Example 1. This example gives sample output of running the server, client 1, and client 2 in this order (from three different terminal windows on the same lab workstation). The input file `a2p2-ex1.dat` contains the following lines.

```
# Transactions file for 2 clients
#    a2p2 -s
#    a2p2 -c 1 a2p2-ex1.dat
#    a2p2 -c 2 a2p2-ex1.dat

1 gtime
1 put   index1.html
1 put   img1.jpg
1 put   video1.mp4
1 delay 2500

2 gtime
2 put   index2.html
2 put   img2.jpg
2 put   video2.mp4
2 delay 1500

1 gtime
1 get   index2.html
1 get   index3.html
1 delete img2.jpg
1 delay  3000

1 gtime
1 quit

2 gtime
2 quit
```

Sample output (edited for clarity):

- Client 1 output:

```
main: do_client (idNumber= 1, inputFile= a2p2-ex1.dat)

Transmitted (src= client:1) GTIME
Received (src= server) (TIME:    1.52)

Transmitted (src= client:1) (PUT: img1.jpg)
Received (src= server) OK

Transmitted (src= client:1) (PUT: video1.mp4)
Received (src= server) OK

*** Entering a delay period of 2500 msec
*** Exiting delay period

Transmitted (src= client:1) GTIME
Received (src= server) (TIME:    4.39)

Transmitted (src= client:1) (GET: index2.html)
Received (src= server) OK

Transmitted (src= client:1) (GET: index3.html)
Received (src= server) (ERROR: object not found)
```

```
Transmitted (src= client:1) (DELETE: img2.jpg)
Received (src= server) (ERROR: client not owner)

*** Entering a delay period of 3000 msec
*** Exiting delay period

Transmitted (src= client:1) GTIME
Received (src= server) (TIME:    7.57)
```

- Client 2 output:

```
main: do_client (idNumber= 2, inputFile= a2p2-ex1.dat)

Transmitted (src= client:2) GTIME
Received (src= server) (TIME:    2.89)

Transmitted (src= client:2) (PUT: index2.html)
Received (src= server) OK

Transmitted (src= client:2) (PUT: img2.jpg)
Received (src= server) OK

Transmitted (src= client:2) (PUT: video2.mp4)
Received (src= server) OK

*** Entering a delay period of 1500 msec
*** Exiting delay period

Transmitted (src= client:2) GTIME
Received (src= server) (TIME:    4.84)
```

- Server's output:

```
a2p2: do_server

Received (src= client:1) GTIME
Transmitted (src= server) (TIME:    1.52)

Received (src= client:1) (PUT: index1.html)
Transmitted (src= server) OK

Received (src= client:1) (PUT: img1.jpg)
Transmitted (src= server) OK

Received (src= client:1) (PUT: video1.mp4)
Transmitted (src= server) OK

Received (src= client:2) GTIME
Transmitted (src= server) (TIME:    2.89)

Received (src= client:2) (PUT: index2.html)
Transmitted (src= server) OK

Received (src= client:2) (PUT: img2.jpg)
Transmitted (src= server) OK

Received (src= client:2) (PUT: video2.mp4)
Transmitted (src= server) OK

Received (src= client:1) GTIME
Transmitted (src= server) (TIME:    4.39)
```

```
Received (src= client:1) (GET: index2.html)
Transmitted (src= server) OK

Received (src= client:1) (GET: index3.html)
Transmitted (src= server) (ERROR: object not found)

Received (src= client:1) (DELETE: img2.jpg)
Transmitted (src= server) (ERROR: client not owner)

Received (src= client:2) GTIME
Transmitted (src= server) (TIME:    4.84)

Received (src= client:1) GTIME
Transmitted (src= server) (TIME:    7.57)
list
Object table:
(owner= 1, name= index1.html)
(owner= 1, name= img1.jpg)
(owner= 1, name= video1.mp4)
(owner= 2, name= index2.html)
(owner= 2, name= img2.jpg)
(owner= 2, name= video2.mp4)

quit
quitting
```

Important: Don't forget to cleanup the running processes by using the command
`pkill -u $USER pattern`
where `pattern` can be a string that appears in the processes you would like to terminate, e.g., `pkill -u $USER clock`.