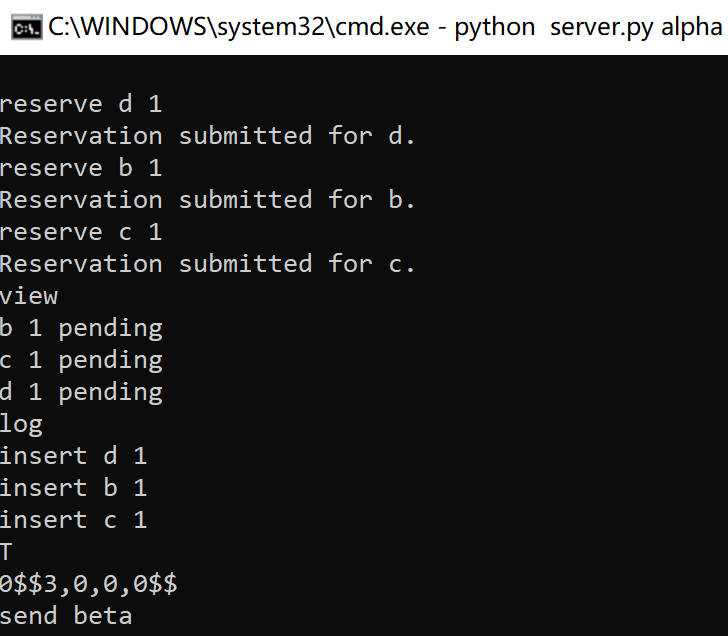
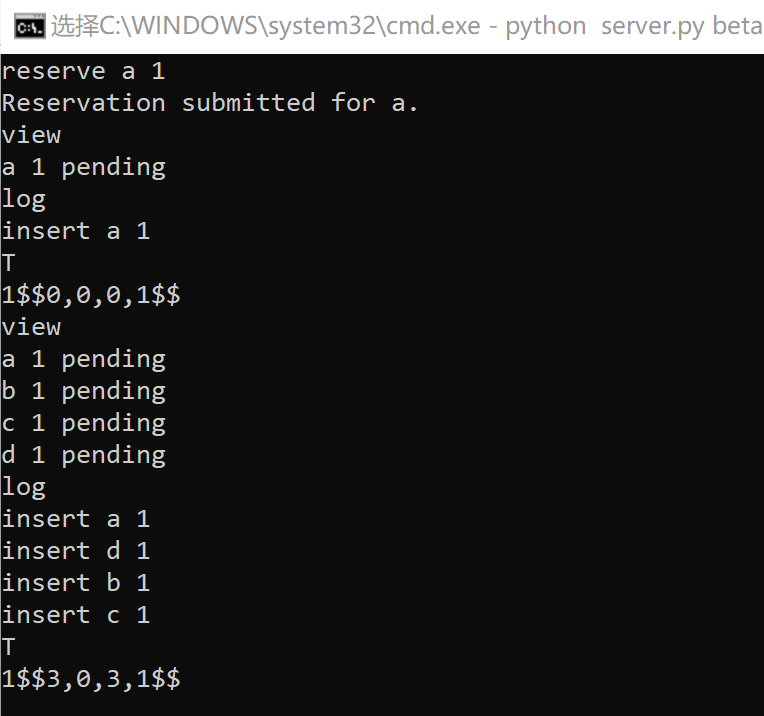
Design and Implementation:

For each site in our system, we start a thread to keep listening on the UDP port to see if there is any message sending to the current site. The main function includes a while loop to get the input and send the messages. We used a python dictionary to store the local copy of reservation list, the key is the client name and the value is the list of the flights. The log is a python list to keep track of the events and the operations(insert/delete) of the events.

Description of what events on send and sendall.

When we get a input ‘send <sent\_id>’, we loop through the log of current site, for each event R in the log, we will add that event in our sent message if the current site knows that current site knows the site of sent\_id does not know the event R. It will also include the current site id and the T matrix in the message that will be sent.

