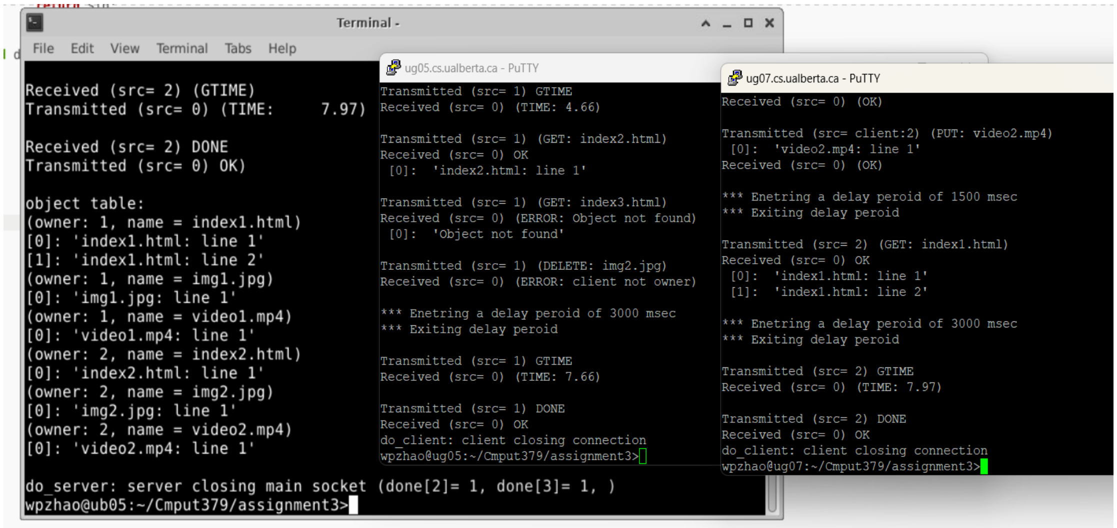
Cmput 379 Project Report

- **Objectives**: state the project objectives and value from your point of view (which may be different from the one mentioned above)
    - The objective of this project is to help us learn how to use TCP sockets for communication over the Internet and create a client-server program using such for communication. By implementing this project, we can gain a better understanding of how these concepts work and how they differ from the previous assignment. Through solving the different issues occurring we can observe these concepts in much greater detail.
- **Design Overview:** highlight in point-form the important features of your design
    - Enums: The program defines an enumerated data type PACKETTYPE that contains several packet types such as PUT, GET, DELETE, GTIME, TIME, OK, ERROR, HELLO and QUIT.
    - Structures: The program defines a structure named PACKET that contains three members: type, client_id, and message, fileline_count, and file_content. The member type is of type PACKETTYPE, client_id is an integer, and message is an array of characters, fileline_count is to count the number of file lines, and file_content is a array to store all of the line content of the file to be send.
    - Functions: The program defines several functions such as print_packet(), send_packet(), rcv_packet(), clientConnect(), serverListen(), do_server() and do_client(). The print_packet() function prints the contents of a PACKET structure, send_packet() sends a PACKET to a given file descriptor, rcv_packet() receives a PACKET from a given file descriptor, and do_client() executes the client side of the program. clientConnect() will connect the client to the server, and serverListen() is for the server to listen for the client connections.
    - In the do_client() it will connect to the server first and process the file entered line by line and send a packet of the commands with available contents the client wants to send to the server if it matches the client id. The packet will be different types depending on the commands it wants to send.
    - The do_server() has two loops, one infinite loop until all of the clients quit or the user types in quit, and another loop that loops through all of the clients to server client server socket connections to pull and see if there is any that wants to PULLIN and will execute from there
        - Additionally the data is being stored in 4 arrays, one 3d array for all of the data names, one 1d array for the number of data the client sends for each client, one 4d array for all of the file contents and another 2d array for the line count for the file contents.
        - The put, get and delete will use these to do its functionalities.

- **Project Status:** describe the status of your project (to what degree the programs work as specified, e.g., tested and working, working with some known issues, etc.) mention difficulties encountered in the implementation

- The assignment was completed successfully and this is shown through the testing. During the test testing the most difficult issue occurred would be setting up the server connection in the do_server function using sockets. That was the second most challenging, as there were many errors. Another big issue was storing the content of a file through the “put” command of the client, there were 2 extra arrays set up to accommodate this fact and finding the index and getting the correct index for each part of delete put and get was the most challenging issue I faced for the assignment.
- Testing showed that all the get, put, delete, delay, get time, list and quit commands worked as intended on the client's end. On the server's end, all of these commands also worked correctly, and the server was capable of handling up to 3 clients simultaneously. The clients were able to run on different lab machines and connect to the server.

