University of Genova

Assignment Report

Advance and Robot Programming, 2019 – 2020

Syed Muhammad Raza Rizvi – S4853521
5-9-2021

Table of Contents

Code Description

The Token used in this system is a struct with two attributes.

1. Data
2. Time stamp

1. Server.c

This is the G process of the system and it serves as the server side on the Socket.
First the server creates and binds a socket with the P Process and then starts to receive the Tokens.
This process receives the Token from the P process using a Socket and writes it into the server Fifo.

2. Client.c

This is the P process of the system and it serves as the client side of the socket.
A socket is created with the Process G to send the Tokens.
First, this process initializes the Token with a manual entry by the user and this Token is passed to the G process. Then this Process uses the SELECT command to see if the there is data in the Signal Fifo or in the Serve Fifo. If there is Data in the Signal Fifo, priority is given to it and the Process reads data from the Signal Fifo.
If the signal read from the Signal Fifo is 'Start', this means that now P can start receiving Tokens from G, and so the start flag is set to 1.

If the signal received from the Signal Fifo is 'Stop', this means that now P is supposed to stop receiving Tokens from G, and so the start flag is set to 0.

If the signal received from the Signal Fifo is 'dump log', this means that now the contents of the log file are to be dumped on the screen and stop receiving new Tokens.

If the Start flag is set to 1, and there is data in the Server Fifo, then P starts to receive Tokens from G through the server Fifo.
After receiving the Token, P computes a new value of the Token and waits for DT microseconds before again sending it to G through the socket.

Meanwhile, all these updates are to be logged, so after selecting between S and G, this step is written into the log Fifo, along with a time stamp. When the new Token is sent to G, this is also written in the log Fifo with a timestamp.

3. Signal.c

The process first creates a named Pipe / Fifo, '/tmp/signalfifo', and opens it.
Then the user is asked to type a signal.

The allowed signals are:
1. start
2. stop
3. dump log

The received signal is then written on the Signal Fifo, and the cycle repeats.

4. logfilecode.c

This file is developed for the Log Process of the system.
It reads the character array present in the log Fifo and appends the received characters inside the log file, 'log.log'.

Results

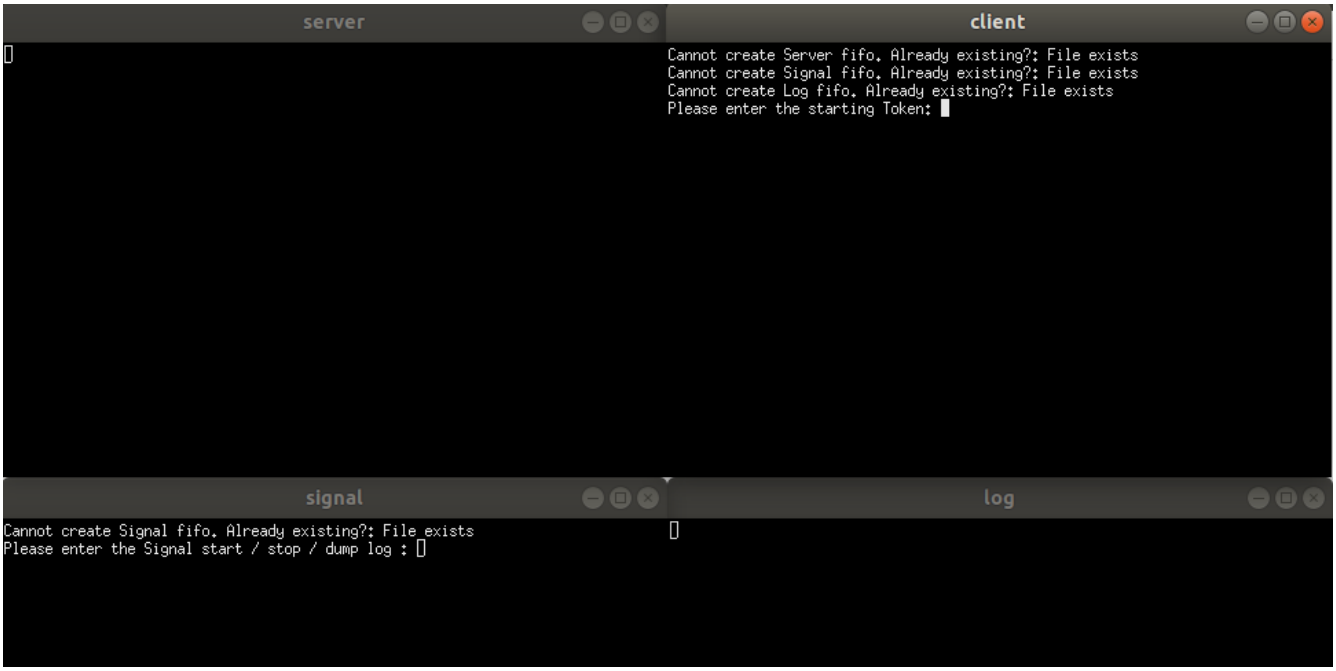
Using the following main.sh file with the following parameters.

