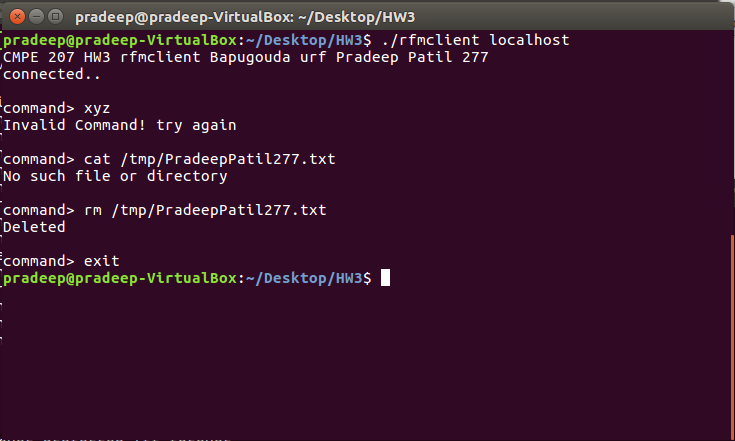
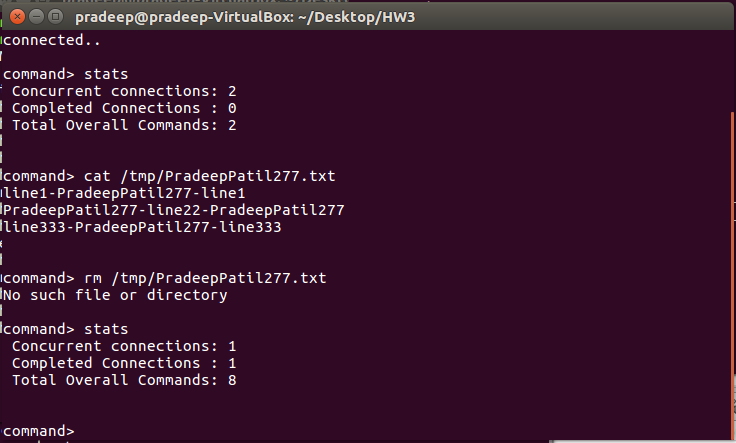
l. At this moment, take screenshots of stdout of the following shells:

