COMPUTER NETWORKS LAB 05

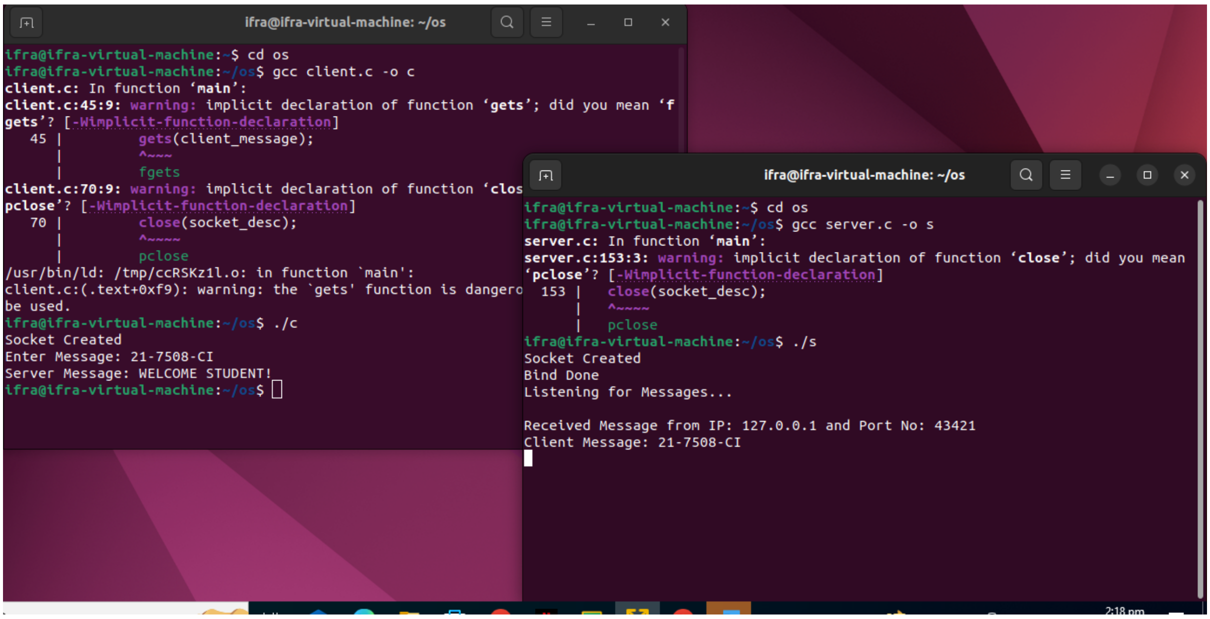
Server.c

```
#include <arpa/inet.h>

#include <stdio.h>

#include <string.h>

#include <sys/socket.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 *persons[10];

  int attendence[10];

  int i;

  for (i = 0; i < 10; i++) {

    persons[i] = "0";

  }

  for (i = 0; i < 10; i++) {

    attendence[i] = 0;

  }

  int k=0;

  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"); // bind your socket to localhost only, if you want connect
                  // any particular ip you should mention it in INET_ADDR.

