Lab Report

Computer Networks

(Group C)

Submitted By

Nitish Rajbongshi

CSM21033

Q.1: Write an RPC program in Mininet to implement the Fibonacci series up to 10 number:

Solution:

Commands that are used in this problem.

1. rpcgen -a -C filename.x
2. make -f Makefile.filename
3. sudo ./filename_server
4. sudo ./filename_client localhost

The myFibbo.x file:

```
struct Myfibbo {
        int max_term;
        int result[100];
};

program MyFibbo {
        version MyFibbo_1 {
        int find_fibbo(Myfibbo) = 1;
        } = 1;

} = 0x20000001;
```

The myFibbo_client.c file:

```
#include "myFibbo.h"


void
myfibbo_1(char *host)
{
        CLIENT *clnt;
        int  *result_1;
        Myfibbo  find_fibbo_1_arg;

#ifndef  DEBUG
        clnt = clnt_create (host, MyFibbo, MyFibbo_1, "udp");
        if (clnt == NULL) {
                clnt_pcreateerror (host);
                exit (1);
        }
```

```
#endif   /* DEBUG */
        printf("Enter the maximum term of the series: ");
        scanf("%d", &find_fibbo_1_arg.max_term);

        result_1 = find_fibbo_1(&find_fibbo_1_arg, clnt);
        if (result_1 == (int *) NULL) {
                clnt_perror (clnt, "call failed");
        }

        else {
                printf("The resulttant series is: %d\n", *result_1);
                printf("The resulttant series is: %d\n", *result_1);
        }
#ifndef  DEBUG
        clnt_destroy (clnt);
#endif    /* DEBUG */
}


int
main (int argc, char *argv[])
{
        char *host;

        if (argc < 2) {
                printf ("usage: %s server_host\n", argv[0]);
                exit (1);
        }
        host = argv[1];
        myfibbo_1 (host);
exit (0);
}
```

The myFibbo_server.c file

```
#include "myFibbo.h"

int *
find_fibbo_1_svc(Myfibbo *argp, struct svc_req *rqstp)
{
        static int  *result;
        int arr[100];

        int i, sum = 0, n1 = 0, n2 = 1;
        result = 0;
```

```
        for(i = 0; i < argp->max_term; i++) {
                printf("%d\n", sum);
                n1 = n2;
                n2 = sum;
                sum = n1 + n2;
                arr[i] = sum;
        }
        result = arr;

        return result;
}
```

Q.2: Write two C program using a raw socket to send a TCP to send a TCP packet where the TCP payload will contain your roll number.

Solution:

Here, we have to two files, one is server file and other is client file.

Client can make request through the client file for a specific task and then server will response according to the server file.

The server file:

```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <unistd.h>
#include <netinet/in.h>
#include <sys/types.h>

int main()
{
   int socket_desc, client_sock, client_size;
   struct sockaddr_in server_addr, client_addr;
   char  client_message[2000];
   char* msg = "CSM21033";
   socket_desc = socket(AF_INET, SOCK_STREAM, 0);
   if(socket_desc < 0){
      printf("Error while creating socket\n");
      return -1;
   }
   printf("Socket created successfully\n");
   server_addr.sin_family = AF_INET;
   server_addr.sin_port = htons(2000);
   server_addr.sin_addr.s_addr = inet_addr("10.0.0.1");
   if(bind(socket_desc, (struct sockaddr*)&server_addr, sizeof(server_addr))<0){
      printf("Couldn't bind to the port\n");
      return -1;
   }
   printf("Done with binding\n");

   if(listen(socket_desc, 1) < 0){
      printf("Error while listening\n");
      return -1;
   }
   printf("\nListening for clients..\n");

   client_size = sizeof(client_addr);
   client_sock = accept(socket_desc, (struct sockaddr*)&client_addr, &client_size);
```

```
if (client_sock < 0){
      printf("Can't accept\n");
      return -1;
   }

   if (recv(client_sock, client_message, sizeof(client_message), 0) < 0){
      printf("Couldn't receive\n");
      return -1;
   }
   printf("Msg from client: %s\n", client_message);


   if (send(client_sock, msg, strlen(msg), 0) < 0){
      printf("Can't send\n");
      return -1;
   }

   send(socket_desc, msg , strlen(msg), 0);

   printf("message sent to client %s\n",msg);

   close(client_sock);
   close(socket_desc);

   return 0;
}
```

The client file:

```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <sys/types.h>

int main()
{
   int socket_desc;
   struct sockaddr_in server_addr;
   char server_message[2000];
   char* msg = "Give me my rollno: ";


   socket_desc = socket(AF_INET, SOCK_STREAM, 0);
```

```
if(socket_desc < 0){
        printf("Unable to create socket\n");
        return -1;
    }

    printf("Socket created successfully\n");

    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(2000);
    server_addr.sin_addr.s_addr = inet_addr("10.0.0.1");

    if(connect(socket_desc, (struct sockaddr*)&server_addr, sizeof(server_addr)) < 0){
        printf("Unable to connect\n");
        return -1;
    }


    printf("Connected with server successfully\n");

    if(send(socket_desc, msg , strlen(msg), 0) < 0){
        printf("Unable to send message\n");
        return -1;
    }

    send(socket_desc, msg , strlen(msg), 0);

    printf("message sent to server\n");

    recv(socket_desc, server_message, sizeof(server_message), 0);
    printf("Server's response: %s\n",server_message);

    close(socket_desc);

    return 0;
}
```
