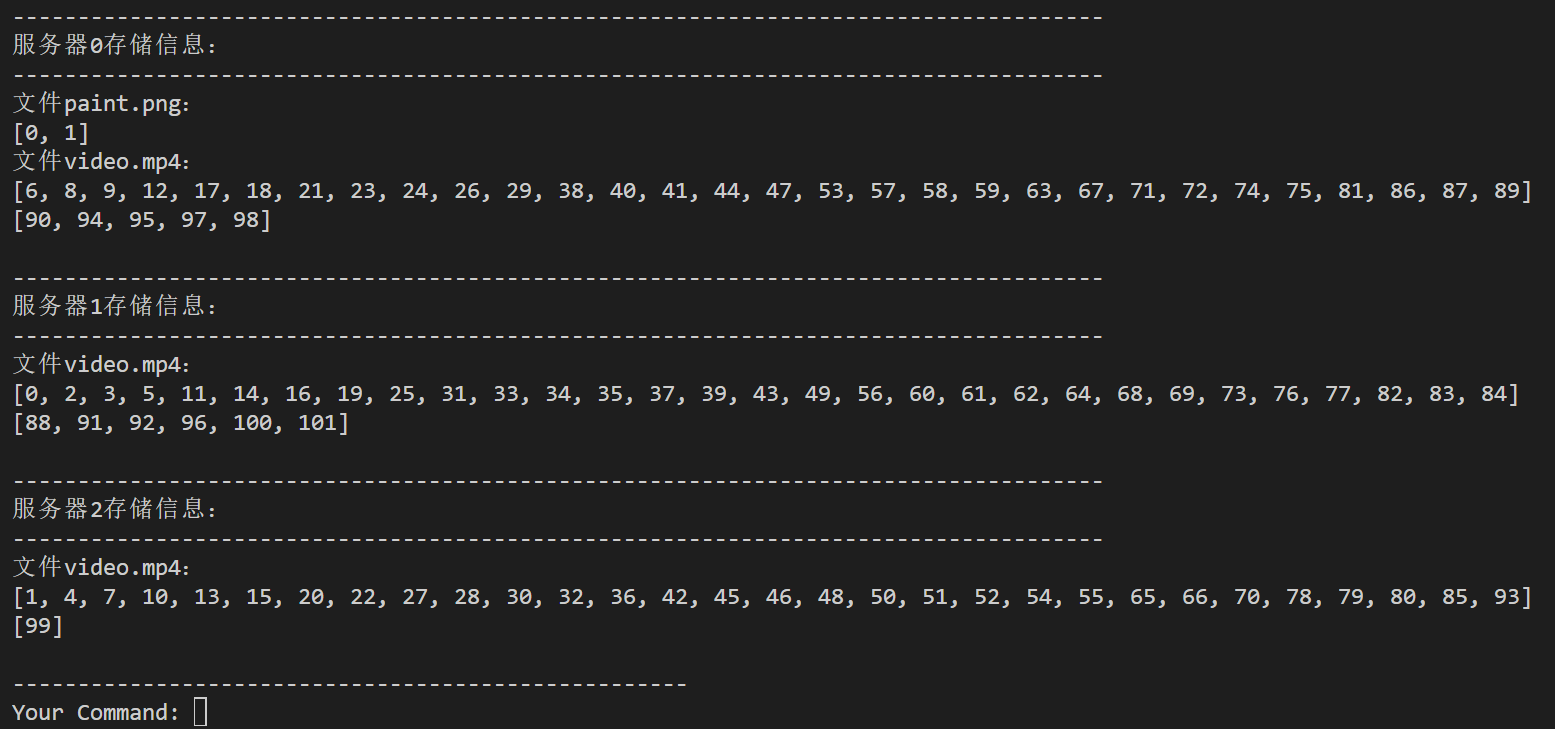
