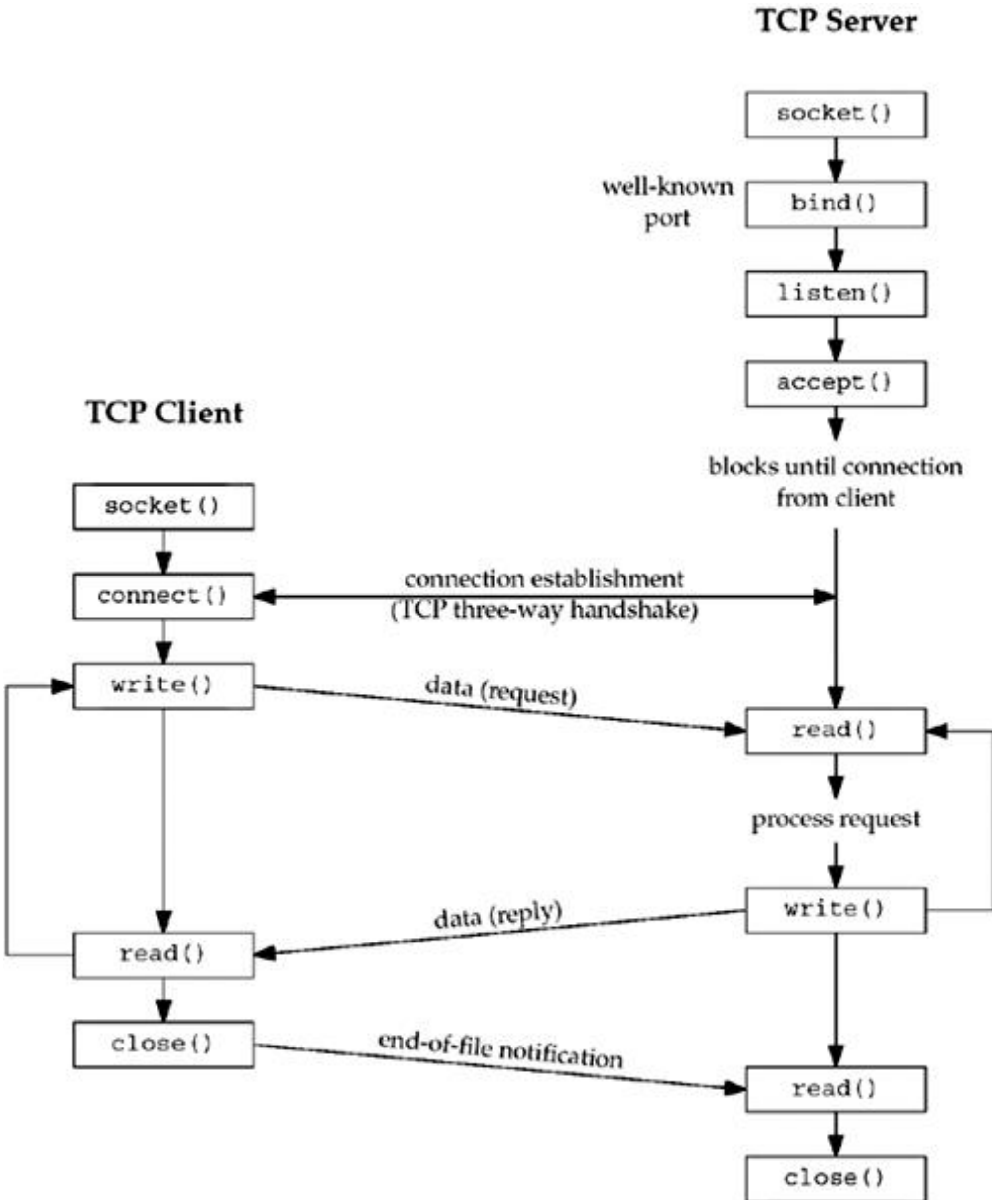
Client/Server programming using TCP



Instructions-1 You have to always include at least these header files in most of the client/ server programs

```
#include<sys/types.h>

#include<sys/socket.h>
```

```
#include<netinet/in.h>

#include<stdlib.h>

#include<netdb.h>

#include<stdio.h>
```