- **Testing and Results:** comment on how you tested your implementation, and discuss the obtained results.
  - The testing was done on the different lab machine with the sample data provided as well as mine own tests which consist of the same commands.

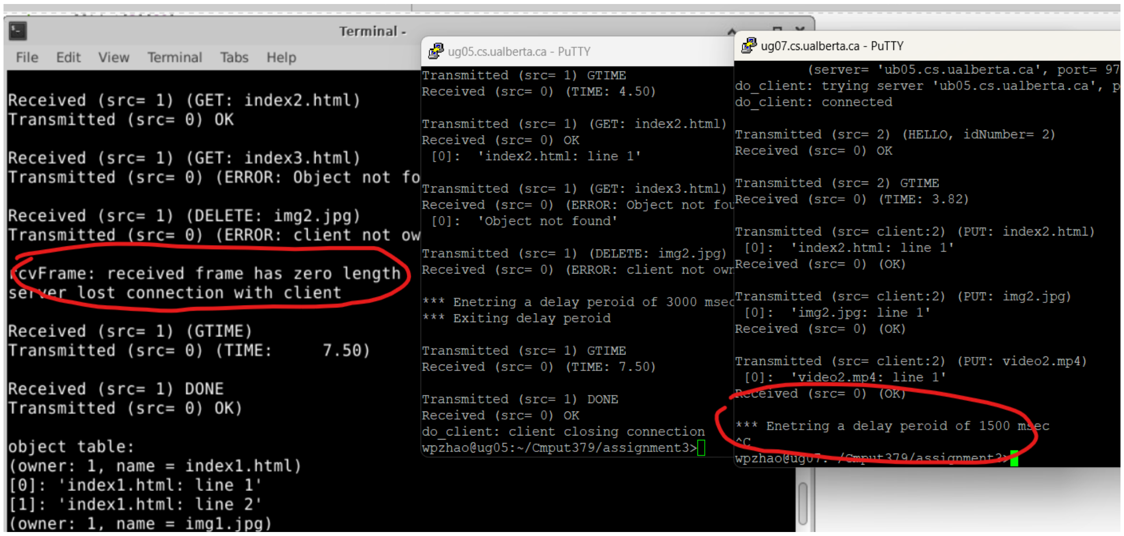


```
Received (src= 2) (GTIME)
Transmitted (src= 0) (TIME:     7.97)

Received (src= 2) DONE
Transmitted (src= 0) OK)

object table:
(owner: 1, name = index1.html)
[0]: 'index1.html: line 1'
[1]: 'index1.html: line 2'
(owner: 1, name = img1.jpg)
[0]: 'img1.jpg: line 1'
(owner: 1, name = video1.mp4)
[0]: 'video1.mp4: line 1'
(owner: 2, name = index2.html)
[0]: 'index2.html: line 1'
(owner: 2, name = img2.jpg)
[0]: 'img2.jpg: line 1'
(owner: 2, name = video2.mp4)
[0]: 'video2.mp4: line 1'

do_server: server closing main socket (done[2]= 1, done[3]= 1, )
wpzhao@ub05:~/Cmput379/assignment3>
```

```
Transmitted (src= 1) GTIME
Received (src= 0) (TIME: 4.66)

Transmitted (src= 1) (GET: index2.html)
Received (src= 0) OK
 [0]:  'index2.html: line 1'

Transmitted (src= 1) (GET: index3.html)
Received (src= 0) (ERROR: Object not found)
 [0]:  'Object not found'

Transmitted (src= 1) (DELETE: img2.jpg)
Received (src= 0) (ERROR: client not owner)

*** Enetring a delay peroid of 3000 msec
*** Exiting delay peroid

Transmitted (src= 1) GTIME
Received (src= 0) (TIME: 7.66)

Transmitted (src= 1) DONE
Received (src= 0) OK
do_client: client closing connection
wpzhao@ug05:~/Cmput379/assignment3>
```

```
Received (src= 0) (OK)

Transmitted (src= client:2) (PUT: video2.mp4)
 [0]:  'video2.mp4: line 1'
Received (src= 0) (OK)

*** Enetring a delay peroid of 1500 msec
*** Exiting delay peroid

Transmitted (src= 2) (GET: index1.html)
Received (src= 0) OK
 [0]:  'index1.html: line 1'
 [1]:  'index1.html: line 2'

*** Enetring a delay peroid of 3000 msec
*** Exiting delay peroid

Transmitted (src= 2) GTIME
Received (src= 0) (TIME: 7.97)

Transmitted (src= 2) DONE
Received (src= 0) OK
do_client: client closing connection
wpzhao@ug07:~/Cmput379/assignment3>
```

  - When running the test provided on eclass we can see that they output the same result and both client and server are running on different lab machines, with the server runned on ub05.cs.ualberta.ca.

