Exp No: 5 Date: 17/09/2020

DAYTIME SERVER USING UDP Name: Swetha Saseendran

Register Number: 185001183

## **Code:**

**SERVER.C**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <time.h>

#include <unistd.h>

#include <fcntl.h>

#include <arpa/inet.h>

#define SA struct sockaddr

#define SERVER\_PORT 8000

#define MAX 256

int server\_fd,n = 1;

struct sockaddr\_in servaddr, clientaddr;

int error(char \*msg)

{

    perror(msg);

    exit(1);

}

void ansOption(time\_t cur\_time, char option)

{

        struct tm \*temp;

    time\_t toronto\_time;

    char time\_buffer[MAX];

    int addrlen,year;

    temp = localtime(&cur\_time);

    printf("\nRequest from Client %d: Option %c\n",n,option);

    if(option == '1')

        strftime(time\_buffer, sizeof(time\_buffer), "%x", temp);

    else if(option == '2')

        strftime(time\_buffer, sizeof(time\_buffer), "%A", temp);

    else if(option == '3')

        strftime(time\_buffer, sizeof(time\_buffer), "%B", temp);

    else if(option == '4')

    {

        year = temp->tm\_year + 1900;

        sprintf(time\_buffer, "%d", year);

    }

    else if(option == '5')

        strftime(time\_buffer, sizeof(time\_buffer), "%I:%M%p", temp);

    else if(option == '6')

    {   bzero(&time\_buffer, sizeof(time\_buffer));

        temp = gmtime(&cur\_time); //GMT

        temp->tm\_hour -= 5; //Toronto is -5H ahead of GMT

        toronto\_time = mktime(temp);

        temp = localtime(&toronto\_time);

        strftime(time\_buffer, sizeof(time\_buffer), "%c", temp);

    }

    else

    {

printf("\n\tInvalid Option.\n");

        strcpy(time\_buffer,"Invalid Option");

    }

    addrlen = sizeof(clientaddr);

    sendto(server\_fd, time\_buffer, sizeof(time\_buffer), 0, (SA\*)&clientaddr,addrlen);

    printf("\nMessage sent to Client %d\n",n);

}

int main()

{

    int bytecount,addrlen, opt;

    char buf[MAX];

        time\_t cur\_time;

    //CREATE SOCKET

    if ((server\_fd = socket(AF\_INET, SOCK\_DGRAM, 0)) <= 0)

            error("\033[0;31mSOCKET ERROR\033[0m");

    //CLEAR servaddr

    bzero(&servaddr , sizeof(servaddr));

// SET IP,PORT,FAMILY OF SERVER

    servaddr.sin\_port = htons(SERVER\_PORT);

    servaddr.sin\_family = AF\_INET;

    servaddr.sin\_addr.s\_addr = INADDR\_ANY;

    //BIND

    if((bind(server\_fd, (SA\*)&servaddr, sizeof(servaddr))) !=0 )

            error("\033[0;31mBIND ERROR\033[0m");

    //RECIEVE DATA

    addrlen = sizeof(clientaddr);

    while(1)

    {

            bzero(buf,sizeof(buf));

        bytecount = recvfrom(server\_fd, buf, sizeof(buf), 0,(SA\*)&clientaddr, &addrlen);

        cur\_time = time(NULL);

        ansOption(cur\_time,buf[0]);

        n++;

    }

    return 0;

}

**CLIENT.C**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define SA struct sockaddr

#define SERVER\_PORT 8000

#define MAX 256

int error(char \*msg){

    perror(msg);

    exit(1);

}

int main(int argc, char const \*argv[])

{

    int sockfd,n;

    int con = 1;

    char opt;

    struct sockaddr\_in servaddr;

    char buf[1024];

    //CREATE SOCKET

    if((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) <= 0)

        error("\033[0;31mSOCKET ERROR\033[0m");

    //CLEAR servaddr

    bzero(&servaddr , sizeof(servaddr));