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");
while (1) {

  // Receive the message from the client

  if (recvfrom(socket_desc, client_message, sizeof(client_message), 0,
               (struct sockaddr *)&client_addr, &client_struct_length) < 0) {
    printf("Receive Failed. Error!!!!!\n");
    return -1;
  }
```

```
printf("Received Message from IP: %s and Port No: %i\n",
       inet_ntoa(client_addr.sin_addr), ntohs(client_addr.sin_port));

printf("Client Message: %s\n", client_message);

// Send the message back to client
int i;

int client_size = strlen(client_message);
char check = client_message[client_size - 1];
int s = 10;
int old_s;
int flag = 0;
int count = 0;
int news;
char*new_persons[10];
int new_attendence[10];

if (check == 'I') {
  for (i = 0; i < s; i++) {
    if (strcmp(persons[i], client_message) == 0)
    {

      if (attendence[i] == 1) {
        strcpy(server_message, "you are already here!");
      } else {
        flag = 1;
        break;
      }
    }
  else
  {

    persons[k] = client_message;
    attendence[k] = 1;
    strcpy(server_message, "WELCOME STUDENT!");
    k++;
    break;

  }
 }
 if (flag == 1) {
  attendence[k] = 1;
  strcpy(server_message, "WELCOME STUDENT!");
  break;
 }
}
else if(check=='O')
{
 for(i=0;i<s;i++)
  {

   if(strcmp(persons[i],client_message)==0)
   {
    if(attendence[i]==0)
    {
     strcpy(server_message,"you did not check in today.Contact System Administrator!");
     break;
    }
    else
    {
          for(i=0;i<s;i++)
           {
             persons[i]=persons[i+1];
             attendence[i]=attendence[i+1];
           }
          s=s-1;
          strcpy(server_message,"Good Bye!!! Have a Nice Day");
          break;

         }
        }
        else
        {
        strcpy(server_message,"you did not check in today");
        break;
        }
      }
   }
   if (sendto(socket_desc, server_message, strlen(server_message), 0,
           (struct sockaddr *)&client_addr, client_struct_length) < 0) {
     printf("Send Failed. Error!!!!!\n");
     return -1;
   }

   memset(server_message, '\0', sizeof(server_message));
   memset(client_message, '\0', sizeof(client_message));
 }

 // Closing the Socket

 close(socket_desc);
```

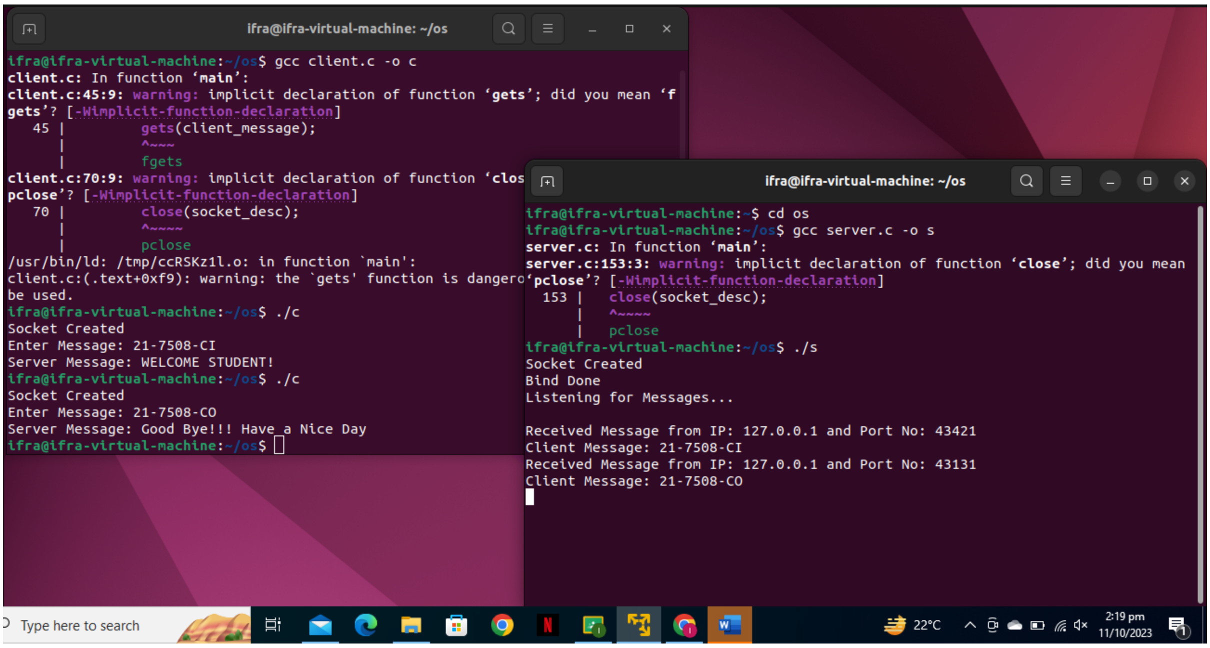
```
  return 0;
}

Client.c:
/*
        UDP_Client. This Program will implement the Client Side for UDP_Socket Programming.
        It will get some data from user and will send to the server and as a reply from the
        server, it will get its data back.
*/

#include <stdio.h>
#include <string.h>
#include <sys/socket.h> // Needed for socket creating and binding
#include <arpa/inet.h> //inet_addr

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);

        //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");

        //Specifying the IP and Port of the server to connect

        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(2000);
        server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");  // bind your socket to localhost only, if
you want connect any particular ip you should mention it in INET_ADDR.