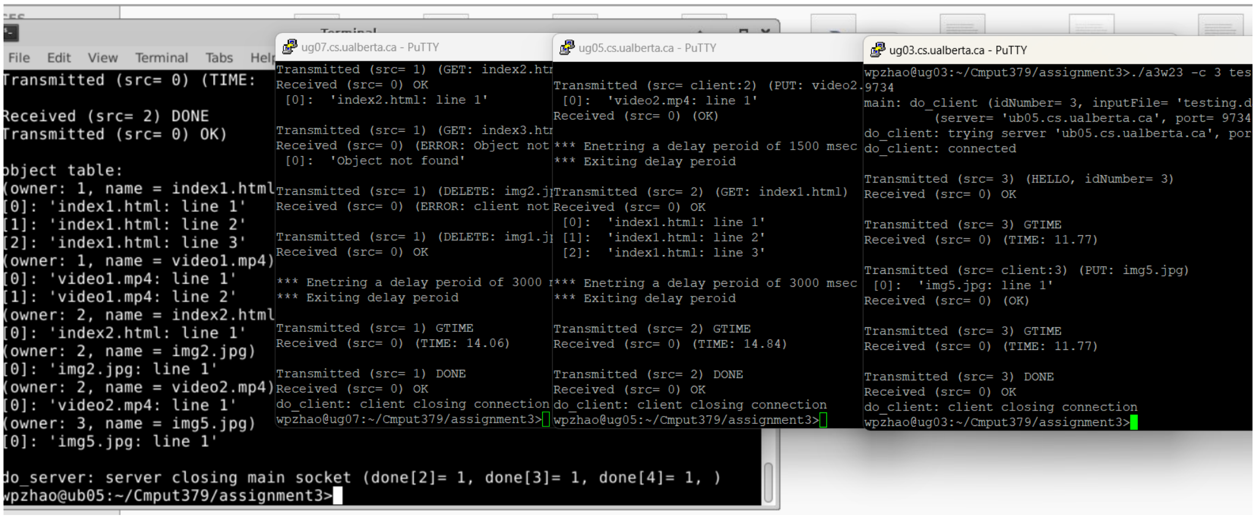


```
Received (src= 1) (GET: index2.html)
Transmitted (src= 0) OK

Received (src= 1) (GET: index3.html)
Transmitted (src= 0) (ERROR: Object not fo

Received (src= 1) (DELETE: img2.jpg)
Transmitted (src= 0) (ERROR: client not ow

rcvFrame: received frame has zero length
server lost connection with client

Received (src= 1) (GTIME)
Transmitted (src= 0) (TIME:     7.50)

Received (src= 1) DONE
Transmitted (src= 0) OK)

object table:
(owner: 1, name = index1.html)
[0]: 'index1.html: line 1'
[1]: 'index1.html: line 2'
(owner: 1, name = img1.jpg)
```

```
Transmitted (src= 1) GTIME
Received (src= 0) (TIME: 4.50)

Transmitted (src= 1) (GET: index2.html)
Received (src= 0) OK
 [0]:  'index2.html: line 1'

Transmitted (src= 1) (GET: index3.html)
Received (src= 0) (ERROR: Object not fou
 [0]:  'Object not found'

Transmitted (src= 1) (DELETE: img2.jpg)
Received (src= 0) (ERROR: client not own

*** Enetring a delay peroid of 3000 msec
*** Exiting delay peroid

Transmitted (src= 1) GTIME
Received (src= 0) (TIME: 7.50)

Transmitted (src= 1) DONE
Received (src= 0) OK
do_client: client closing connection
wpzhao@ug05:~/Cmput379/assignment3>
```

```
        (server= 'ub05.cs.ualberta.ca', port= 97
do_client: trying server 'ub05.cs.ualberta.ca', p
do_client: connected

Transmitted (src= 2) (HELLO, idNumber= 2)
Received (src= 0) OK

Transmitted (src= 2) GTIME
Received (src= 0) (TIME: 3.82)

Transmitted (src= client:2) (PUT: index2.html)
 [0]:  'index2.html: line 1'
Received (src= 0) (OK)

Transmitted (src= client:2) (PUT: img2.jpg)
 [0]:  'img2.jpg: line 1'
Received (src= 0) (OK)

Transmitted (src= client:2) (PUT: video2.mp4)
 [0]:  'video2.mp4: line 1'
Received (src= 0) (OK)

*** Enetring a delay peroid of 1500 msec
^C
wpzhao@ug07:~/Cmput379/assignment3>
```

