

Computer Networks Lab

Week-5

Aug 23, 2021

Venkata Naga Sai Ram Nomula

RA1911033010021

L2 - SWE

Aim: To implement a TCP/IP day time server (concurrent server) that handles multiple client requests. Once the client establishes connection with the server, the server sends its day-time details to the client which the client prints in its console.

Procedure:

Server:

Include the necessary header files.

- Create a socket using socket function with family AF_INET, type as SOCK_STREAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin_family to AF_INET, sin_addr to INADDR_ANY, sin_port to statically assigned port number.
- Bind the local host address to socket using the bind function.
- Within a for loop, accept connection request from the client using accept function.
- Use the fork system call to spawn the processes.
- Calculate the current date and time using the ctime() function. Change the format so that it is appropriate for human readable form and send the date and time to the client using the write function.

Client:

- Include the necessary header files.
- Create a socket using socket function with family AF_INET, type as SOCK_STREAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin_family to AF_INET.
- Get the server IP address from the console.

- Using gethostbyname function assign it to a hostent structure, and assign it to sin_addr of the server address structure.
- Request a connection from the server using the connect function.
- Within an infinite loop, receive the date and time from the server using the read function and print the date and time on the console.

Code:

Server: server.c

```
#include<time.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<unistd.h>
#include<stdio.h>
#include<string.h>
#include<netinet/in.h>
#include<netdb.h>
int main(int argc,char *argv[])
{
    int sd,ad;
    char buff[1024];
    struct sockaddr_in servaddr,cliaddr;
    //socklen_t clilen=sizeof(cliaddr);
    time_t t1;
    bzero(&servaddr,sizeof(servaddr));
    /*Socket address structure*/
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
    servaddr.sin_port=htons(1507);
    /*TCP socket is created, an Internet socket address structure is filled with
    wildcard address & server's well known port*/
    sd=socket(AF_INET,SOCK_STREAM,0);
    /*Bind function assigns a local protocol address to the socket*/
    bind(sd,(struct sockaddr*)&servaddr,sizeof(servaddr));
    /*Listen function specifies the maximum number of connections that kernel should
    queue
```

```

for this socket*/
listen(sd,5);
printf("Server is running...\n");
/*The server to return the next completed connection from the front of the
completed connection Queue calls it*/
ad=accept(sd,(struct sockaddr *)NULL,NULL);
while(1)
{
bzero(&buff,sizeof(buff));
/*Library function time returns the Coordinated Universal Time*/ t1=time(NULL);
/*Prints the converted string format*/
snprintf(buff,sizeof(buff),"%24s\r\n",ctime(&t1));
send(ad,buff,sizeof(buff),0);
}

```

Client: client.c

```

#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netdb.h>
#include<netinet/in.h>
#include<unistd.h>
#include<time.h>
int main(int argc,char *argv[])
{
int sd,ad;
char buff[1024];
struct sockaddr_in cliaddr,servaddr;
struct hostent *h;
h=gethostbyname(argv[1]);
bzero(&servaddr,sizeof(servaddr));
/*Socket address structure*/
servaddr.sin_family=AF_INET;
memcpy((char*)&servaddr.sin_addr.s_addr,h->h_addr_list[0],h->h_length);
servaddr.sin_port=htons(1507);

```

```

/*TCP socket is created, an Internet socket address structure is filled with
wildcard address & server's well known port*/
sd=socket(AF_INET,SOCK_STREAM,0);
/*Connect establishes connection with the server using server IP address*/
connect(sd,(struct sockaddr*)&servaddr,sizeof(servaddr));
recv(sd,buff,sizeof(buff),0);
printf("Day time of server is: %s\n",buff);
}

```

Output:

The screenshot displays a code editor with two files: `client.c` and `server.c`. The `client.c` file contains the following code:

```

1 #include<stdio.h>
2 #include<sys/types.h>
3 #include<sys/socket.h>
4 #include<netdb.h>
5 #include<string.h>
6 #include<netinet/in.h>
7 #include<unistd.h>
8 #include<time.h>
9 int main(int argc,char *argv[])
10 {
11     int sd,ad;
12     char buff[1024];
13     struct sockaddr_in cliaddr,servaddr;
14     struct hostent *h;
15     h=gethostbyname(argv[1]);
16     bzero(&servaddr,sizeof(servaddr));
17     /*Socket address structure*/
18     servaddr.sin_family=AF_INET;
19     memcpy((char*)&servaddr.sin_addr.s_addr,h->h_addr_list[0],h->h_length);
20     servaddr.sin_port=htons(1507);
21     /*TCP socket is created, an Internet socket address structure is filled
22     wildcard address & server's well known port*/
23     sd=socket(AF_INET,SOCK_STREAM,0);
24     /*Connect establishes connection with the server using server IP address
25     */
26     connect(sd,(struct sockaddr*)&servaddr,sizeof(servaddr));
27     recv(sd,buff,sizeof(buff),0);
28     printf("Day time of server is: %s\n",buff);
29 }

```

The `server.c` file contains the following code:

```

1 #include<time.h>
2 #include<sys/types.h>
3 #include<sys/socket.h>
4 #include<unistd.h>
5 #include<stdio.h>
6 #include<string.h>
7 #include<netinet/in.h>
8 #include<netdb.h>
9 int main(int argc,char *argv[])
10 {
11     int sd,ad;
12     char buff[1024];
13     struct sockaddr_in servaddr,cliaddr;
14     //socklen_t clien=sizeof(cliaddr);
15     time_t t1;
16     bzero(&servaddr,sizeof(servaddr));
17     /*Socket address structure*/
18     servaddr.sin_family=AF_INET;
19     servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
20     servaddr.sin_port=htons(1507);
21     /*TCP socket is created, an Internet socket address structure is filled
22     wildcard address & server's well known port*/ sd=socket(AF_INET,SOCK_ST
23     /*Bind function assigns a local protocol address to the socket*/
24     bind(sd,(struct sockaddr*)&servaddr,sizeof(servaddr));
25     /*Listen function specifies the maximum number of connections that kern
26     for this socket*/
27     listen(sd,5);
28     printf("Server is running...\n");

```

Below the code editor, there are two terminal windows. The left terminal shows the execution of the client program:

```

bash - "ip-172-31-9-200" x
RA1911033010017:~/environment/RA1911033010021/Day time UDP $ clear
RA1911033010017:~/environment/RA1911033010021/Day time UDP $ cc server.c
RA1911033010017:~/environment/RA1911033010021/Day time UDP $ ./a.out
Server is running...
RA1911033010017:~/environment/RA1911033010021/Day time UDP $

```

The right terminal shows the execution of the server program:

```

bash - "ip-172-31-9-200" x
RA1911033010017:~/environment/RA1911033010021/Day time UDP $ cc client.c
RA1911033010017:~/environment/RA1911033010021/Day time UDP $ ./a.out 127.0.0.1
1
Day time of server is: Mon Aug 23 05:39:25 2021
RA1911033010017:~/environment/RA1911033010021/Day time UDP $

```