    // SET IP,PORT,FAMILY OF SERVER

    servaddr.sin\_port = htons(SERVER\_PORT);

    servaddr.sin\_family = AF\_INET;

    servaddr.sin\_addr.s\_addr = INADDR\_ANY;

    //SEND REQUEST

   printf("\nRequest to Server:\n\t[1] Date\n\t[2] Day\n\t[3] Month\n\t[4] Year\n\t[5] Time\n\t[6] Toronto Time\nYour Option: ");

    scanf("%c", &opt);

    buf[0] = opt;

    sendto(sockfd, buf, sizeof(buf), 0, (SA\*)&servaddr,sizeof(servaddr));

    bzero(buf,sizeof(MAX));

    //RECIEVE INFO

    printf("\nRequesting information from server\n\n");

    int addrlen = sizeof(servaddr);

    recvfrom(sockfd, buf, sizeof(buf), 0, (SA\*)&servaddr,&addrlen);

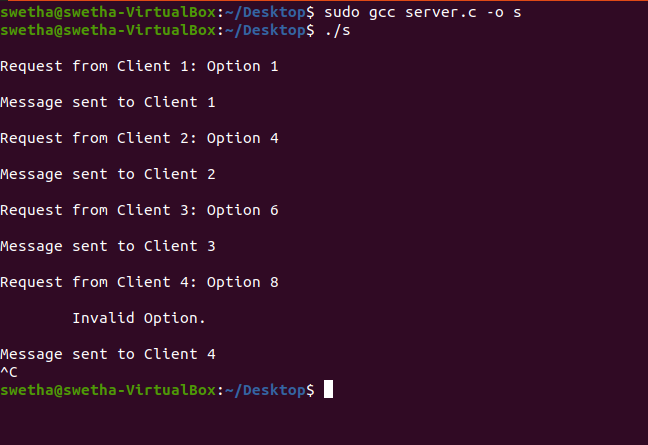
    printf("\nRecieved from server: %s\n",buf);

    return 0;

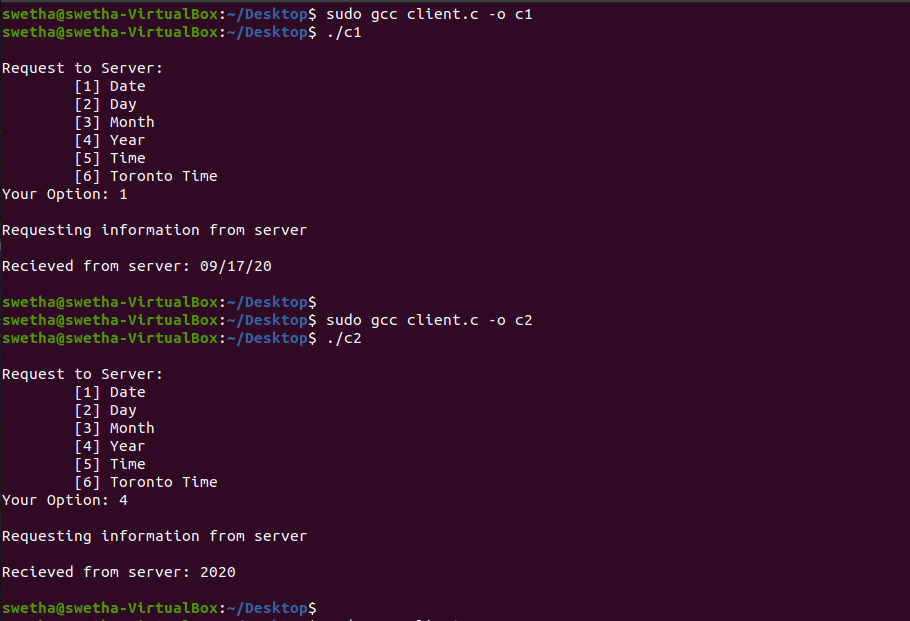
}

**Sample Input Output:**

**SERVER SIDE:**

****

**CLIENT SIDE:**

****

