

صورت پروژه های سوکت :

در تمامی پروژه ها ما یک سرور و یک (یا چندین) کلاینت می توانیم داشته باشیم . سرور باید وظیفه سرویس دهی به کلاینت ها را بر عهده داشته باشد . به این صورت که ابتدا یک مشخصه سوکت تعریف می شود . سپس به این مشخصه ، یک آدرس IP و یک پورت نسبت داده می شود و پس از این سرور از این پورت به خط گوش داده تا کلاینتی اقدام به برقراری ارتباط کند . در صورتی که از این پورت کلاینتی تقاضای ارتباط کرد ، سرور آن را توسط ACCEPT قبول کرده و از این پس می تواند به کلاینت خدمات دهد .

- 1- چت : در این روش باید از سیستم تعریف سوکت آسنکرون کمک گرفت . یعنی اینکه نباید سرور یک کلاینت را قبول کرده و تازمانی که ارتباط برقرار است بقیه ارتباطات با خود را به صف انتظار ببرد . بلکه باید بتواند دو یا چندین ارتباط را به طور همزمان قبول کند . حال سرور در بین این چند کلاینت نقش واسط را دارد . ما می توانیم توسط سرور ، کلاینت هایی که به آن مرتبط شده اند را ببینیم و برای آن کلاینت پیغام بفرستیم و سرور در این بین پیغام یک کلاینت را دریافت کرده و روی کلاینت دوم می فرستد . یعنی یک کلاینت به کلاینت دیگر مستقیم پیغام نمی فرستد .
- 2- FTP : در این روش ما یک سرور داریم که سرویس دهنده فایل است . یعنی کلاینت بعد از اتصال به سرور می تواند فایل های روی آن را مشاهده کند و هرکدام را که خواست دانلود نماید و یا فایلی را روی سرور در مسیر دلخواه آپلود کند .

کدهای زیر طریقه اتصال کلاینت و سرور را در دو کد C# و PHP نشان می دهد :

```
// compile code with visual C#
// serverside code
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.Net.Sockets;

namespace serverside
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Server Runing...");
            Socket mysock = new Socket(AddressFamily.InterNetwork,
            SocketType.Stream, ProtocolType.Tcp);
            IPEndPoint ipe = new IPEndPoint(IPAddress.Any, 8888);
            mysock.Bind(ipe);
            mysock.Listen(10);
            mysock = mysock.Accept();
            while (true)
            {
                try
                {
                    byte[] buffer = new byte[500];
                    mysock.Receive(buffer);
                    string data = Encoding.ASCII.GetString(buffer);
                    Console.WriteLine("message from client : " + data + " ");
                    string str;
                    str = Console.ReadLine();
                    byte[] sev = Encoding.ASCII.GetBytes(str);
                    mysock.Send(sev);
                }
            }
        }
    }
}
```



```
$host = "127.0.0.1";

$port = 9898;

$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not create
socket\n");

socket_set_option($socket, SOL_SOCKET, SO_REUSEADDR, 1);

socket_bind($socket, $host, $port);

socket_listen($socket);

while(true){

$spawn = socket_accept($socket);

$data = socket_read($spawn, 10240) or die("Could not read input\n");

$load = file_get_contents('test.txt');

file_put_contents('test.txt',$load.$data);

}

socket_close($socket);

?>

// Client Side

<?php

set_time_limit(0);

$host = "127.0.0.1";

$port = 9898;

$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP) or die("Could not
create socket\n");

$connect = socket_connect($socket, $host, $port);

$msg = " this is a test ";

socket_write($socket, $msg, strlen($msg));

socket_close($socket);
```

```
print file_get_contents('test.txt');
```

```
?>
```