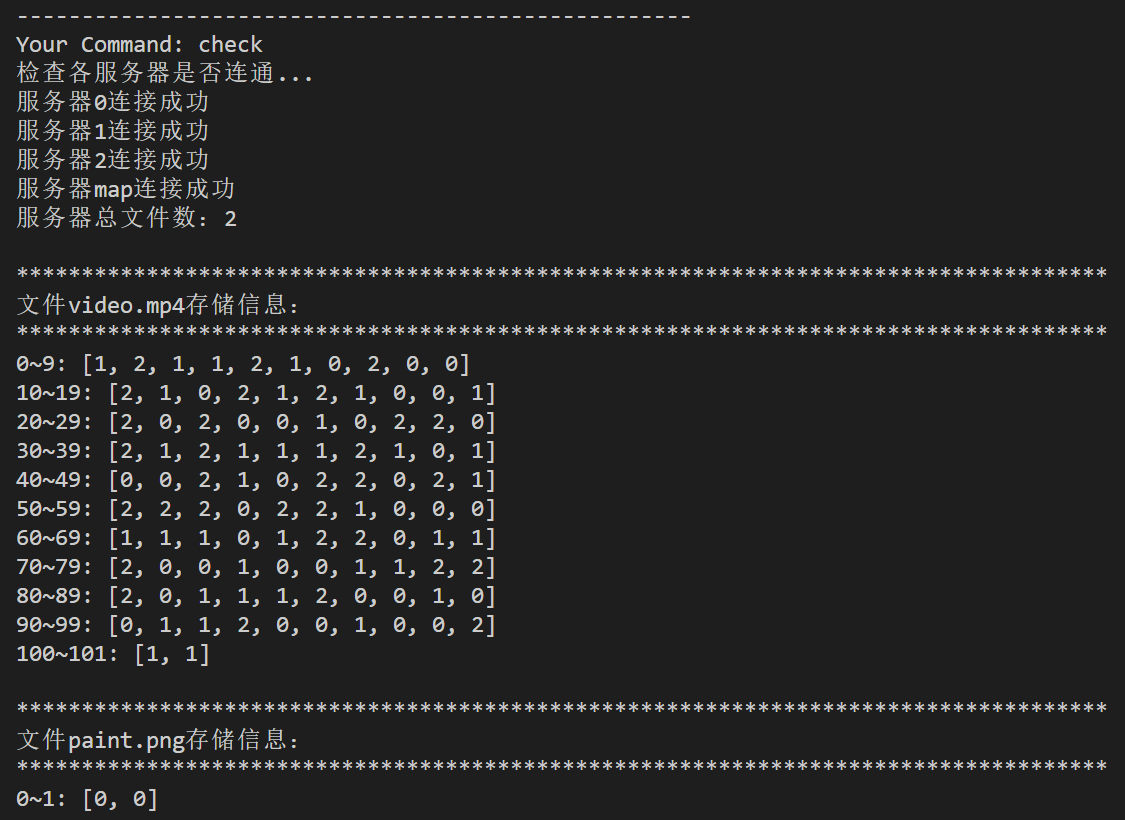