Open ▾ main.sh ~/UniGe/ARP/ARPAssignment Save

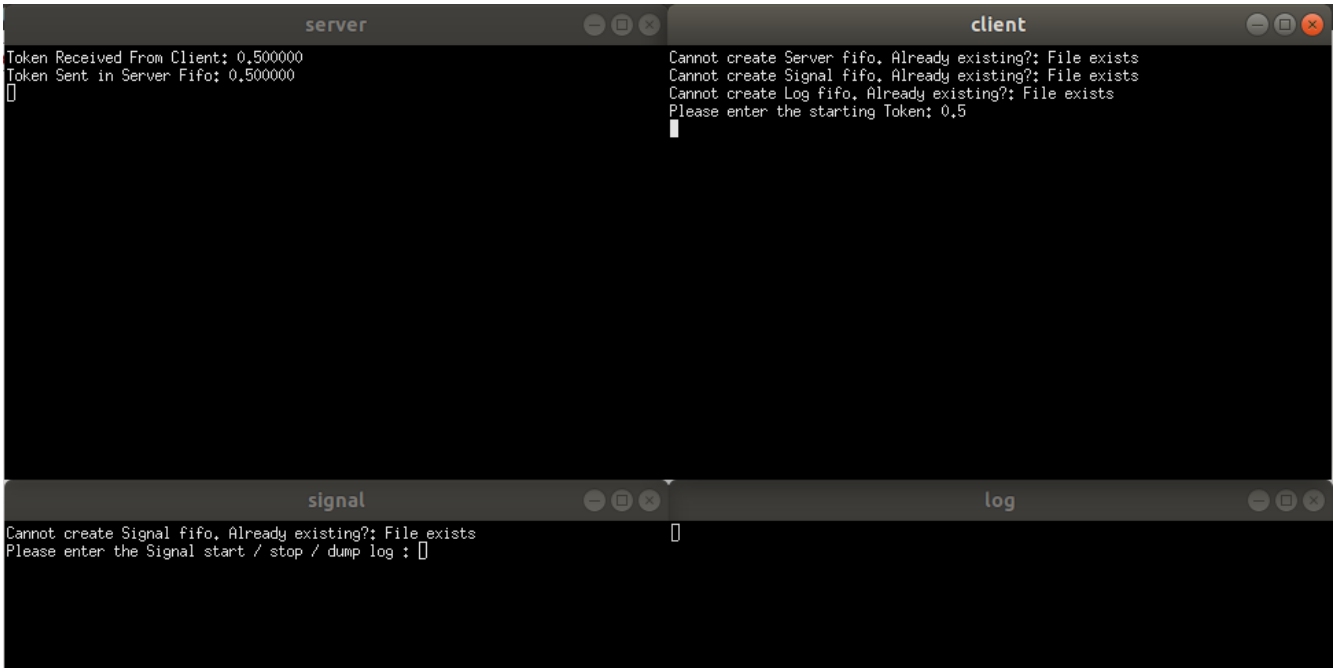
```
#!/bin/bash

#Configuration Parameters

DT="1000000"

RF="1000"

port="51717"

hostname="raza"

#Compiling the codes

gcc client.c -lm -o client
gcc server.c -o server
gcc logfilecode.c -o log
gcc signal.c -o signal

#Executing the processes

xterm -e ./server 51717 &
sleep 2

xterm -e ./client $hostname $port $DT $RF &
sleep 2

xterm -e ./signal &
sleep 1

xterm -e ./log
sleep 1
```

sh ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

When the main.sh file is executed, these four terminals are opened.

