Motivation of Sockets

- TCP, UDP, IP and related protocols are complex
- Application programmer should not have to worry about TCP/IP protocol details
- Provide simple and common interface to send and receive data from application
    - Send: write()
    - Receive: read()
- Sockets programming constructs provide this interface
- Common across operating systems and programming languages

TCP Stream Sockets

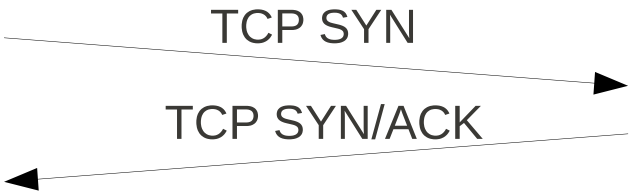
- Client

1. Create socket
2. Connect to server

- Server

1. Create socket
2. Bind socket to server address
3. Specify number of connections to listen
4. Wait to accept new connection

   (block)



(return from accept)

Client

Server

Create a socket

```
Socket_ID=
socket(address_type,socket_type,protocol)
```

Bind the socket to an address

```
bind(Socket_ID,address,address_size)
```

Listen for connections

```
listen(Socket_ID,queued_connections)
```

Create a socket

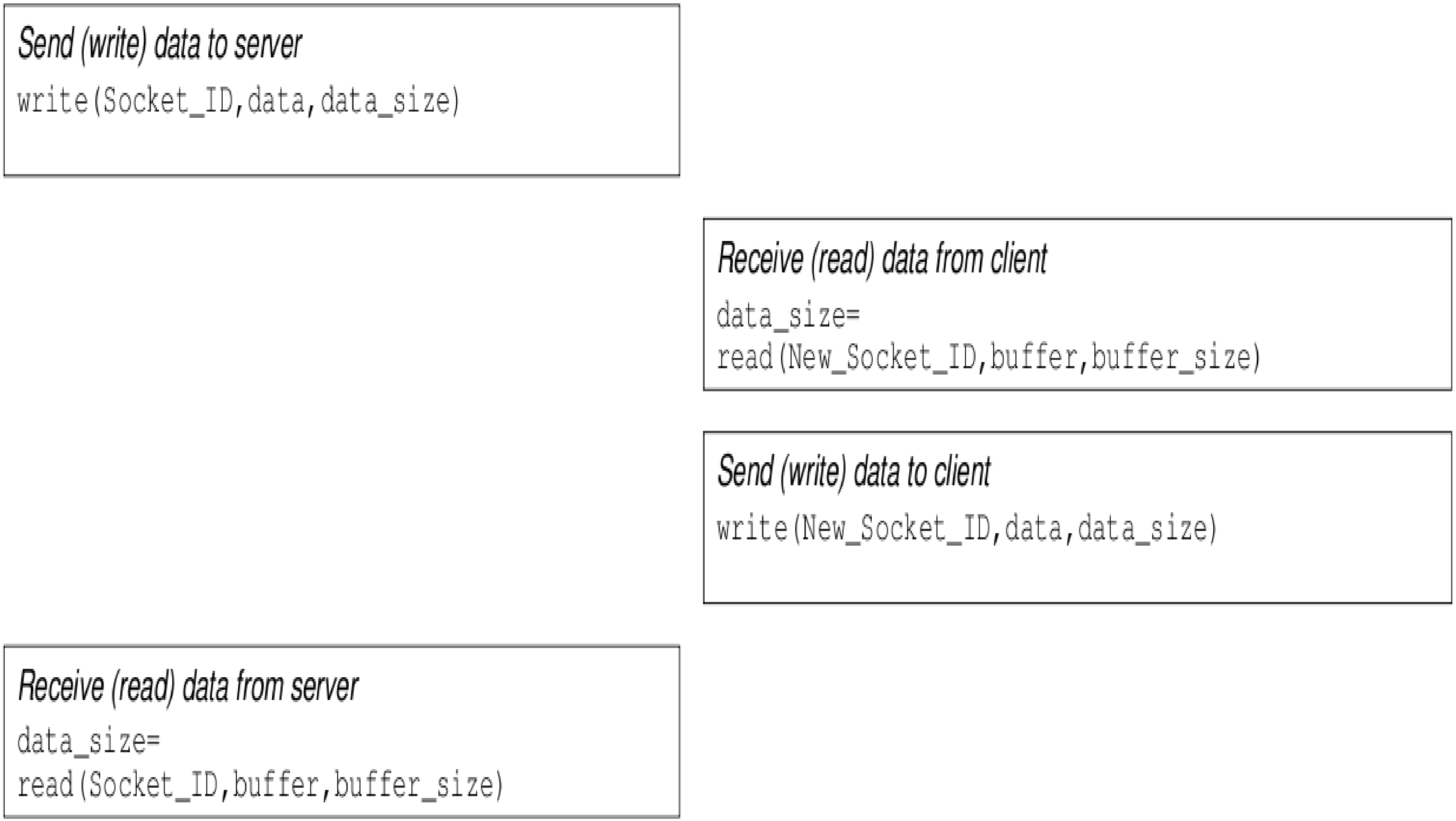
```
Socket_ID=
socket(address_type,socket_type,protocol)
```

Connect to server

```
connect(Socket_ID,server_address,
          server_address_size)
```

Accept a new connection from client

```
New_Socket_ID=
accept(Socket_ID,&client_add,&client_add_size)
```



Send (write) data to server

```
write(Socket_ID,data,data_size)
```

Receive (read) data from client

```
data_size=
read(New_Socket_ID,buffer,buffer_size)
```

Send (write) data to client

```
write(New_Socket_ID,data,data_size)
```

Receive (read) data from server

```
data_size=
read(Socket_ID,buffer,buffer_size)
```