4.4 下载不存在的文件（会打印提示信息）

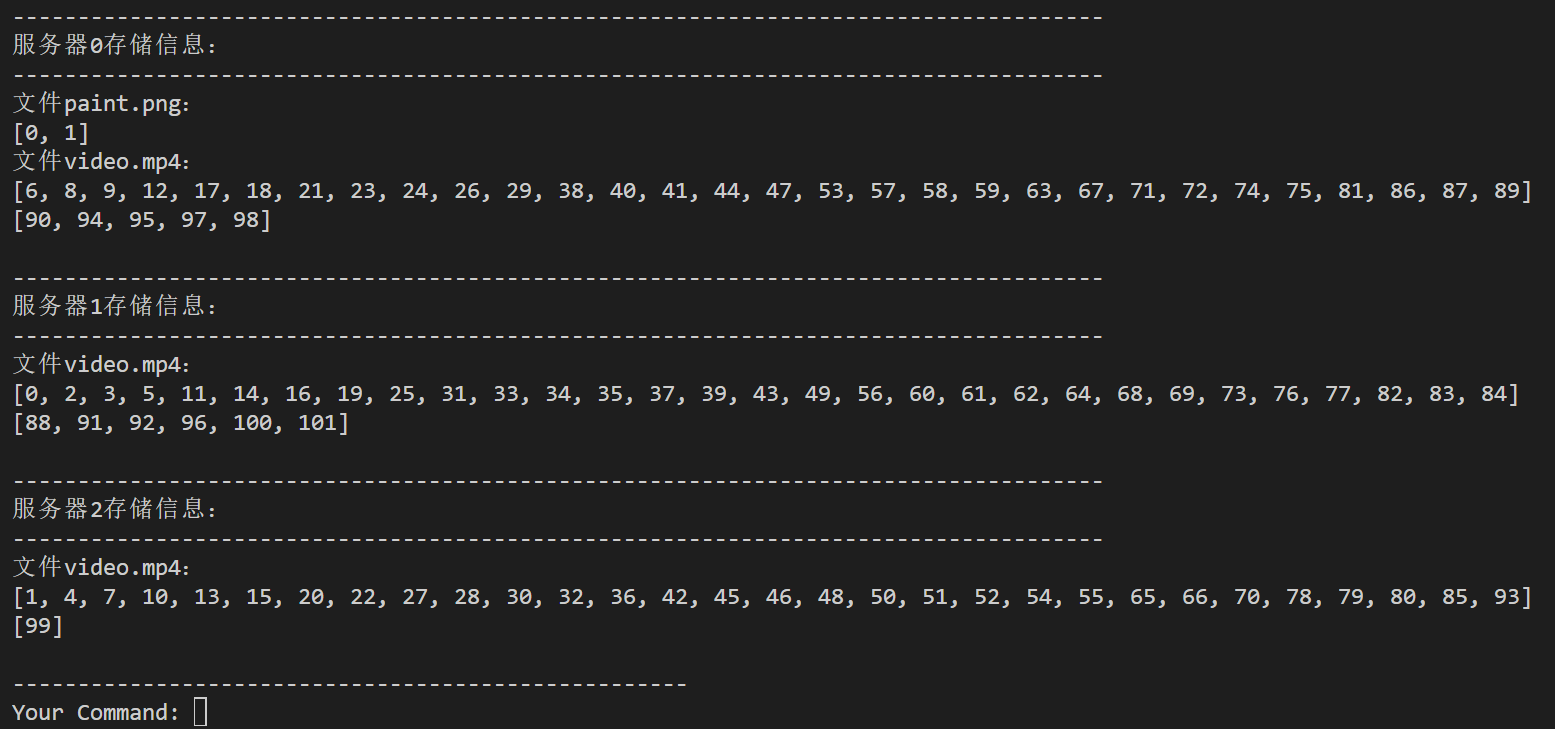


**5.check**

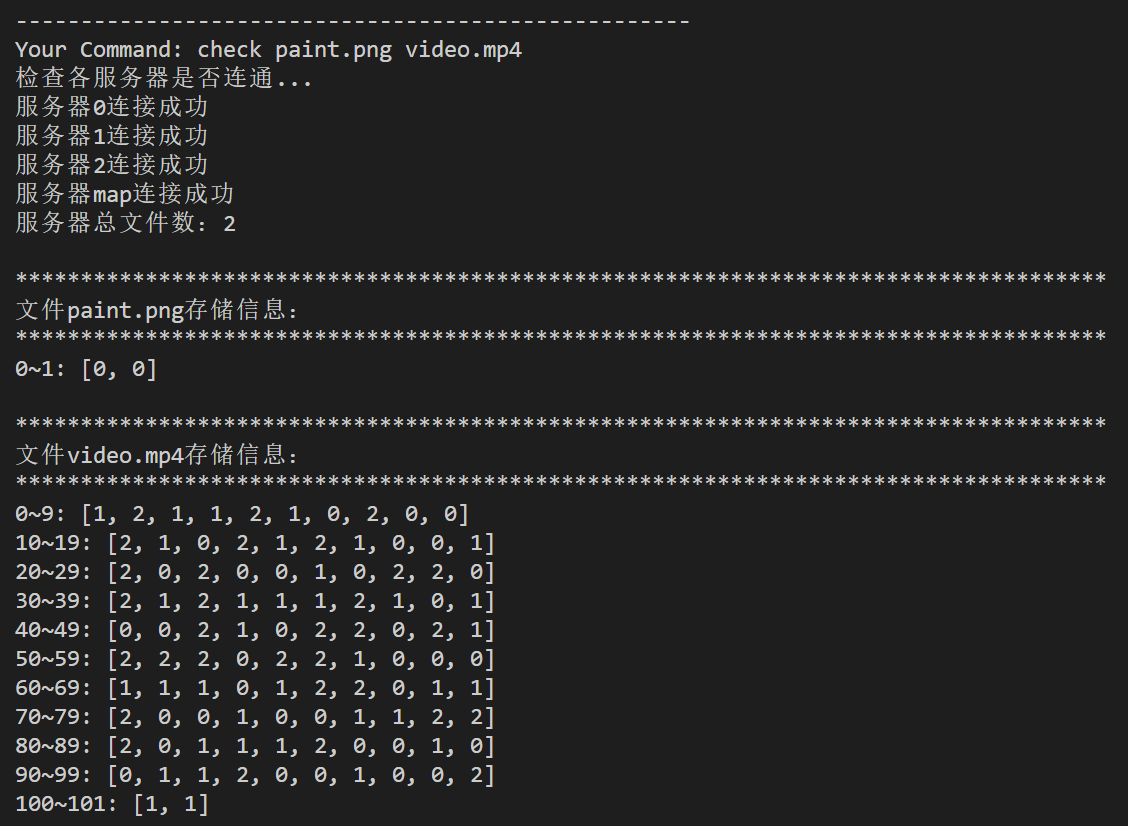
5.1 无参数（测试连接、显示文件映射信息、显示各服务器存储信息。其中，文件映射信息展示了文件各块分别位于哪一个服务器（一行有10块的信息），服务器存储信息会显示存了哪些文件的哪些块（一行有30块的信息））