- We can also see from the example run, we can see that client 2 is interrupted and the server can detect that.
- There were also other tests created similar to the example test to test the delete command to see if the content as well as the file name is correctly deleted. And testing putting an object that already exists, while changing 1 or 2 lines of the same test to see if it is the correct output, and the results are correct.
- The test testing.dat uses 3 clients to see if actually can run with 3 clients, which through running shows it can. This is shown below.



```
Transmitted (src= 0) (TIME:

Received (src= 2) DONE
Transmitted (src= 0) OK)

object table:
(owner: 1, name = index1.html
[0]: 'index1.html: line 1'
[1]: 'index1.html: line 2'
[2]: 'index1.html: line 3'
(owner: 1, name = video1.mp4)
[0]: 'video1.mp4: line 1'
[1]: 'video1.mp4: line 2'
(owner: 2, name = index2.html
[0]: 'index2.html: line 1'
(owner: 2, name = img2.jpg)
[0]: 'img2.jpg: line 1'
(owner: 2, name = video2.mp4)
[0]: 'video2.mp4: line 1'
(owner: 3, name = img5.jpg)
[0]: 'img5.jpg: line 1'

do_server: server closing main socket (done[2]= 1, done[3]= 1, done[4]= 1, )
wpzhao@ub05:~/Cmput379/assignment3>
```

```
Transmitted (src= 1) (GET: index2.htm
Received (src= 0) OK
 [0]:  'index2.html: line 1'

Transmitted (src= 1) (GET: index3.htm
Received (src= 0) (ERROR: Object not
 [0]:  'Object not found'

Transmitted (src= 1) (DELETE: img2.jp
Received (src= 0) (ERROR: client not

Transmitted (src= 1) (DELETE: img1.jp
Received (src= 0) OK

*** Enetring a delay peroid of 3000 m
*** Exiting delay peroid

Transmitted (src= 1) GTIME
Received (src= 0) (TIME: 14.06)

Transmitted (src= 1) DONE
Received (src= 0) OK
do_client: client closing connection
wpzhao@ug07:~/Cmput379/assignment3>
```

```
Transmitted (src= client:2) (PUT: video2.
 [0]:  'video2.mp4: line 1'
Received (src= 0) (OK)

*** Enetring a delay peroid of 1500 msec
*** Exiting delay peroid

Transmitted (src= 2) (GET: index1.html)
Received (src= 0) OK
 [0]:  'index1.html: line 1'
 [1]:  'index1.html: line 2'
 [2]:  'index1.html: line 3'

*** Enetring a delay peroid of 3000 msec
*** Exiting delay peroid

Transmitted (src= 2) GTIME
Received (src= 0) (TIME: 14.84)

Transmitted (src= 2) DONE
Received (src= 0) OK
do_client: client closing connection
wpzhao@ug05:~/Cmput379/assignment3>
```

```
wpzhao@ug03:~/Cmput379/assignment3>./a3w23 -c 3 tes
9734
main: do_client (idNumber= 3, inputFile= 'testing.d
        (server= 'ub05.cs.ualberta.ca', port= 9734
do_client: trying server 'ub05.cs.ualberta.ca', por
do_client: connected

Transmitted (src= 3) (HELLO, idNumber= 3)
Received (src= 0) OK

Transmitted (src= 3) GTIME
Received (src= 0) (TIME: 11.77)

Transmitted (src= client:3) (PUT: img5.jpg)
 [0]:  'img5.jpg: line 1'
Received (src= 0) (OK)

Transmitted (src= 3) GTIME
Received (src= 0) (TIME: 11.77)

Transmitted (src= 3) DONE
Received (src= 0) OK
do_client: client closing connection
wpzhao@ug03:~/Cmput379/assignment3>
```

- **Acknowledgments:** acknowledge sources of assistance
  - Cmput 379 lab notes posted on eclass
  - Advanced Programming in the UNIX® Environment, Third Edition
  - Operating System Concepts, 10th Edition