**OPERATING SYSTEM ASSIGNMENT # 2**

**TASK 1 REPORT**

**GROUP MEMBERS:**

* Muhibullah Sherwani
* Faraz Ahmed
* Faris Jamil

Class ID: 101348

Date: 21-April-2019

**TASK 1 CODE EXPLANATION:**

**Client.h**

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <fcntl.h>

#include <limits.h>

#include <sys/types.h>

#include <sys/stat.h>

#define SERVER\_FIFO\_NAME "/tmp/serv\_fifo"

#define CLIENT\_FIFO\_NAME "/tmp/cli\_%d\_fifo"

#define BUFFER\_SIZE 20

struct data\_to\_pass\_st {

// Client Process ID

pid\_t client\_pid;

// Data Storage Buffer Size

char some\_data[BUFFER\_SIZE - 1];

};

**Server.c**

#include "client.h"

#include <ctype.h>

int main()

{

// initialize server and client FIFO

int server\_fifo\_fd, client\_fifo\_fd;

// struct create to get client pid and BufferSize from client.h

struct data\_to\_pass\_st my\_data;

int read\_res;

// client FIFO string size

char client\_fifo[256];

// temporary var use for data save

char \*tmp\_char\_ptr;

// create server FIFO name and set read only

mkfifo(SERVER\_FIFO\_NAME, 0777);

// open server FIFO and set Read Only

server\_fifo\_fd = open(SERVER\_FIFO\_NAME, O\_RDONLY);

if (server\_fifo\_fd == -1)

{

fprintf(stderr, "Server fifo failure\n");

exit(EXIT\_FAILURE);

}

sleep(10); /\* lets clients queue for demo purposes \*/

do {

read\_res = read(server\_fifo\_fd, &my\_data, sizeof(my\_data));

if (read\_res > 0)

{

// Getting Data from server FIFO to client FIFO

tmp\_char\_ptr = my\_data.some\_data;

while (\*tmp\_char\_ptr)

{

// changing message LowerCase to UpperCase

\*tmp\_char\_ptr = toupper(\*tmp\_char\_ptr);

tmp\_char\_ptr++;

}

// Concatenate the Client FIFO Name with Client Process ID

sprintf(client\_fifo, CLIENT\_FIFO\_NAME, my\_data.client\_pid);

// Open Client FIFO and set Write only

client\_fifo\_fd = open(client\_fifo, O\_WRONLY);

// check FIFO value != -1 then write data and close it

if (client\_fifo\_fd != -1)

{

write(client\_fifo\_fd, &my\_data, sizeof(my\_data));

close(client\_fifo\_fd);

}

}

}

// prints all output results and then Shutdown the server FIFO

while (read\_res > 0) ;

close(server\_fifo\_fd);

// Remove Server FIFO from File System

unlink(SERVER\_FIFO\_NAME);

exit(EXIT\_SUCCESS);

}

**Client.c**

#include "client.h"

#include <ctype.h>

int main()

{

// initialize server and client FIFO

int server\_fifo\_fd, client\_fifo\_fd;

// struct create to get client pid and BufferSize from client.h

struct data\_to\_pass\_st my\_data;

int times\_to\_send;

// client FIFO string size

char client\_fifo[256];

// create server FIFO name and set Write only

server\_fifo\_fd = open(SERVER\_FIFO\_NAME, O\_WRONLY);

// If Server FIFO equals to -1

if (server\_fifo\_fd == -1)

{

fprintf(stderr, "Sorry, no server\n");

exit(EXIT\_FAILURE);

}

// Getting Client Process ID

my\_data.client\_pid = getpid();

// Concatenate the Client FIFO Name with Client Process ID

sprintf(client\_fifo, CLIENT\_FIFO\_NAME, my\_data.client\_pid);

// Make client Pipe and set to read only in condition

if (mkfifo(client\_fifo, 0777) == -1)

{

fprintf(stderr, "Sorry, can’t make %s\n", client\_fifo);

exit(EXIT\_FAILURE);

}

// Client sents data to server 5 times

for (times\_to\_send = 0; times\_to\_send < 5; times\_to\_send++)

{

// Client Data and Client ID in my\_data

sprintf(my\_data.some\_data, "GetOSMessage from %d", my\_data.client\_pid);

printf("%d sent By: %s, ", my\_data.client\_pid, my\_data.some\_data);

write(server\_fifo\_fd, &my\_data, sizeof(my\_data));

// Client FIFO open and sets Read only

client\_fifo\_fd = open(client\_fifo, O\_RDONLY);

// If Client FIFO value != -1

if (client\_fifo\_fd != -1)

{

// Read message from server pipe and sent to client

if (read(client\_fifo\_fd, &my\_data, sizeof(my\_data)) > 0)

{

printf("received By: %s\n", my\_data.some\_data);

}

// Close client FIFO

close(client\_fifo\_fd);