Instructions-2 See the code for the file sum.h

```
struct args {

   long arg1;

   long arg2;


    };

  struct result {

    long sum;

      };
```

Instruction-3 See the code for client’s site and understand.

```
#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<stdlib.h>

#include<netdb.h>

#include<stdio.h>

#define maxline 4096

#include "sum.h"


void str_cli(FILE *,int);

int

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

```
 {
    int sockfd,n,cn;
    char recvline[maxline+1];
    struct sockaddr_in servaddr;

  if(argc !=2)
  printf("usage: a.out <IPaddress>");
  if((sockfd=socket(AF_INET,SOCK_STREAM,0))<0)
   printf("socket error");

bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_port=htons(4500);

if(inet_pton(AF_INET,argv[1],&servaddr.sin_addr)<=0)
  printf("inet_pton error for %s", argv[1]);

cn=connect(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr));
if(cn < 0)
  {
  printf("\nconnect error\n");
  exit(0);
 }
else
  printf("\nconnection has been estabished");

 str_cli(stdin,sockfd);
 exit(0);
}

/* code for str_cli() function  */

void str_cli(FILE *fp,int sockfd)
 {
      char sendline[maxline],recvline[maxline];
      struct args args;
      struct result result;

      while(fgets(sendline,maxline,fp)!=NULL)
       {
if(sscanf(sendline,"%ld %ld", &args.arg1, &args.arg2) !=2)
         {
          printf("\ninvlid input:%S\n",sendline);
          continue;
       }

 printf("the values are %ld%ld\n",args.arg1,args.arg2);
   write(sockfd,&args,sizeof(args));

   if(read(sockfd,&result,sizeof(result))==0)
    printf("error");

     printf("socket desc read is %d\n",&sockfd);
    printf("%ld\n",result.sum);
}

}
```

Instruction-4 See the code for server's site and understand

```
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<netdb.h>
#include<stdio.h>

#include<time.h>
#define maxline 4096
#define LISTENQ 5
#include "sum.h"

void str_echo(int);
int main(int argc,char *argv)
 {
  int listenfd,connfd,bd;
  pid_t childpid;

  struct sockaddr_in servaddr;
  char buff[maxline];
  time_t ticks;
```

```
if(( listenfd=socket(AF_INET,SOCK_STREAM,0))<0)

  {

  printf("error");

  }


bzero(&servaddr,sizeof(servaddr));


servaddr.sin_family=AF_INET;

servaddr.sin_addr.s_addr=htonl(INADDR_ANY);

servaddr.sin_port=htons(4500);


if((bd=bind(listenfd,(struct
sockaddr*)&servaddr,sizeof(servaddr)))<0)

 {

  printf("\nbind error");

  exit(0);

 }


listen(listenfd,LISTENQ);


for(;;){

 connfd=accept(listenfd,(struct sockaddr*)NULL,NULL);

   if((childpid=fork())==0) {

   close(listenfd);

   str_echo(connfd);

   exit(0);
}

close(connfd);

}

}



/* code for str_echo function   */


  void str_echo(int sockfd)

    {



      ssize_t n;

      struct args args;

      struct result result;

      char  line[maxline];

     for(;;) {

    n= read(sockfd,&args,sizeof(args));

    if(n==0)

    return;

printf("the  server are %ld%ld\n",args.arg1,args.arg2);

  result.sum=args.arg1 + args.arg2;

   // printf("the value returned by server is %ld\n ",result.sum);

//   break;




//result.sum = args.arg1 - args.arg2;
```

```
//break;



          write(sockfd,&result,sizeof(result));

  }


}
```