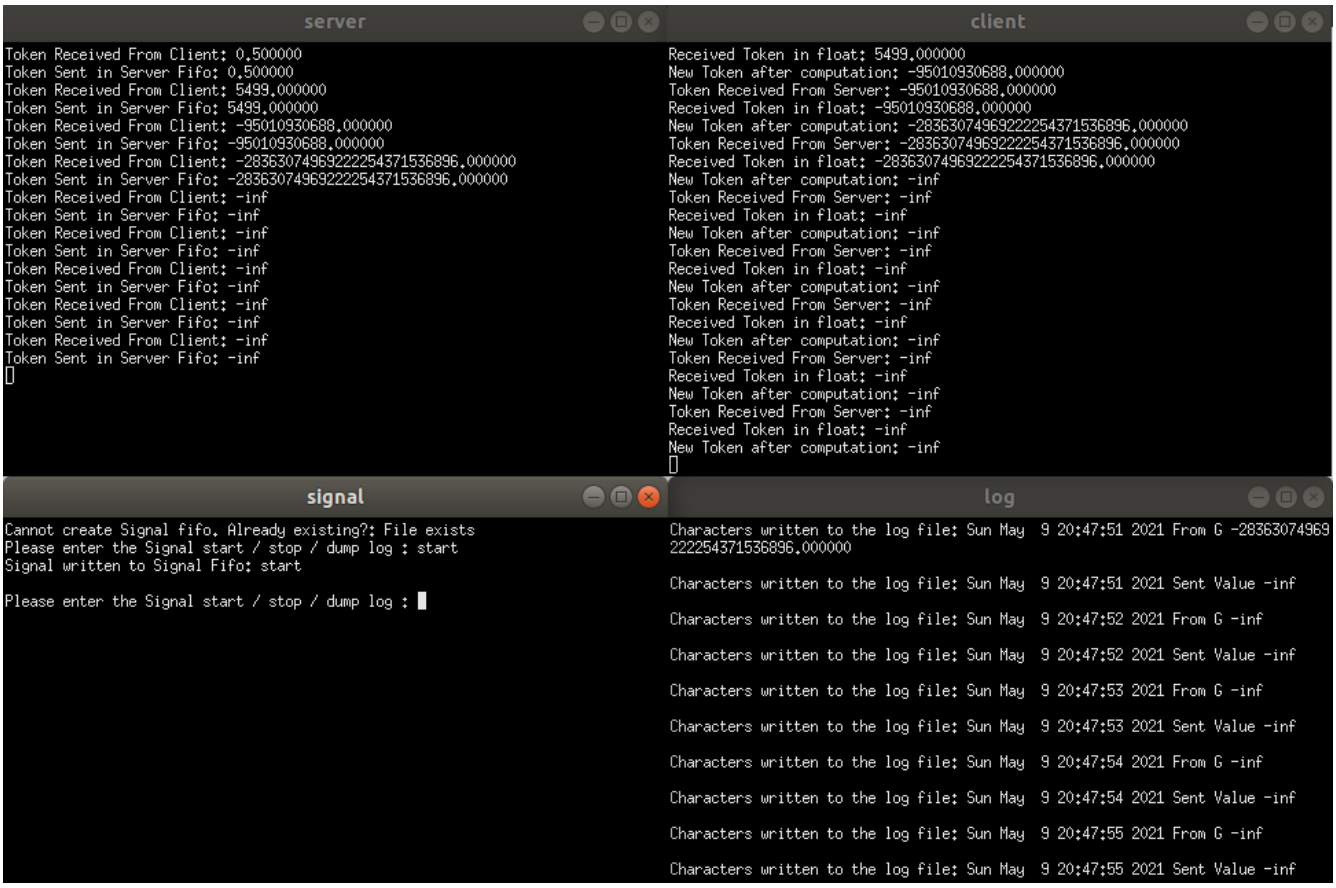


As can be seen, the client window, which is the process P, is asking for an input from the user. This input is the Initial Token to start the system.
Once the Initial Token is provided by the User, the following result is obtained.



As can be seen in the image above, now the initial Token is sent from client (P) to the server (G), using a socket. The server then writes the received Token into the Server Fifo. But since the user has not given the ‘Start’ signal in the Signal window, the process is not moving forward.