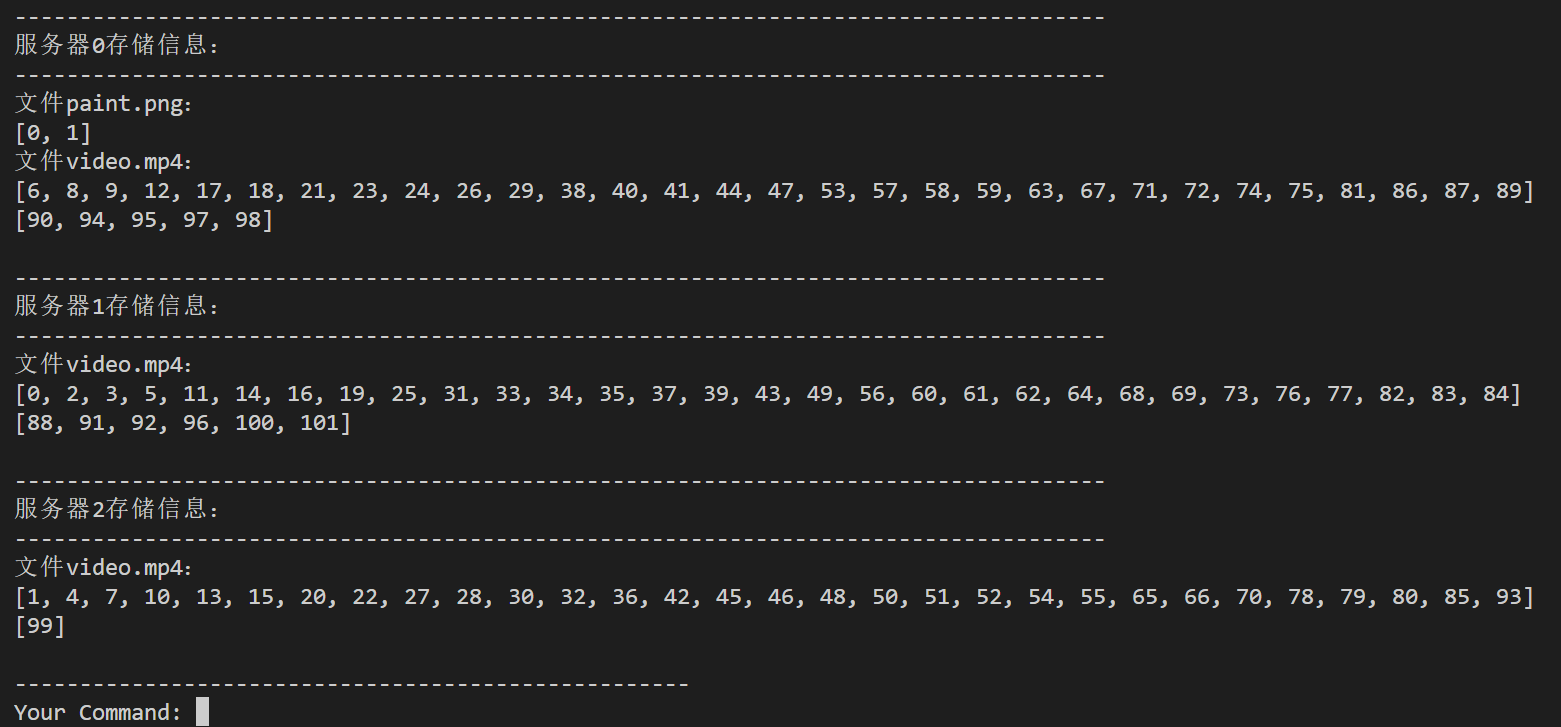


5.2 单个参数（只显示video.mp4的信息）

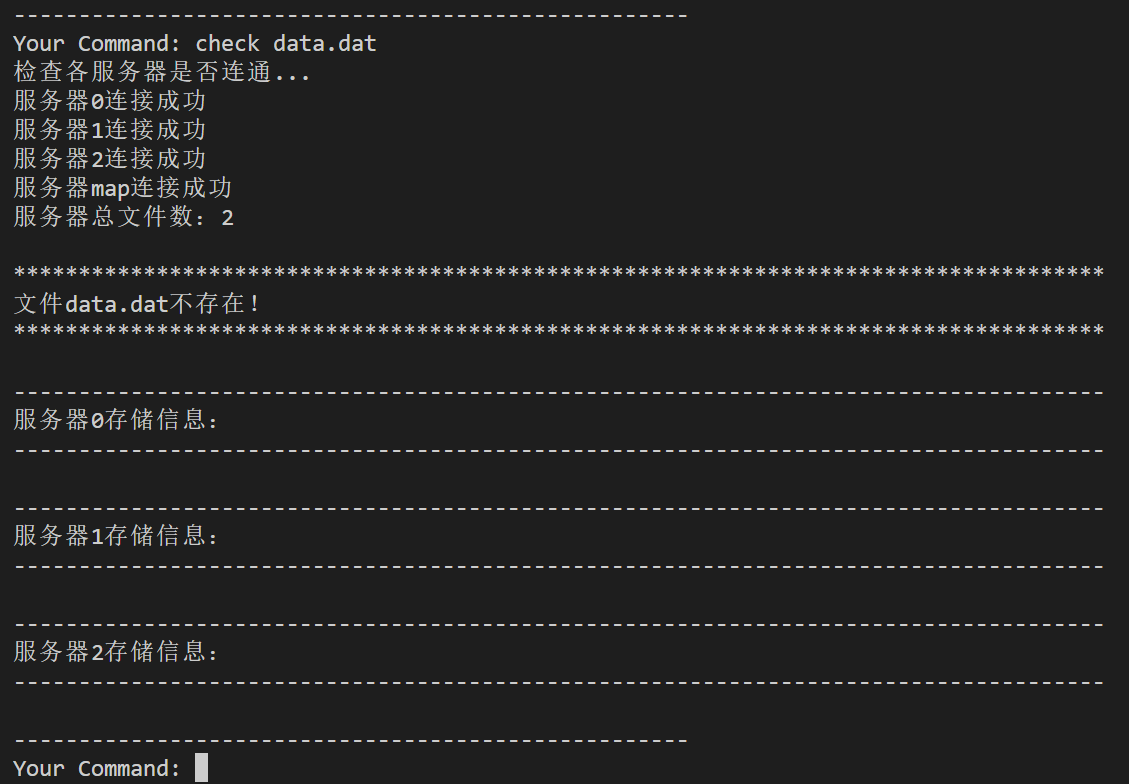


5.3 多个参数（显示paint.png和video.mp4的信息）

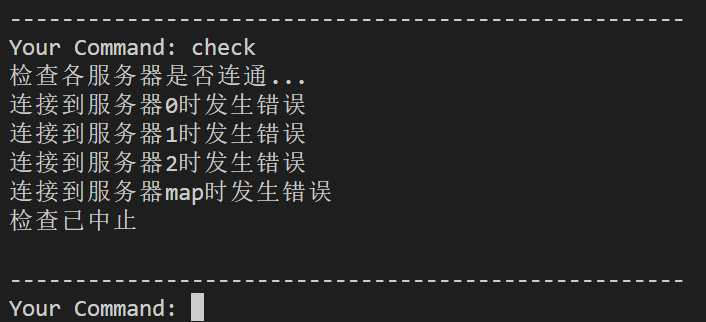




5.4 参数中出现了服务器上没有的文件（）

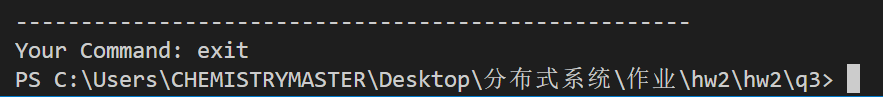


5.5 服务器无法连接时



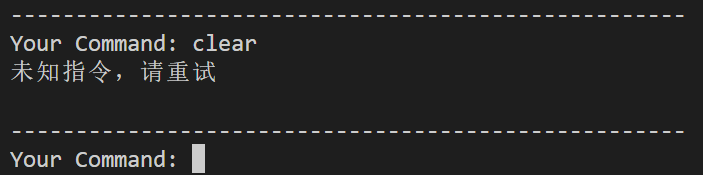
**6.exit**

退出客户端程序



**7.未知指令**

会打印提示信息



**补充**

**1.映射文件的结构**

映射文件是一个二进制文件，文件头为4字节，表示原文件的块数。后续每8个字节中，4个字节为块号，4个字节为所在服务器号。如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 总块数 | 块号 | 服务器号 | 块号 | 服务器号 |
| 块号 | 服务器号 | 块号 | 服务器号 | ...... |

