**CIS 527: Computer Networks**

P1 : Socket Programming

Submitted By: Manasi Kshirsagar

10 Dec 2021

Table of Contents

[Problem Statement 3](#_Toc90065249)

[Approach 4](#_Toc90065250)

[Instruction to compile and run the code 6](#_Toc90065251)

[Testcases 7](#_Toc90065252)

[Source Code 11](#_Toc90065253)

## Problem Statement

The yamotd server performs two functions:   
• returns the message of the day to any user that sends the server a MSGGET message   
• allows a client to send the server a MSGSTORE message to upload one messages of the day   
to the server. The uploaded message will replace the originally stored message, to be returned   
to another client upon receiving the MSGGET message.

## Approach

I have created two programs one for Client and another for Server.

I have implemented two functionalities for client and server communication. The functions are described below:

1. MSGGET –

This function is implemented to get the “message of the day” value from Server to client. Client will receive that value and displays it.

The MSGGET will be executed as follows:

1. The Client will send a MSGGET request to the Server.
2. MSGGET request will be encoded and sent over the network to the Server.
3. Server will receive the request from client.
4. It will decode the received request.
5. If received request is “MSGGET” then the Server will send encoded message to the client.

Encoded message is consisting of - The ACK message followed by new line character followed by Message of the day value followed by new line character.

1. The Client will complete this command
2. MSGSTORE –

This function is implemented to update the “Message of the day” value stored on the Server.

The MSGSTORE will be executed as follows:

1. The Client will send a MSGSTORE request to the Server.
2. MSGSTORE request will be encoded by the Client and sent over the network to the Server.
3. Server receives the message and decodes it.
4. If received message is “MSGSTORE” then the Server will send the encoded ACK message to the Client.
5. Client will receive the message and decodes it. If received message is ACK message, then the Client will ask user if he/she needs to update the “Message of the day”.

If user provides any message of the day value from command prompt, then the Client will encode that message and send to the Server.

Else the Client will send the default value for Message of the day.

1. The Server will receive the encoded message, it will decode the message and update the “Message of the day” value with new “Message of the day” value.
2. The Server will send the encoded ACK message followed by new line character to the Client.
3. The Client will receive the encoded message, it will decode the message and prints it on the command prompt.
4. The Client will complete this command
5. EXIT –

This function is implemented to exit Program

If “EXIT” is received by the Client. The Client will terminate the program.

Note: ACK message = 200 OK

If user input is anything other than MSGGET, MSGSTORE or EXIT, the program will prompt the message 'Unrecognised Input, Please try again' .

## Instruction to compile and run the code

1. Open two terminals one for Server side and another for Client side.
2. On the terminal, browse to the folder containing script
3. Let’s find out the IP address of the Server; SERVER\_IP.
4. First, Let’s start the Server by executing the Server code. Run the below command on 1st terminal window.

python3 P1\_Server.py

1. Secondly, Let’s execute the Client code. To execute the Client-side code, we need to pass the Server IP address as a parameter.

python3 P1\_Client.py 192.168.0.11. or python3 P1\_Client.py SERVER\_IP

1. On the Client side, Select the function MSGGET, MSGSTORE or EXIT
2. If selected function is MSGGET, program will return ACK message (200 OK ) followed by new line character followed by ‘Message of the day’ followed by new line character.
3. If Selected function is MSGSTORE, it will ask to provide update value for ‘Message of the day’.

If user provide the a ‘message of the day’ value, it will forward the value to the Server and update the ‘message of the value’. Whereas if user do not want to provide a value for ‘message of the day’, program will provide the default update value for message of the day.

1. If user provides any input other than MSGGET, MSGSTORE or EXIT, program will print a message ‘Unrecognised Input, Please try again', then user should provide new input value.
2. After completion of the activity, user should provide ‘Exit’ command to exit the program.

## Testcases

1. If function input provided from Client side is not MSGET, MSGSTORE or EXIT.

Program will show a message ‘Unrecognised Input, Please try again' and wait for new input

Client Side:

Text

Description automatically generated

Server Side:

Text

Description automatically generated

1. While terminating a program user provide either ‘exit’ or ‘Exit’ or ‘EXIT’, then it will terminate the program.

Server Side:

Text

Description automatically generated

Client Side:

Text

Description automatically generated

1. If the Server IP address provided as parameter is wrong, program will throw an exception and Client program will get terminated.

Server is still up and running. Client can re-enter the Server IP address as parameter and re-run the program.

Server Side: Server is still up and running

Text

Description automatically generated

Client Side: Client has thrown an exception and will terminate the program.

Text

Description automatically generated

Client program can be re-run after exception.