**How to determine reservation to be confirmed:**

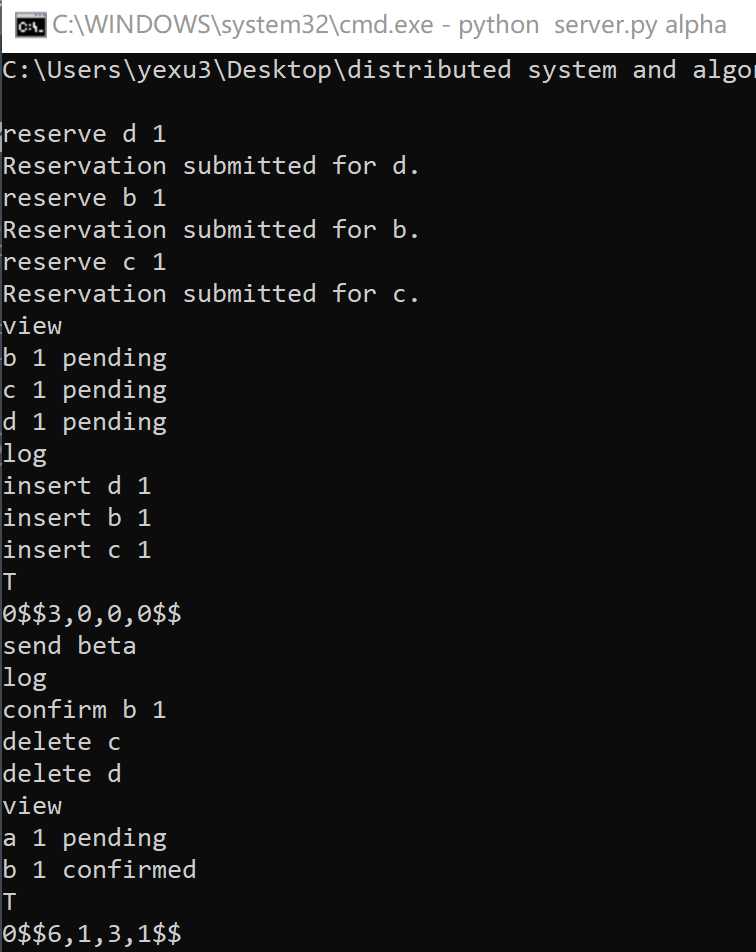
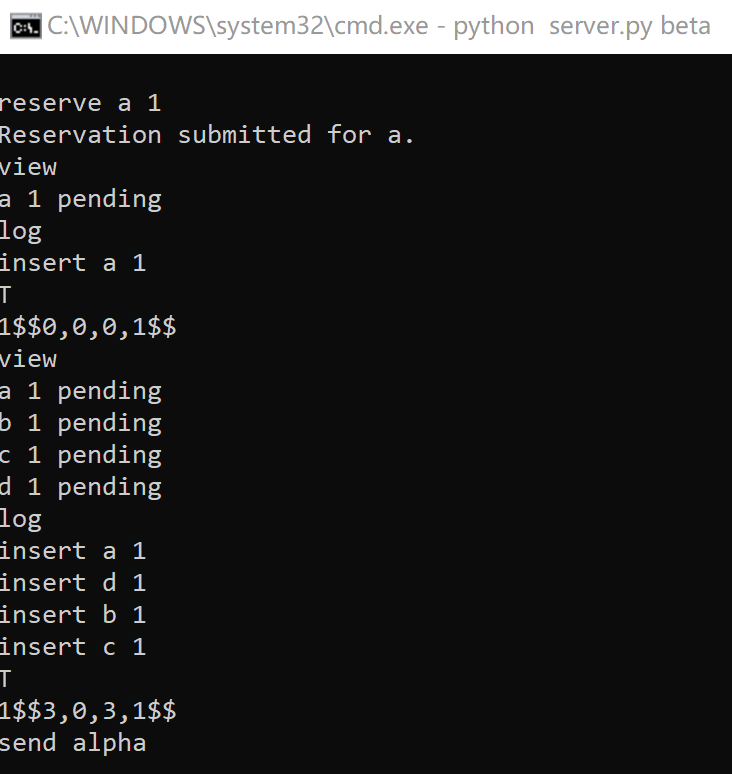
**1.** We have four steps to confirm one reservation: (1) this process which reserve a seat should send its PL and T to all the other process, and get all the other processes massages back. (2) after sending this reservation and getting massages back from all the other processes, the process could decide whether to confirm this reservation or cancel it, because at that time, the process finally know whether the seats are available or the seats has been taken by other processes. (3) before we get feedback from all the other processes, all the process reserved about all the flight will still be in the partial log and all the reserved information will be in dictionary with status of “pending”. (4) after we decide whether confirm the reservation or cancel the reservation, we will build a new event and put the new confirm or delete event in partial log.

Send

Send

3, 0 3, 0

0, 0 3, 1



Send

Send

6, 1 3, 0

3, 1 3, 1

**2.** The reservation could be confirmed under either of two conditions: (1) no other processes have taken the seats, and the seats are available now. (2) our clients for the seats are alphabetically smaller than the other processes who have taken the seats without confirming their reservation.

**How to test projects(A description of how you tested your project (be sure to consider failures, recovery, and message loss).**

**1.** Corner test: over reserve the seats on one flight by two different process, and keep sending massage between these two process until they get the same dictionary.

你结了很多图，贴一组比较trick的进来。然后我把这个图改成case。

**2.** Failures test:这部分方程还没写，如果需要，这个方程我可以写，写好了我贴给你。应该很快的。

**Design:** We design a check\_failure() function, in which the site will send a message “checking, site\_id ” to every site. Whenever one site has receive a message start with “checking”, the site will send its own site\_id back to check\_failure() function calling process.

**Test:** I implement 3 process, and site alpha reserve one event and send it to all the other process. Then I shut alpha down and I let site beta to implement check\_failure() function and I will find site alpha didn’t send site\_id back.

写好之后，贴一个图上去

**3.** recovery test: 这部分我忽然有这么个想法，我们现在paper里写我们说的那个特别复杂的算法。就存字符串的算法。但我们暂时别实现。因为submitty上暂时也不测这个。等到我们要去TA约的 时候，我们再把这个东西做好，这样我们就还有好几天。我觉得TA不至于让我们当场下载文件当场给他跑对吧。肯定是我们拿着我们的笔记本上去找他对吧。如果他不问我们就不说，他问我们就说当时做的时候有个小bug，小case没问题，大case保存会出错。