COE19B055
K. Ram Mohan

Question

Implement a Ring Counter network using SP:

RCN consists of N clients nodes & one server. Server will start counter with '0' & send to Client 1. Client 1 will increment the counter value and send to Client 2. This process is repeated (in circular) until Server receives final counter value (which is nothing but no: of nodes in network)

Each client need to print its client id and received counter value.

write server and client programs for above configuration using UDP protocol

Server

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<pthread.h>
int count=0;

int main()
{
        char buff[256];
        int recv_sock;
        struct sockaddr_in receiver, sender;
        recv_sock = socket(AF_INET, SOCK_DGRAM, 0);
        receiver.sin_addr.s_addr = INADDR_ANY;
        receiver.sin_family = AF_INET;
        receiver.sin_port = htons(9001);
        bind(receiver, (struct sockaddr*) &receiver, sizeof(receiver));
        socklen_t add;
        add = sizeof(sender);
        sprintf(buff, "%d", count);
        sendto(recv_sock, buff, sizeof(buff), 0, (struct sockaddr *) &receiver,
                sizeof(receiver));
        printf("Sending count value to client \n");
        recvfrom(recv_sock, buff, sizeof(buff), 0, (struct sockaddr*) &sender,
                add);
        int final_count = atoi(buff);
        printf("Count received from last client is %d \n", final_count);
    return 0;
}
```

Clients

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<pthread.h>
int count;
void create_client(void *id);
int main()
{
    char buff[256];
    int clients;        pthread_t pid[100];
    Printf("Enter no of clients: ");
    scanf("%d", &clients)

    for(i=0; i<clients; i++)

    {
    int send_sock;
    struct sock_addr_in sender, receiver;
    send_sock = socket(AF_INET, SOCK_DGRAM, 0);

    sender.sin_family = AF_INET;
    sender.sin_port = htons(9001);
    sender.sin_addr.s_addr = INADDR_ANY;

    socklen_t addr;
    addr = sizeof(sender);
    recvfrom(send_sock, buff, sizeof(buff), 0, (struct sockaddr *) &receiver,
            &addr);

    count = atoi(buff); count=count+1;
    Printf("Client @1: count = %d \n", count);
    for(i=1; i<client; i++)
    {
        pthread_create(&pid[i], NULL, create_client, (void*) i);

    }
    for(i=0; i<client; i++)
    {
        pthread_join(pid[i], NULL);
    }
}
```

```
sprintf(buff, "%d", count);
sendto(send_sock, buff, sizeof(buff), 0, (struct sockaddr *)&sender,
        sizeof(sender));

Printf("Sending final count to Server\n");

}
```

```
void * clientthread(void * id)
{
    int cid = *(int *) id;
    int c_sock;

    c_sock = socket(AF_INET, SOCK_DGRAM, 0);

    struct sockaddr_in sender, receiver;

    sender.sin_family = AF_INET;
    sender.sin_port = htons(9001);
    sender.sin_addr.s_addr = INADDR_ANY;

    Printf(" Client %d : Count received = %d \n", c_id+1, count);

    count = count+1;

    close(c_sock);
    Pthread_exit(NULL);
}
```