2CS502-CN **Practical-10**

Name:- Avi Tayal **Roll No.:-20BCE024**

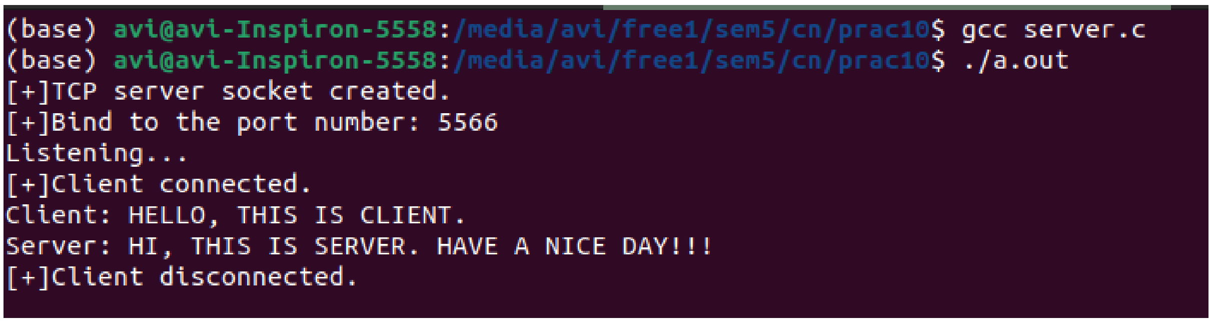
Aim: **Implement an echo client server using TCP/UDP sockets.**

TCP_socket :

Code of tcp_server:-

```c
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>


int main(){


 char *ip = "127.0.0.1";

 int port = 5566;


 int server_sock, client_sock;

 struct sockaddr_in server_addr, client_addr;

 socklen_t addr_size;

 char buffer[1024];

 int n;


 server_sock = socket(AF_INET, SOCK_STREAM, 0);

 if (server_sock < 0){

   perror("[-]Socket error");

   exit(1);

 }
 printf("[+]TCP server socket created.\n");


 memset(&server_addr, '\0', sizeof(server_addr));
 server_addr.sin_family = AF_INET;
 server_addr.sin_port = port;
 server_addr.sin_addr.s_addr = inet_addr(ip);


 n = bind(server_sock, (struct sockaddr*)&server_addr,
sizeof(server_addr));
 if (n < 0){
   perror("[-]Bind error");
   exit(1);
 }
 printf("[+]Bind to the port number: %d\n", port);


 listen(server_sock, 5);
 printf("Listening...\n");


 while(1){
   addr_size = sizeof(client_addr);
   client_sock = accept(server_sock, (struct sockaddr*)&client_addr,
&addr_size);
   printf("[+]Client connected.\n");


   bzero(buffer, 1024);
   recv(client_sock, buffer, sizeof(buffer), 0);
   printf("Client: %s\n", buffer);


   bzero(buffer, 1024);
   strcpy(buffer, "HI, THIS IS SERVER. HAVE A NICE DAY!!!");
   printf("Server: %s\n", buffer);

   send(client_sock, buffer, strlen(buffer), 0);


   close(client_sock);

   printf("[+]Client disconnected.\n\n");


 }


 return 0;

}
```

Output:



```
(base) avi@avi-Inspiron-5558:/media/avi/free1/sem5/cn/prac10$ gcc server.c
(base) avi@avi-Inspiron-5558:/media/avi/free1/sem5/cn/prac10$ ./a.out
[+]TCP server socket created.
[+]Bind to the port number: 5566
Listening...
[+]Client connected.
Client: HELLO, THIS IS CLIENT.
Server: HI, THIS IS SERVER. HAVE A NICE DAY!!!
[+]Client disconnected.
```

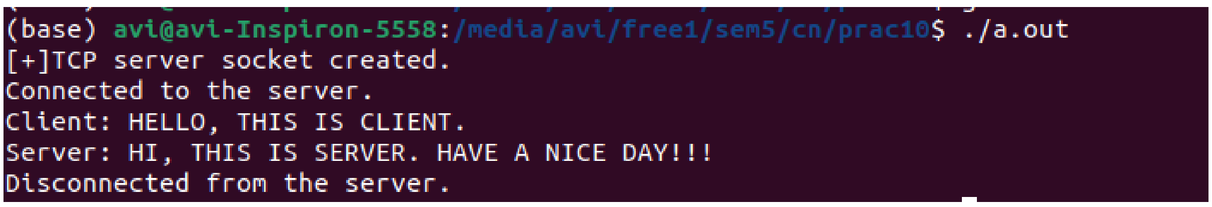
Code of tcp_client:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>


int main(){


 char *ip = "127.0.0.1";
```

```c
int port = 5566;


int sock;

struct sockaddr_in addr;

socklen_t addr_size;

char buffer[1024];

int n;


sock = socket(AF_INET, SOCK_STREAM, 0);

if (sock < 0){

  perror("[-]Socket error");

  exit(1);

