



# Socket Programming

A Socket is a :-

- It is an API
- It is an end-end connection point between two machines
- Socket address is a combination of IP address and the port number

# The Socket Interface

socket  
interface

Application  
1

user

kern  
el

**Socket**

Underlying  
communicati  
on Protocols

Communication  
s network

Application  
2

user

kern  
el

**Socket**

Underlying  
communicati  
on Protocols

socket  
interface

A decorative graphic on the left side of the slide featuring three balloons: a light green one at the top, a light blue one in the middle, and a light purple one at the bottom. Each balloon has a string and several small yellow triangular flags attached to it.

# Port numbers

Well known ports:-

- 0-1024

Registered ports

- 1024-49151

Dynamic ports

- 49152-65535



# System calls for connection oriented protocol [TCP/IP]

## **Server side system calls :-**

- Socket
- Bind
- Listen
- Accept
- Read
- Write



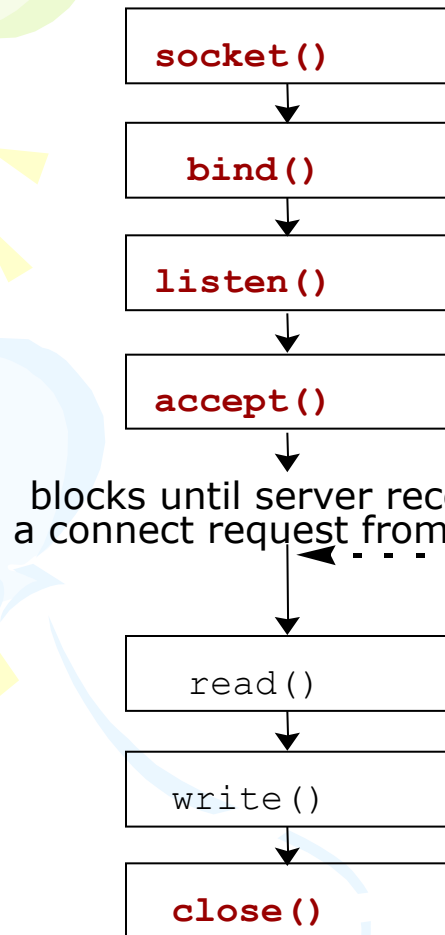
# System calls for connection oriented protocol [TCP/IP]

## **Client side system calls :-**

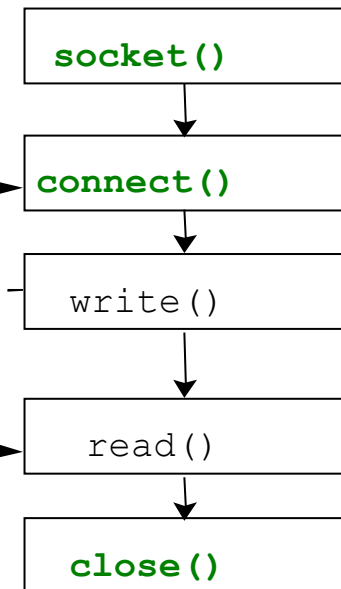
- Socket
- Connect
- Write
- Read

# TCP client-server

Server



Client



connect negotiation

data

data



System calls for connection less  
protocol [UDP/IP]