        //Get Input from the User

        printf("Enter Message: ");
        gets(client_message);

        //Send the message to Server

        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;
        }

        //Receive the message back from the server
```

```
	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));


	//Closing the Socket


	close(socket_desc);


	return 0;

}
```


ifra@ifra-virtual-machine: ~/os
ifra@ifra-virtual-machine: ~/os

```
ifra@ifra-virtual-machine:~$ cd os
ifra@ifra-virtual-machine:~/os$ gcc client.c -o c
client.c: In function ‘main’:
client.c:45:9: warning: implicit declaration of function ‘gets’; did you mean ‘f
gets’? [-Wimplicit-function-declaration]
   45 |         gets(client_message);
      |         ^~~~
      |         fgets
client.c:70:9: warning: implicit declaration of function ‘clos
pclose’? [-Wimplicit-function-declaration]
   70 |         close(socket_desc);
      |         ^~~~~
      |         pclose
/usr/bin/ld: /tmp/ccRSKz1l.o: in function `main':
client.c:(.text+0xf9): warning: the `gets' function is dangero
be used.
ifra@ifra-virtual-machine:~/os$ ./c
Socket Created
Enter Message: 21-7508-CI
Server Message: WELCOME STUDENT!
ifra@ifra-virtual-machine:~/os$ 
```

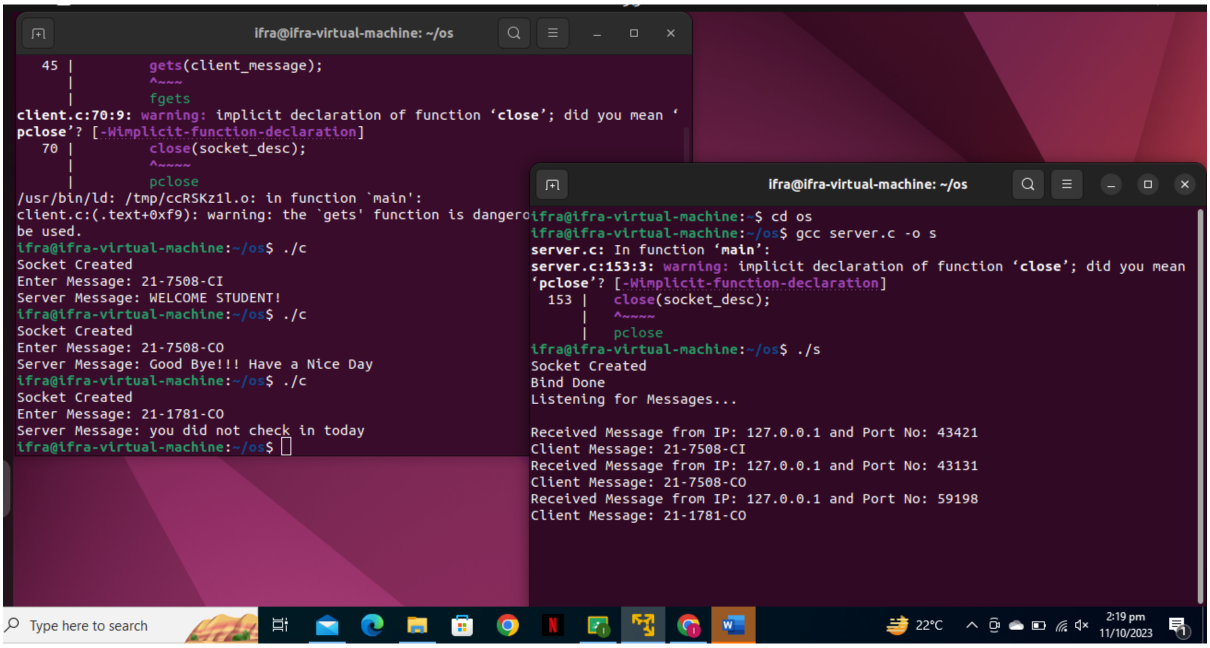
```
ifra@ifra-virtual-machine:~$ cd os
ifra@ifra-virtual-machine:~/os$ gcc server.c -o s
server.c: In function ‘main’:
server.c:153:3: warning: implicit declaration of function ‘close’; did you mean
‘pclose’? [-Wimplicit-function-declaration]
  153 |   close(socket_desc);
      |   ^~~~~
      |   pclose
ifra@ifra-virtual-machine:~/os$ ./s
Socket Created
Bind Done
Listening for Messages...

