**Lab 6**

Server:  
#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <unistd.h>

#include <stdlib.h>

int main(void)

{

int socket\_desc;

struct sockaddr\_in server\_addr, client\_addr;

char server\_message[2000], client\_message[2000];

int client\_struct\_length = sizeof(client\_addr);

char\* msg;

char \*Rollnum[] = {"12-1237-CI\n", "12-1236-CI\n", "12-1235-CI\n", "12-1234-CI\n", "12-1233-CI\n", "12-1232-CI\n", "12-1231-CI\n", "12-1230-CI\n"};

int Present[] = {0, 0, 0, 0, 0, 0, 0, 0};

char \*Rollnum2[] = {"12-1237-CO\n", "12-1236-CO\n", "12-1235-CO\n", "12-1234-CO\n", "12-1233-CO\n", "12-1232-CO\n", "12-1231-CO\n", "12-1230-CO\n"};

int s = 8;

// Cleaning the Buffers

memset(server\_message, '\0', sizeof(server\_message));

memset(client\_message, '\0', sizeof(client\_message));

// Creating UDP Socket

socket\_desc = socket(AF\_INET, SOCK\_DGRAM, IPPROTO\_UDP);

if (socket\_desc < 0)

{

printf("Could Not Create Socket. Error!!!!!\n");

return -1;

}

printf("Socket Created\n");

// Binding IP and Port to socket

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(2000);

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

if (bind(socket\_desc, (struct sockaddr \*)&server\_addr, sizeof(server\_addr)) < 0)

{

printf("Bind Failed. Error!!!!!\n");

return -1;

}

printf("Bind Done\n");

printf("Listening for Messages...\n\n");

// Receive the message from the client

while (1)

{

if (recvfrom(socket\_desc, client\_message, sizeof(client\_message), 0, (struct sockaddr \*)&client\_addr, &client\_struct\_length) < 0)

{

printf("Receive Failed. Error!!!!!\n");

continue; // Continue listening for the next message

}

printf("Received Message from IP: %s and Port No: %i\n", inet\_ntoa(client\_addr.sin\_addr), ntohs(client\_addr.sin\_port));

int i;

int res = 0;

int x;

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

{

if (strcmp(client\_message, Rollnum[i]) == 0)

{

res = 1;

if(Present[i]==1)

{

printf("Student Already Present\n");

msg="Student Already Present\n";

break;

}

else{

Present[i] = 1;

printf("Welcome Student %s\n", client\_message);

msg="Welcome Student";

break; // Exit the loop once a match is found

}

}

}

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

{

if (strcmp(client\_message, Rollnum2[i]) == 0)

{

res = 1;

if(Present[i]==0)

{

printf("You didn't Check In Today . Contact Administration\n");

msg="You didn't Check In Today . Contact Administration\n";

break;

}

else{

Present[i] = 0;

msg="GoodBye";

printf("Good Bye Student %s . Have a Nice Day\n", client\_message);

break; // Exit the loop once a match is found

}

}

}

if (res == 0)

{

printf("Student Not Found\n");

}

for(x=0;x<s;x++)

{

if(Present[x]==0)

{

printf("ABSENT - %s",Rollnum[x]);

}

else{

printf("PRESENT - %s",Rollnum[x]);

}

}

// Send the message back to the client

strcpy(server\_message, msg);

if (sendto(socket\_desc, server\_message, strlen(server\_message), 0, (struct sockaddr \*)&client\_addr, client\_struct\_length) < 0)

{

printf("Send Failed. Error!!!!!\n");

continue; // Continue listening for the next message

}

memset(server\_message, '\0', sizeof(server\_message));

memset(client\_message, '\0', sizeof(client\_message));

}

// The program should not reach this point because of the infinite loop

// close(socket\_desc); // Avoid exiting the program here

return 0;

}

**Client:**

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include<unistd.h>

#include<stdlib.h>

int main(void)

{

int socket\_desc;

struct sockaddr\_in server\_addr;

char server\_message[2000], client\_message[2000];

int server\_struct\_length = sizeof(server\_addr);

memset(server\_message,'\0',sizeof(server\_message));

memset(client\_message,'\0',sizeof(client\_message));

socket\_desc = socket(AF\_INET, SOCK\_DGRAM, IPPROTO\_UDP);

if(socket\_desc < 0)

{

printf("Could Not Create Socket. Error!!!!!\n");

return -1;

}

printf("Socket Created\n");

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(2000);

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

printf("Enter Message: ");

fgets(client\_message, sizeof(client\_message), stdin);

if(sendto(socket\_desc, client\_message, strlen(client\_message), 0, (struct sockaddr\*)&server\_addr, server\_struct\_length) < 0)

{

printf("Send Failed. Error!!!!\n");

return -1;

}

if(recvfrom(socket\_desc, server\_message, sizeof(server\_message),0, (struct sockaddr\*)&server\_addr, &server\_struct\_length) < 0)

{

printf("Receive Failed. Error!!!!!\n");

return -1;

}

printf("Server Message: %s\n",server\_message);

memset(server\_message,'\0',sizeof(server\_message));

memset(client\_message,'\0',sizeof(client\_message));

close(socket\_desc);

return 0;

}