}

printf("[+]TCP server socket created.\n");


memset(&addr, '\0', sizeof(addr));

addr.sin_family = AF_INET;

addr.sin_port = port;

addr.sin_addr.s_addr = inet_addr(ip);


connect(sock, (struct sockaddr*)&addr, sizeof(addr));

printf("Connected to the server.\n");


bzero(buffer, 1024);

strcpy(buffer, "HELLO, THIS IS CLIENT.");

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

send(sock, buffer, strlen(buffer), 0);


bzero(buffer, 1024);
```

```
 recv(sock, buffer, sizeof(buffer), 0);

 printf("Server: %s\n", buffer);


 close(sock);

 printf("Disconnected from the server.\n");


 return 0;


}
```

Output:



```
(base) avi@avi-Inspiron-5558:/media/avi/free1/sem5/cn/prac10$ ./a.out
[+]TCP server socket created.
Connected to the server.
Client: HELLO, THIS IS CLIENT.
Server: HI, THIS IS SERVER. HAVE A NICE DAY!!!
Disconnected from the server.
```

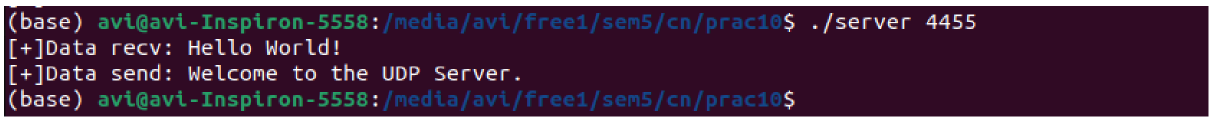
UDP_socket :

Code of udp_server:-

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main(int argc, char **argv){

 if (argc != 2) {

   printf("Usage: %s <port>\n", argv[0]);

   exit(0);
 }

 char *ip = "127.0.0.1";

 int port = atoi(argv[1]);

 int sockfd;

 struct sockaddr_in server_addr, client_addr;

 char buffer[1024];

 socklen_t addr_size;

 int n;

 sockfd = socket(AF_INET, SOCK_DGRAM, 0);

 if (sockfd < 0) {

   perror("[-]socket error");

   exit(1);

 }

 memset(&server_addr, '\0', sizeof(server_addr));

 server_addr.sin_family = AF_INET;

 server_addr.sin_port = htons(port);

 server_addr.sin_addr.s_addr = inet_addr(ip);

 n = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));

 if (n < 0){

   perror("[-]bind error");

   exit(1);

 }

 bzero(buffer, 1024);

 addr_size = sizeof(client_addr);

 recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr,
&addr_size);

 printf("[+]Data recv: %s\n", buffer);

 bzero(buffer, 1024);

 strcpy(buffer, "Welcome to the UDP Server.");
```

```
 sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr,
sizeof(client_addr));

 printf("[+]Data send: %s\n", buffer);

 return 0;

}
```

Output:



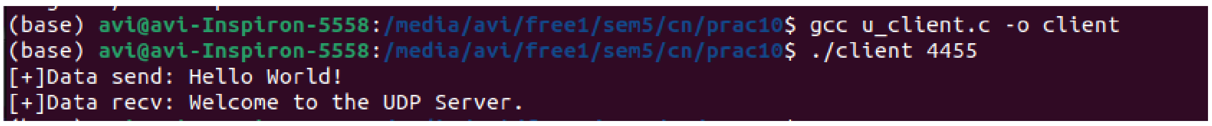
```
(base) avi@avi-Inspiron-5558:/media/avi/free1/sem5/cn/prac10$ ./server 4455
[+]Data recv: Hello World!
[+]Data send: Welcome to the UDP Server.
(base) avi@avi-Inspiron-5558:/media/avi/free1/sem5/cn/prac10$
```

Code of udp_client:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main(int argc, char **argv){

 if (argc != 2) {

   printf("Usage: %s <port>\n", argv[0]);

   exit(0);

 }

 char *ip = "127.0.0.1";

 int port = atoi(argv[1]);

 int sockfd;

 struct sockaddr_in addr;

 char buffer[1024];

 socklen_t addr_size;

 sockfd = socket(AF_INET, SOCK_DGRAM, 0);
```

```c
 memset(&addr, '\0', sizeof(addr));

 addr.sin_family = AF_INET;

 addr.sin_port = htons(port);

 addr.sin_addr.s_addr = inet_addr(ip);

 bzero(buffer, 1024);

 strcpy(buffer, "Hello World!");

 sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr,
sizeof(addr));

 printf("[+]Data send: %s\n", buffer);

 bzero(buffer, 1024);

 addr_size = sizeof(addr);

 recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr,
&addr_size);

 printf("[+]Data recv: %s\n", buffer);

 return 0;

}
```

Output:



```
(base) avi@avi-Inspiron-5558:/media/avi/free1/sem5/cn/prac10$ gcc u_client.c -o client
(base) avi@avi-Inspiron-5558:/media/avi/free1/sem5/cn/prac10$ ./client 4455
[+]Data send: Hello World!
[+]Data recv: Welcome to the UDP Server.
```