Text

Description automatically generated

## Source Code

Server side: P1\_Server.py

#!/usr/bin/env python3

#######################################################

# CIS 527: Computer Networks

# Name: Manasi Kshirsagar

# UMID- 05494807

# Instructor: Prof. Zheng Song

# Semester: Fall 2021

# P1 assignment: Socket Programming - Server Side

#######################################################

import socket

#############################################################################

################ Disclaimer ################

###### Following section of code finds IP address automatically

# when this file runs on any computer(connected to internet)

# However, port 80 might be busy, in that case,

# this file when run will produce errors on terminal.

# In that casem Kindly comment following section and hardcode appropriate server IP below.

# Find server IP address

s = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

s.connect(("8.8.8.8", 80))

print("Server IP: ", s.getsockname()[0]) # Server IP address

SERVER\_IP = s.getsockname()[0]

s.close()

################ End of Disclaimer ################

#############################################################################

# Configure Server IP below

# SERVER\_IP = '192.168.0.11'

# Define server port using UMID

SERVER\_PORT = 4807

acknowledgment = '200 OK' # Intializing the ACK signal value

message\_of\_the\_day = 'Anyone who has never made a mistake has never tried anything new.' # Initializing Message of the day value

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.bind((SERVER\_IP, SERVER\_PORT)) # Associate the socket with a specific network interface and port number

s.listen() # listens for connections from client

conn, addr = s.accept() # to establish a connection to the server and client

with conn:

while True: # loop will iterate till client connection is active

receivedData = conn.recv(1024)

receivedData = receivedData.decode()

if not receivedData:

print('No data received from client. Exiting program.\n')

break

elif (receivedData == 'MSGGET\n'): ## MSGGET Command

sending = acknowledgment +'\n '+ message\_of\_the\_day + '\n' # send the ACK and Message of the day from server to client

conn.sendall(sending.encode()) # Encode the message being sent

elif (receivedData == 'MSGSTORE\n'): ## MSGSTORE Command

#print('This is msg store\n',receivedData)

conn.sendall(acknowledgment.encode()) # Send the encoded ACK message from server to client

receivedData = conn.recv(1024) # Receive the message of the day sent from client

receivedData = receivedData.decode() # Decode the message of the day sent by client

message\_of\_the\_day = receivedData # Update the existing Message of the day by new one sent by client

conn.sendall((acknowledgment+'\n').encode()) # Send ACK to client

Client side: P1\_Client.py

#!/usr/bin/env python3

#######################################################

# CIS 527: Computer Networks

# Name: Manasi Kshirsagar

# UMID- 05494807

# Instructor: Prof. Zheng Song

# Semester: Fall 2021

# P1 assignment: Socket Programming - Client Side

#######################################################

import socket

import sys

SERVER\_IP = sys.argv[1] #'127.0.0.1' # The server's hostname or IP address

print("server ip provided", SERVER\_IP)

SERVER\_PORT = 4807 # Server port

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

try:

s.connect((SERVER\_IP,SERVER\_PORT)) # Connection established between client and server

while(True):

print('\n')

client\_input = input('Please provide input MSGGET or MSGSTORE or EXIT: ')

if (client\_input == 'MSGGET'): ## MSGGET Command

client\_input = client\_input +'\n'

print('c: MSGGET')

s.sendall(client\_input.encode()) # Send the encoded data over the network from client to server

receivedData = s.recv(1024) # Receieved the encoded data from Server

receivedData = receivedData.decode() # Decode the received data received from server

print('s: '+ receivedData) # print the received decoded data

elif (client\_input == 'MSGSTORE'): ## MSGSTORE Command

client\_input = client\_input +'\n'

print(client\_input)

print('c: MSGSTORE')

s.sendall(client\_input.encode()) # Send the encoded data over the network from client to server

receivedData = s.recv(1024) # Receieved the encoded data from Server

receivedData = receivedData.decode()# Decode the received data received from server

#print("received:",receivedData)

if (receivedData == '200 OK'): # If Received data is ACK i.e '200 OK' , proceed if further steps

print('s: '+receivedData)

message\_of\_the\_day = 'Imagination is more important than knowledge.' # default update value for message of the day

print('Default input message of the day is: ', message\_of\_the\_day)

user\_input\_message\_of\_the\_day = input('Please provide message of the day: ') # User input for message of the day

if (user\_input\_message\_of\_the\_day != ''):

message\_of\_the\_day = user\_input\_message\_of\_the\_day

print('c: '+ message\_of\_the\_day)

s.sendall(message\_of\_the\_day.encode()) # Encode the message of the day value and send it from client to server

receivedData = s.recv(1024) # Receive the ACK message

receivedData = receivedData.decode() # Decode the received ACK message

print('s: '+receivedData) # Print the ACK message

else:

print('ACK not received. Please resend the data')

continue

elif (client\_input == 'EXIT' or client\_input == 'exit' or client\_input =='Exit'): # Exit the program

print('Exit requested by user. Exiting the program.')

break

else:

print('Unrecognised Input, Please try again') # Unrecognised input

continue

except ConnectionRefusedError:

print('Connection can not be established. Please Reverify the Server IP.')