例如，paint.png有4块，分别存储在服务器0、2、1、1，那么映射文件如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 0100 | 0000 0000 | 0000 0000 | 0000 0001 | 0000 0010 |
| 0000 0010 | 0000 0001 | 0000 0011 | 0000 0001 | × |

而一个映射文件就表示了一个文件在服务器上的存储情况。例如上面的映射文件会被命名为paint.png.mp，存储在map服务器的目录下。当有多个文件存储在服务器上时，map服务器目录下就会有多个.mp文件。

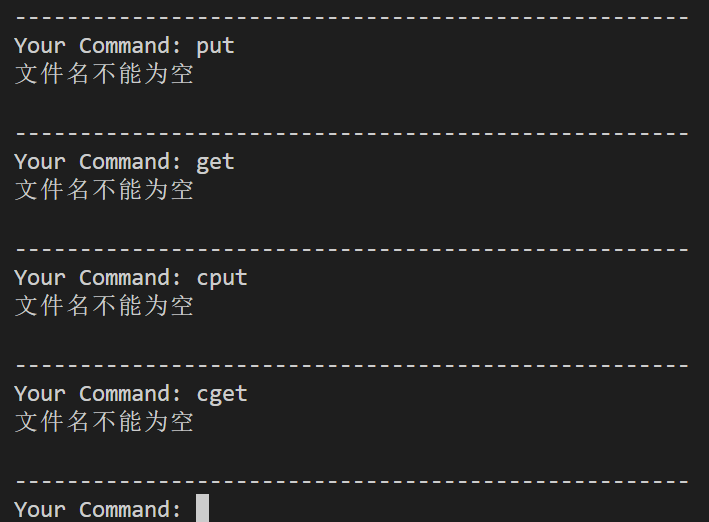
**2.断点模拟**

1.在上传过程中，中止服务器端程序

2.直接删除块文件，并将映射文件中删除块的块号和服务器号删除（由于顺序上传，只能删除从末尾开始连续的块及其存储信息，这与实际情况（方法1）情况一致）

**3.基本功能缺失文件名的情况**

会打印提示信息，且不做任何额外操作



**4.关于jar包**

用java19编译，生成版本也为java19