}

}

// Close Server FIFO

close(server\_fifo\_fd);

// Unlink Client FIFO from System File

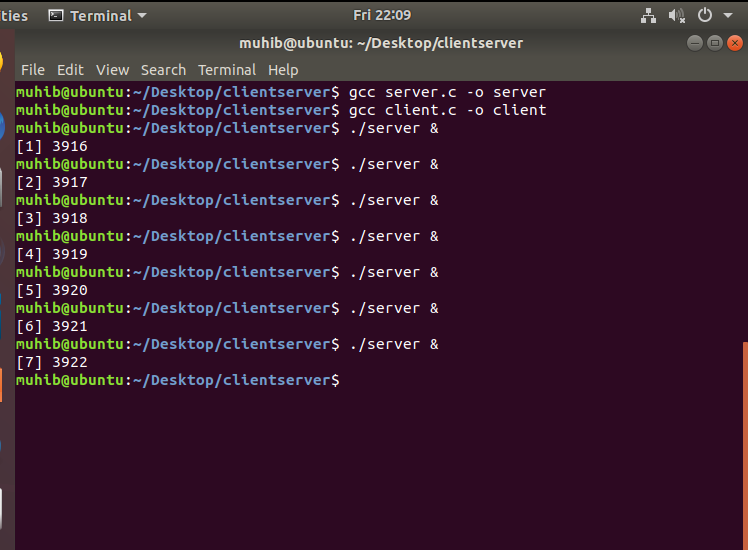
unlink(client\_fifo);

exit(EXIT\_SUCCESS);

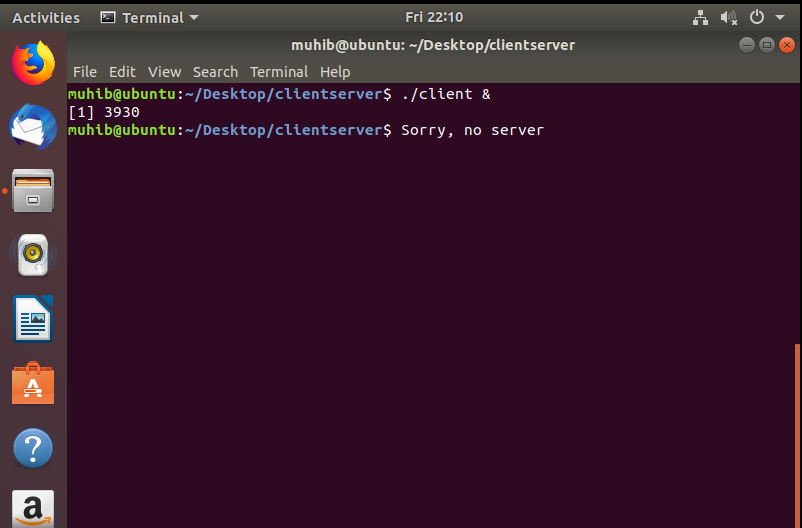
}

**TASK 1 CODE EXCECUTION SCREENSHOTS:**

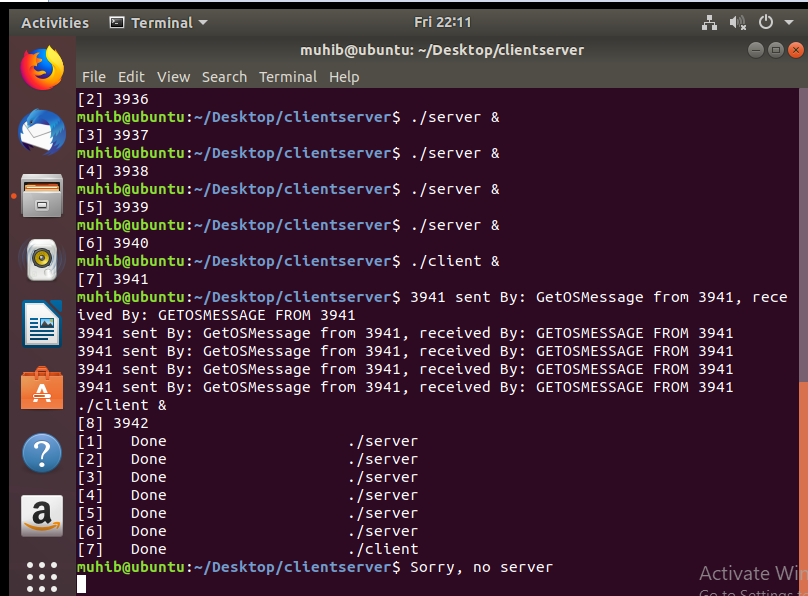
**SERVER FIFO CREATES:**

****

**CLIENT RUNS WITHOUT SERVER STARTS:**

****

**CLIENT SENTS MESSAGE 5 TIMES:**

****