**Server side system calls :-**

- Socket
- Bind
- Recvfrom
- Sendto



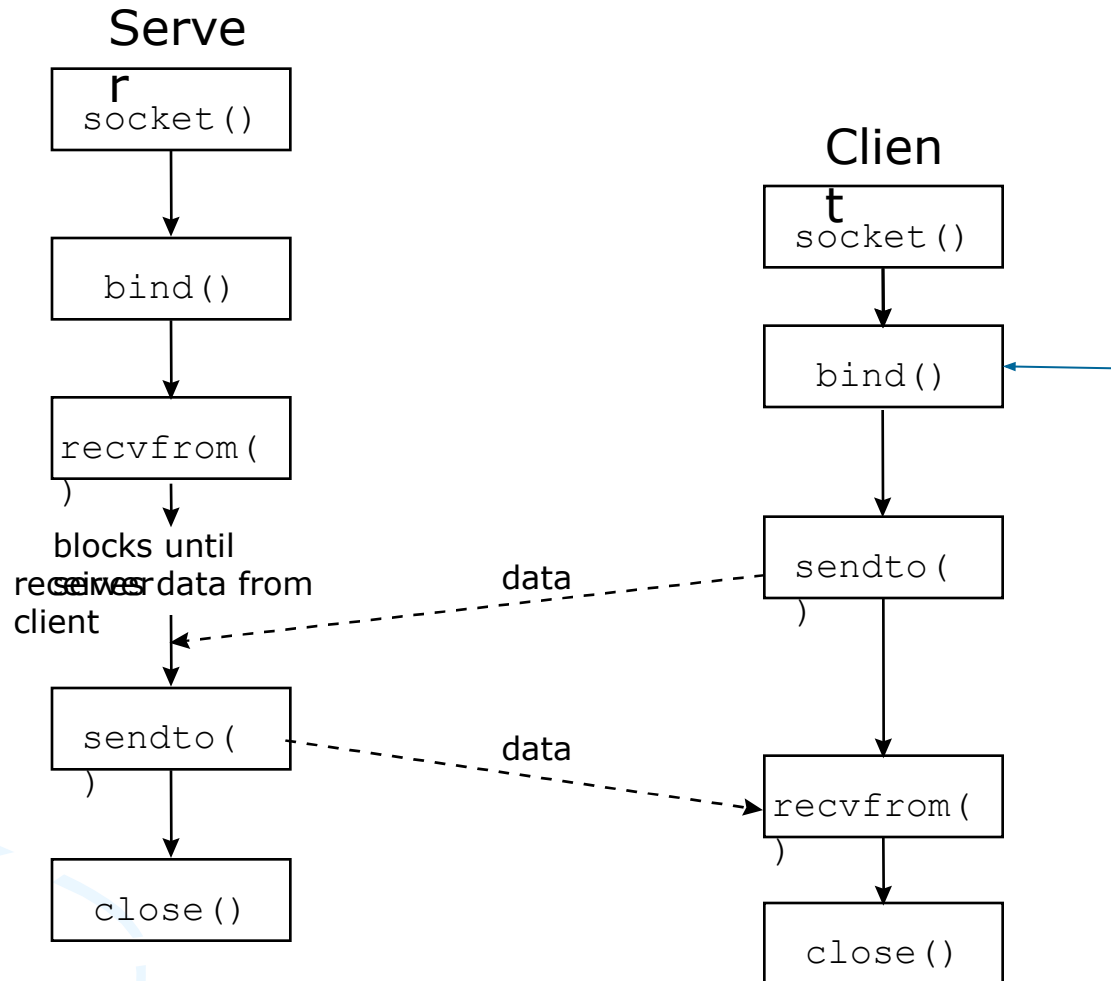
# System calls for connection less protocol [UDP/IP]

## **Client side system calls :-**

- Socket
- Bind
- Sendto
- recvfrom



# UDP Socket Calls



**Not  
needed**



# Socket programming

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

```
struct in_addr {  
    in_addr_t s_addr ;  
};
```

4 bytes in size/32 bit IP address

# Socket programming

```
struct sockaddr_in
{
    uin8_t sin_len; /* structure length ,16 */

    sa_family_t sin_family; /* AF/PF */

    in_port_t sin_port; /* TCP,UDP port,16 bit TCP/UDP
    port    number */

    struct in_addr sin_addr; /* 32 bit IPv4 addr */

    char sin_zero[8]; /* unused */
};
```



# Socket programming

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

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

```
int socket(int family,int type,int  
protocol );
```

Descriptor if OK,-1 on error.

Family : AF/PF

Type : SOCK\_STREAM,SOCK\_DGRAM,SOCK\_RAW

Protocol :TCP/IP or UDP/IP



# Socket programming

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

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

```
int bind (int sockfd, struct sockaddr  
*myaddr, int addrlen);
```

0:OK and -1:error

Assigns local protocol address to a  
socket, (IP address + port number).



# Socket programming

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

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

```
int connect (int sockfd, struct sockaddr  
             *servaddr, socklen_t addrlen);
```

- For connecting to the server.



# Socket programming

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

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

```
int listen (int sockfd,int backlog);
```

```
0:OK,-1:error
```

Listen converts an unconnected socket to a passive socket...



# Socket programming

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

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



```
int accept (int sockfd, struct sockaddr  
*peer, int *addrlen);
```

It creates a new socket, to which the properties of old socket are passed.





# Socket programming

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

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

```
int close (int fd);
```

Three balloons (green, blue, and purple) are positioned vertically on the left side of the slide. Each balloon has a small yellow starburst graphic next to it. The green balloon is at the top, the blue one in the middle, and the purple one at the bottom. The blue balloon has a long, thin blue string trailing off to the right.

# Socket programming

## Threads