Received Message from IP: 127.0.0.1 and Port No: 43421
Client Message: 21-7508-CI
```



ifra@ifra-virtual-machine: ~/os
ifra@ifra-virtual-machine: ~/os
Type here to search
22°C
2:19 pm
11/10/2023

```
ifra@ifra-virtual-machine:~/os$ gcc client.c -o c
client.c: In function ‘main’:
client.c:45:9: warning: implicit declaration of function ‘gets’; did you mean ‘f
gets’? [-Wimplicit-function-declaration]
   45 |         gets(client_message);
      |         ^~~~
      |         fgets
client.c:70:9: warning: implicit declaration of function ‘clos
pclose’? [-Wimplicit-function-declaration]
   70 |         close(socket_desc);
      |         ^~~~~
      |         pclose
/usr/bin/ld: /tmp/ccRSKz1l.o: in function `main':
client.c:(.text+0xf9): warning: the `gets' function is dangero
be used.
ifra@ifra-virtual-machine:~/os$ ./c
Socket Created
Enter Message: 21-7508-CI
Server Message: WELCOME STUDENT!
ifra@ifra-virtual-machine:~/os$ ./c
Socket Created
Enter Message: 21-7508-CO
Server Message: Good Bye!!! Have a Nice Day
ifra@ifra-virtual-machine:~/os$
```

```
ifra@ifra-virtual-machine:~$ cd os
ifra@ifra-virtual-machine:~/os$ gcc server.c -o s
server.c: In function ‘main’:
server.c:153:3: warning: implicit declaration of function ‘close’; did you mean
‘pclose’? [-Wimplicit-function-declaration]
  153 |   close(socket_desc);
      |   ^~~~~
      |   pclose
ifra@ifra-virtual-machine:~/os$ ./s
Socket Created
Bind Done
Listening for Messages...

Received Message from IP: 127.0.0.1 and Port No: 43421
Client Message: 21-7508-CI
Received Message from IP: 127.0.0.1 and Port No: 43131
Client Message: 21-7508-CO
```



ifra@ifra-virtual-machine: ~/os
ifra@ifra-virtual-machine: ~/os
Type here to search
22°C
2:19 pm
11/10/2023

```
   45 |         gets(client_message);
      |         ^~~~
      |         fgets
client.c:70:9: warning: implicit declaration of function ‘close’; did you mean ‘
pclose’? [-Wimplicit-function-declaration]
   70 |         close(socket_desc);
      |         ^~~~~
      |         pclose
/usr/bin/ld: /tmp/ccRSKz1l.o: in function `main':
client.c:(.text+0xf9): warning: the `gets' function is dangero
be used.
ifra@ifra-virtual-machine:~/os$ ./c
Socket Created
Enter Message: 21-7508-CI
Server Message: WELCOME STUDENT!
ifra@ifra-virtual-machine:~/os$ ./c
Socket Created
Enter Message: 21-7508-CO
Server Message: Good Bye!!! Have a Nice Day
ifra@ifra-virtual-machine:~/os$ ./c
Socket Created
Enter Message: 21-1781-CO
Server Message: you did not check in today
ifra@ifra-virtual-machine:~/os$
```

```
ifra@ifra-virtual-machine:~$ cd os
ifra@ifra-virtual-machine:~/os$ gcc server.c -o s
server.c: In function ‘main’:
server.c:153:3: warning: implicit declaration of function ‘close’; did you mean
‘pclose’? [-Wimplicit-function-declaration]
  153 |   close(socket_desc);
      |   ^~~~~
      |   pclose
ifra@ifra-virtual-machine:~/os$ ./s
Socket Created
Bind Done
Listening for Messages...

Received Message from IP: 127.0.0.1 and Port No: 43421
Client Message: 21-7508-CI
Received Message from IP: 127.0.0.1 and Port No: 43131
Client Message: 21-7508-CO
Received Message from IP: 127.0.0.1 and Port No: 59198
Client Message: 21-1781-CO
```