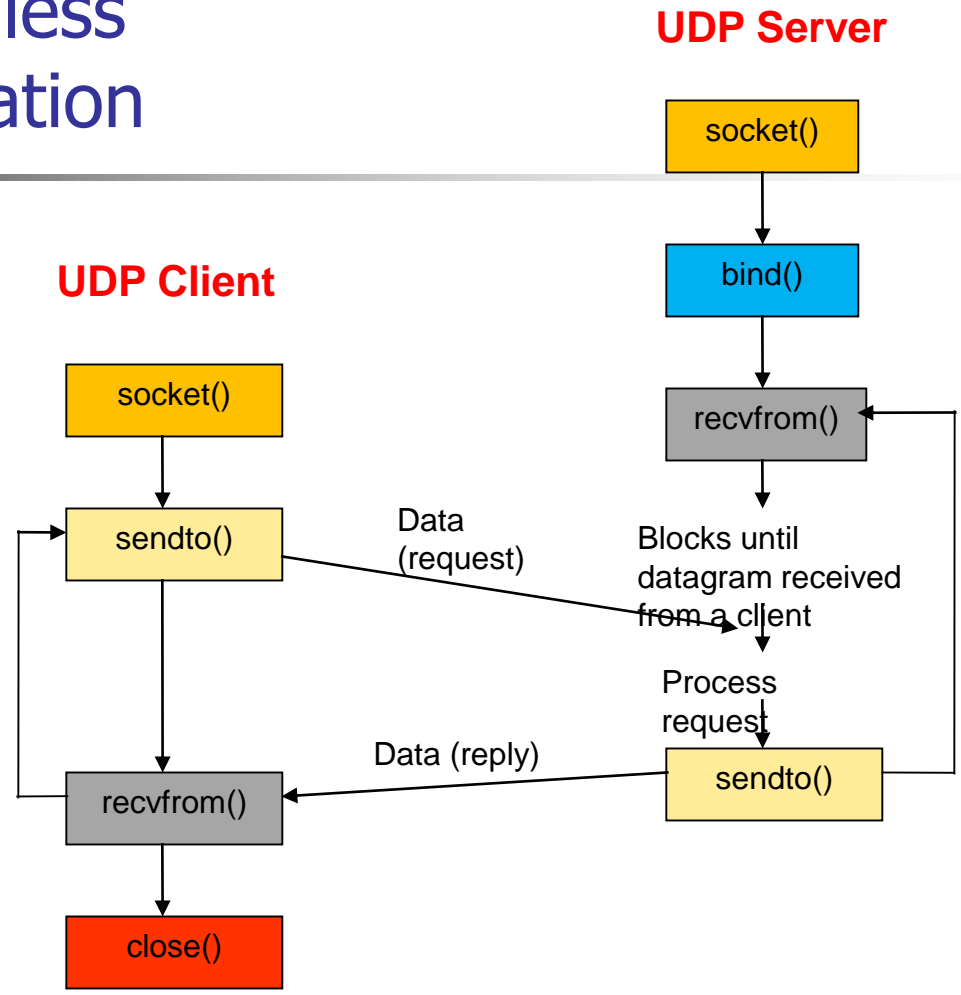




# Socket Programming – UDP client server

---

# Socket functions for connectionless communication





# Socket()

---

- `int s = socket(domain, type, protocol);`

where

- `s`: socket descriptor, an integer (like a file-handle) and -1 on error
- `domain`: integer, communication domain
  - e.g., `AF_INET` (IPv4 protocol)
  - **Note. We'll use `AF_INET`**
- `type`: communication type
  - `SOCK_STREAM`: reliable, 2-way, connection-based service
  - `SOCK_DGRAM`: unreliable, connectionless
  - **Note. We'll use `SOCK_DGRAM`**
- `protocol`: **We'll set to 0**



# Bind()

- The bind function assigns a local protocol address to a socket.
  - The protocol address is the combination of either a **32-bit IPV4 address** or a **128-bit IPV6 address**, along with a **16-bit port number**

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

```
int bind(int sockfd, struct sockaddr *address, int addr len)
```

- sockfd: a socket descriptor returned by the socket()
- \*address: a pointer to a protocol-specific address.
- addrlen: the size of the socket address structure
  
- Returns on success: 0, on error: -1



# Recvfrom()

---

- This function is similar to the read function, but additional arguments are required

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

```
int recvfrom(int sockfd, void *buff, size_t nbyte, int flag, struct  
sockaddr *from, socklen_t *addrlen);
```

- sockfd – socket descriptor
- \*buff – pointer to buffer to read.
- nbytes – number of bytes to read.
- flag – **We'll set to 0**
- from - socket address structure of who sent the datagram
- addrlen – size of the socket address structure

- Returns: number of bytes read if OK,-1 on error



# Sendto()

- This function is similar to the write function, but additional arguments are required

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

```
int sendto(int sockfd, const void *buff, size_t nbyte, int flag, const struct  
sockaddr *to, socklen_t addrlen);
```

- sockfd – socket descriptor
- \*buff – pointer to buffer to write from
- nbytes – number of bytes to write.
- flag – **We'll set to 0**
- to – socket address structure containing the protocol address of where the data is to be sent
- addrlen – size of the socket address structure
  
- Returns: number of bytes written if OK,-1 on error



# Close()

---

- The close function is used to close a socket and terminate a connection

```
#include <unistd.h>
```

```
int close (int sockfd);
```

- sockfd: This socket descriptor is no longer useable
- Returns on success: 0, on error: -1