 VM2/client 1

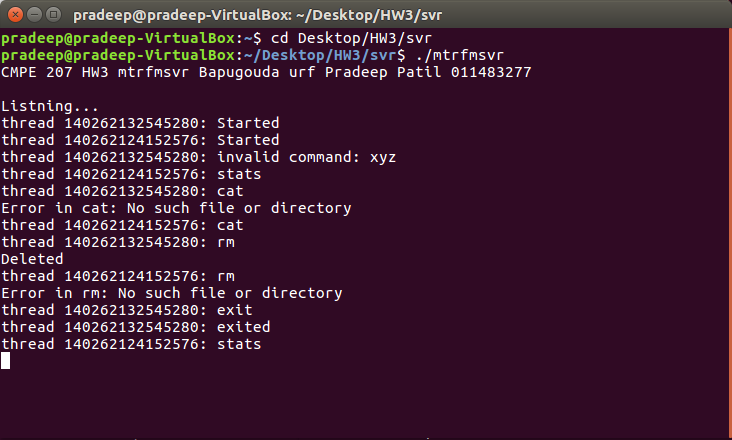




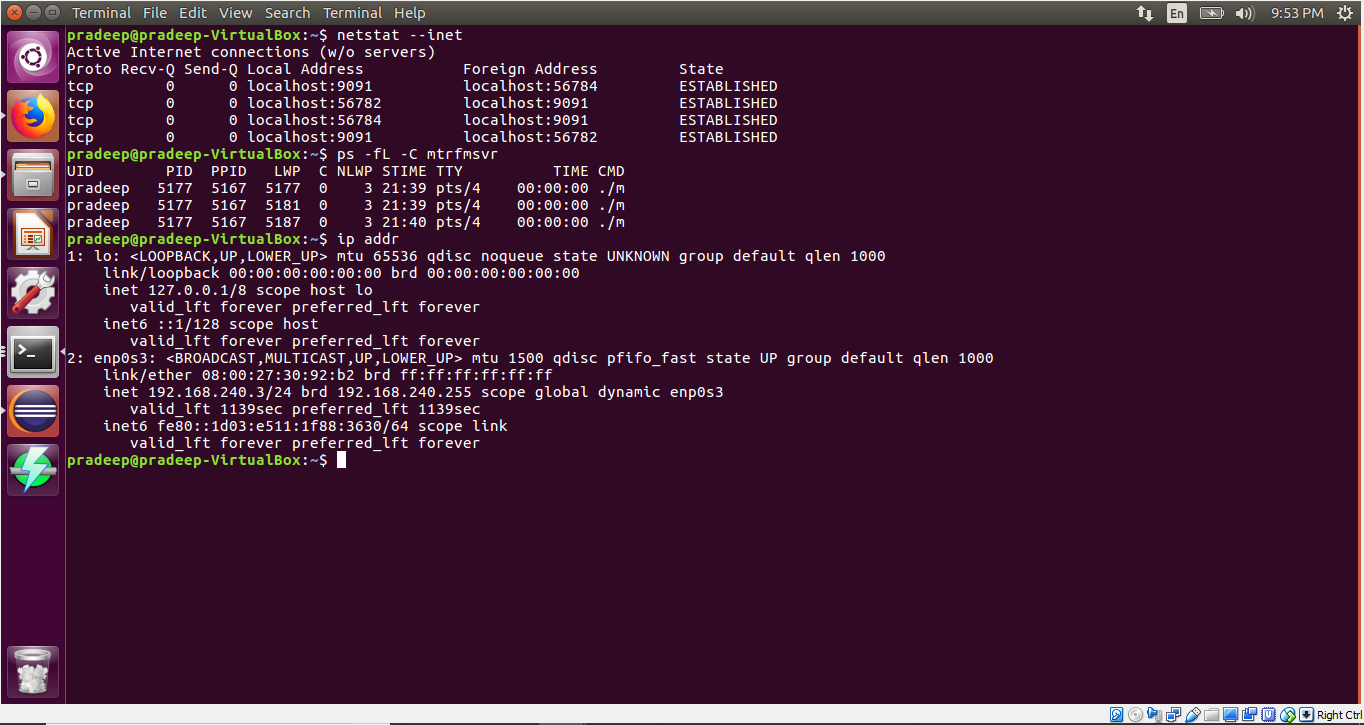
VM2/client 2



 VM1 server



 VM1/other-shell



a. generic request format, generic response format (both success case & error case). For each field, indicate its data type (binary, text), length, network byte order, etc.) 4

answer:

generic request format: command <Path><00>

generic response format: response<00>

b. Use “cat /tmp/foo” as an example: illustrate request format, illustrate response format (success case & error case)

answer:

request fromat: “cat /tmp/foo00”

Opcode: cat

Path: /tmp/foo

Reaponse: “<responsestring>00”

c. Use “rm /tmp/bar” as an example: illustrate request format, illustrate response format (success case & error case)

request fromat: “rm /tmp/bar00”

Opcode:rm

Path: /tmp/bar

Reaponse: “Deleted00”

d. Use “stats” as an example: illustrate request format, illustrate response format (success case & error case)

request fromat: “stats<00>”

Opcode: stats

Path: NONE

Reaponse: “<status string>00”

e. Use “exit” as an example: illustrate request format, illustrate response format (success case & error case)

request fromat: “exit<00>”

Shutdown(SHUT\_WR);

Opcode: exit

Path: NONE

Reaponse: “<exit>00”

Shutdown(SHUT\_WR);

5. (15 pts) Each client uses a single/persistent TCP connection to communicate with the server, and the client can

send multiple requests through the same persistent connection. Include server-side code snippet with comments to describe how the server-side pthread retrieves each client request in its entirety via a single/persistent connection.

Answer:

Client always pads 00 at the end of the command string which is detected at server and the comman is processed and appropriate value is returned.

Following is the code snippet

static const char CmdEnd[]="00";

while(1){

rd=read(ClientSock,ReadBuffer,sizeof(ReadBuffer)-1);

if(rd<0){ strerror\_r(errno,GERRORF,sizeof(GERRORF));printf("Error: %s",GERRORF);return 0;}

//printf("\ninreading..");

if((ReadBuffer[strlen(ReadBuffer)-1]==CmdEnd[1]) & (ReadBuffer[strlen(ReadBuffer)-2]==CmdEnd[0])) break;

}

break when last two chars are oo.