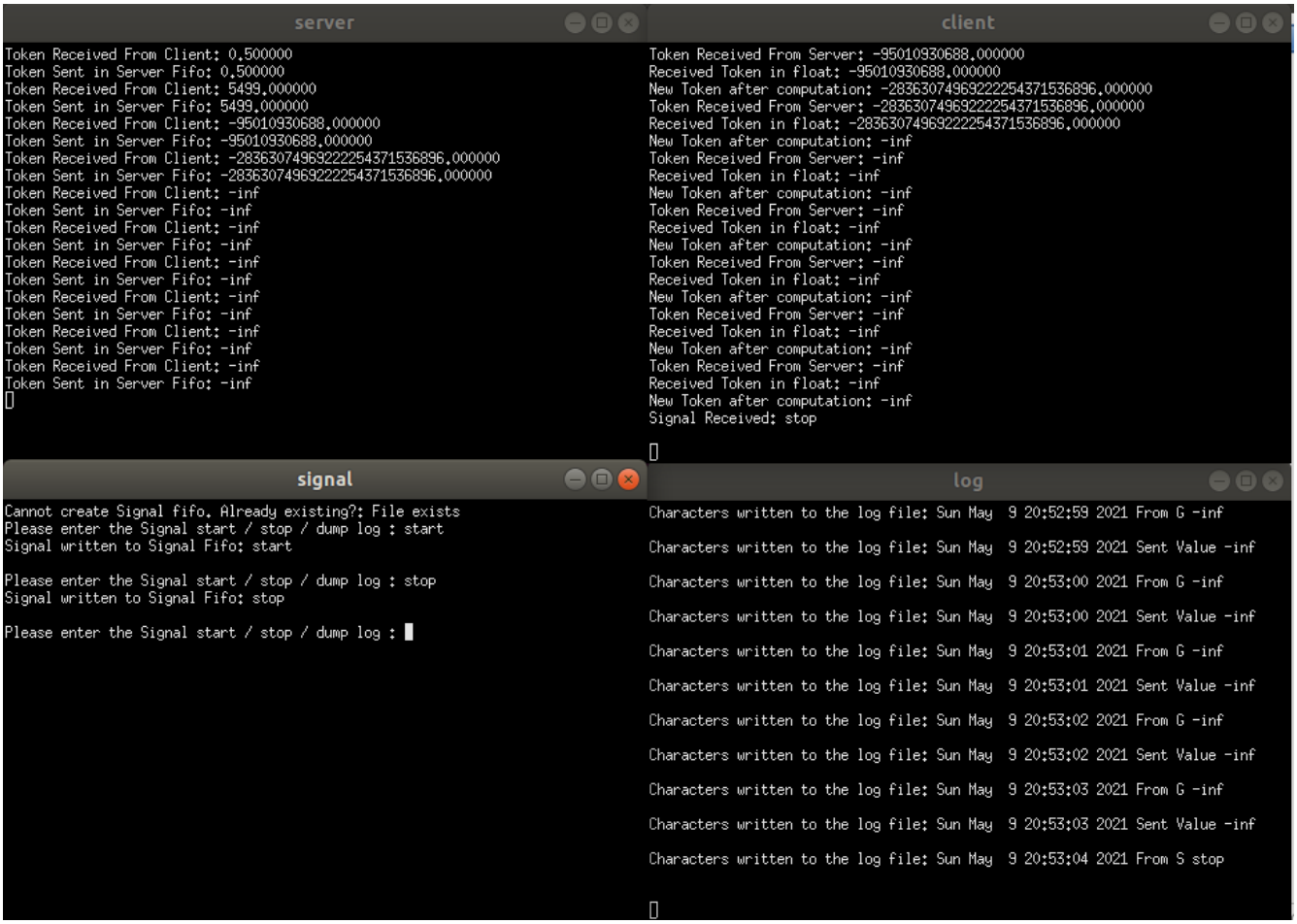
Once a ‘Start’ signal is given in the signal window, the process starts as can be seen in the image below.



