Exercise 6:

Server:

```
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<string.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<unistd.h>
#define SA struct sockaddr
#define PORT 8080
int main()
{
      int i,sd,l,ad,len,b,f;
      char msg[100],df[100],str1[100];
      struct sockaddr_in server,client;
      server.sin_family=AF_INET;
      server.sin_port=htons(8127);
      server.sin_addr.s_addr=htonl(INADDR_ANY);
      sd=socket(AF_INET,SOCK_STREAM,0);
      if(sd>0)
            printf("Socket created\n");
      else
            printf("Socket error\n");
      b=bind(sd,(struct sockaddr*)&server,sizeof(server));
      if(b>=0)
            printf("Bind created\n");
      else
            printf("Bind error\n");
      l=listen(sd,5);
      if(l==0)
            printf("Listen\n");
      else
            printf(" Error in listen\n");
      len=sizeof(client);
      for( i=0;i<3;i++)
      {
            printf("\n");
            ad=accept(sd,(struct sockaddr*)&client,&len);
            if(ad>0)
                  printf("Accept sucess\n");
            else
                  printf("Accept error\n");
            printf("Connection Established\n");
            // do concurrent process
            f=fork();
            if(f==0) // Child Process
            {
                  if(i==0) // Client 1
```

```
                {
                        recv(ad,msg,100,0);
                        strcpy(str1,msg);
                        printf("Client 1:%s",msg);
                }
                if(i==1) // Client 2
                {
                        recv(ad,msg,100,0);
                        printf("Client 2:%s",msg);
                }
                if(i==2) // Client 3
                {
                        strcat(str1,msg);
                        send(ad,str1,100,0);
                        printf("Client 3:%s",str1);
                }
        }
        if(f>0)
                wait();
    }
        close(sd);
}
```

Client 1:

```
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<string.h>
#include<sys/socket.h>
#include<sys/types.h>
#define SA struct sockaddr
#define PORT 8080
int main()
{
    int sd,c;
    char msg[100];
    struct sockaddr_in server,client;
    server.sin_family=AF_INET;
    server.sin_port=htons(8127);
    server.sin_addr.s_addr=htonl(INADDR_ANY);
    sd=socket(AF_INET,SOCK_STREAM,0);
    if(sd>0)
        printf("Socket created\n");
    else
        printf("Socket error\n");
    c=connect(sd,(struct sockaddr*)&server,sizeof(server));
    if(c==0)
        printf("Connected\n");
    else
```

```
            printf("Notconnected\n");
      printf("Connection Established\n");
      printf("\nenter a string:");
      scanf("%s",msg);
      send(sd,msg,100,0);
      close(sd);
}
```

client 2:

```
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<string.h>
#include<sys/socket.h>
#include<sys/types.h>
#define SA struct sockaddr
int main()
{
      int sd,c;
      char msg[100];
      struct sockaddr_in server,client;
      server.sin_family=AF_INET;
      server.sin_port=htons(8127);
      server.sin_addr.s_addr=htonl(INADDR_ANY);
      sd=socket(AF_INET,SOCK_STREAM,0);
      if(sd>0)
            printf("Socket created\n");
      else
            printf("Socket error\n");
      c=connect(sd,(struct sockaddr*)&server,sizeof(server));
      if(c==0)
            printf("Connected\n");
      else
            printf("Notconnected\n");
      printf("Connection Established\n");
      printf("\nenter a String:");
      scanf("%s",msg);
      send(sd,msg,100,0);
      close(sd);
}
```

client 3:

```
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<string.h>
```

```
#include<sys/socket.h>
#include<sys/types.h>
#define SA struct sockaddr
int main()
{
      int sd,c;
      char msg[100];
      struct sockaddr_in server,client;
      server.sin_family=AF_INET;
      server.sin_port=htons(8127);
      server.sin_addr.s_addr=htonl(INADDR_ANY);
      sd=socket(AF_INET,SOCK_STREAM,0);
      if(sd>0)
            printf("Socket created\n");
      else
            printf("Socket error\n");
      c=connect(sd,(struct sockaddr*)&server,sizeof(server));
      if(c==0)
            printf("Connected\n");
      else
            printf("Notconnected\n");
      printf("Connection Established\n");
      recv(sd,msg,100,0);
      printf("String concatenation:%s\n",msg);
      close(sd);
}
```