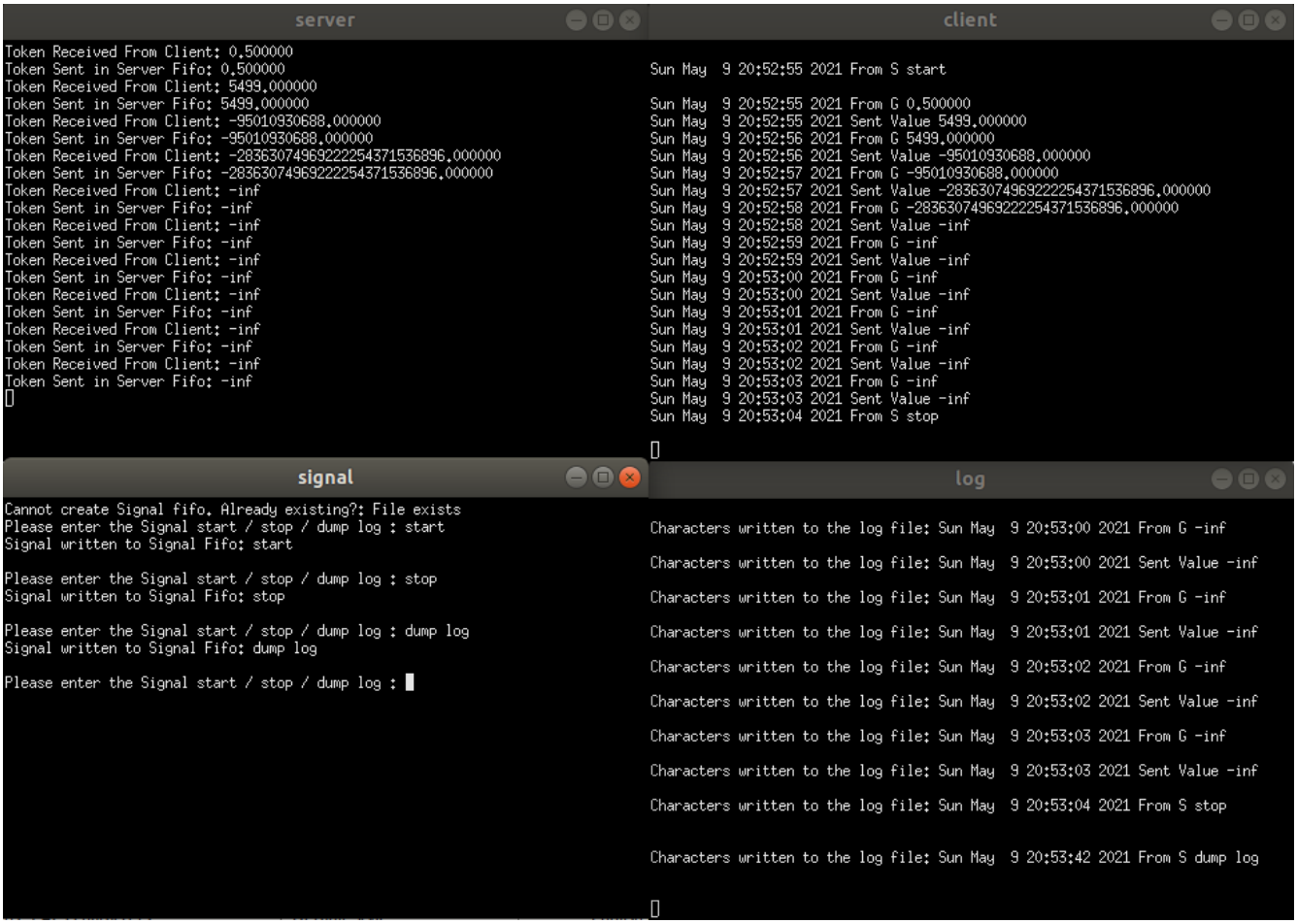
As can be seen in the image above, after receiving the ‘start’ signal, the client (P) starts to read the Token from the server (G). The client (P) then computes a new Token and sends to the server (G) through the socket, and the cycle repeats.
Meanwhile, as can be seen in the log window, all these steps are being logged in the ‘log.log’ file.

As can be seen, the Token value is reaching -inf, this is because there is an error in the provided formula for calculating the new Token.

If we enter the ‘stop’ signal, the client (P) stops receiving Tokens from the server (G), and hence doesn’t compute and new Token, as can be seen in the image below.



At last, if we enter the signal ‘dump log’, the client (P) dumps all the contents written in the log.log file, on the screen, and stops receiving any more Tokens. This can be seen in the image below.



The contents of the log.log file can also be seen in the log.log file, shown below.