

IN 3230 / IN 4230

Oracle Session – Week 1

Praveensankar Manimaran

02/09/2021

Outline

1. Socket introduction and Unix sockets
2. Chat Application using unix socket

Section 1 – Socket Introduction and Unix Sockets

What is Socket ?

1. It's an endpoint used by processes to communicate with other processes
2. Each process will be identifier using a port number

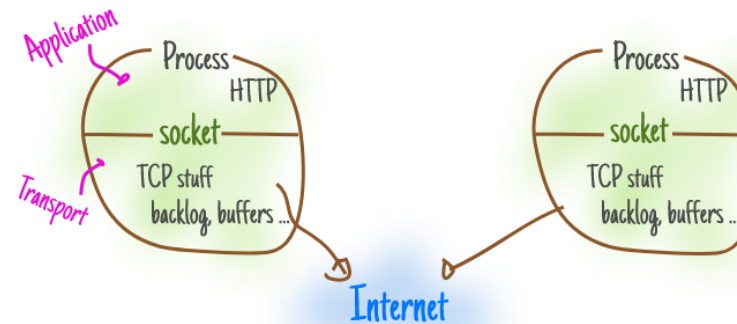


Fig: socket [1]

[1] <https://ops.tips/blog/a-tcp-server-in-c/>

What is Socket ?

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

Domain:

- AF_INET – IPV4 protocol
- AF_INET6 – IPV6 protocol
- AF_UNIX – local communication (AF_LOCAL)

Refer the following link for more info:

<https://man7.org/linux/man-pages/man2/socket.2.html>

What is Socket ?

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

Type:

- **SOCK_STREAM** - sequenced, reliable, two-way, connection-based byte streams.
- **SOCK_DGRAM** - Supports datagrams (connectionless, unreliable messages of a fixed maximum length)
- **SOCK_SEQPACKET** - Provides a sequenced, reliable, two-way connection-based for datagrams
- **SOCK_RAW** - Provides raw network protocol access

Protocol:

- 0 – to use default protocol

Refer the following link for more info:

<https://man7.org/linux/man-pages/man2/socket.2.html>

Unix Sockets

- sockets for local interprocess communication

```
struct sockaddr_un
{
    sa_family_t sun_family;    /* AF_UNIX */
    char sun_path[108];       /* Pathname */
};
```

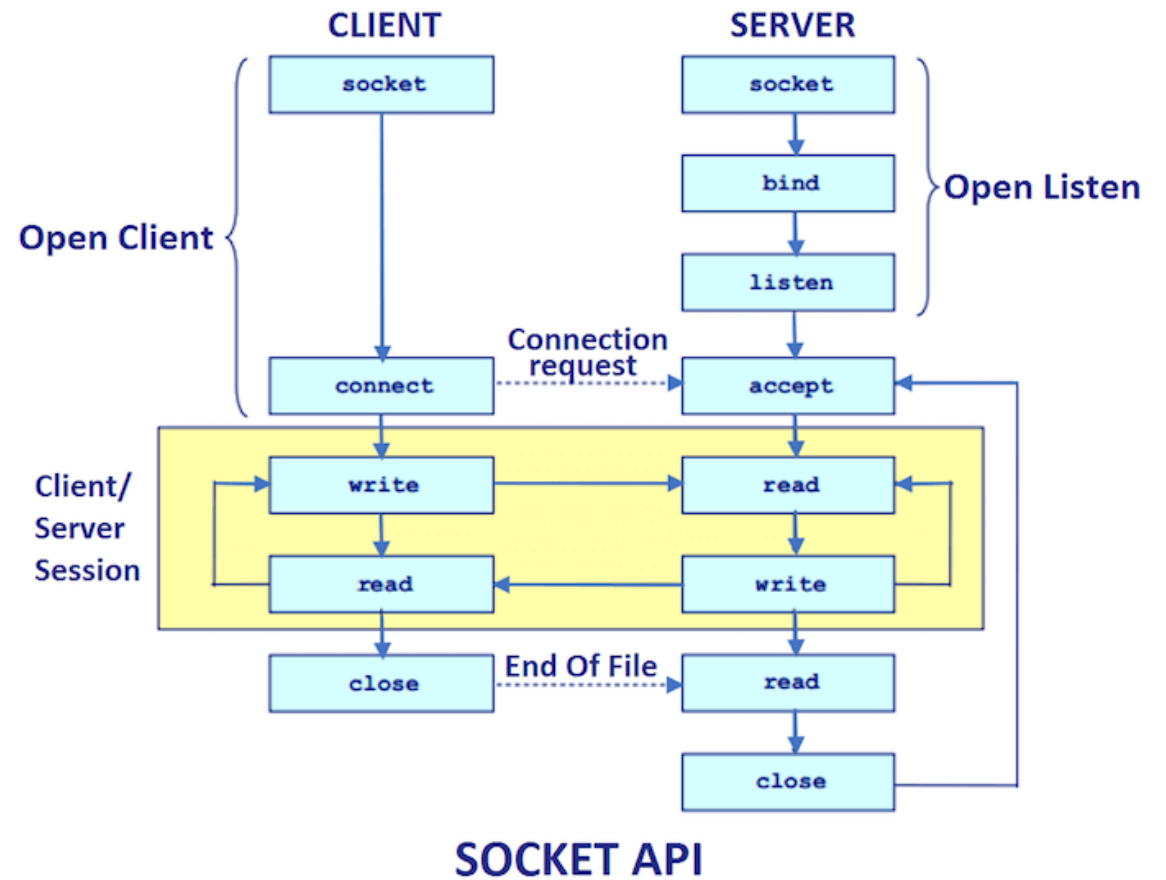
Refer the following link for more info:

<https://man7.org/linux/man-pages/man7/unix.7.html>

Section 2 – Chat Application

Socket Programming Flow

- Domain – AF_UNIX
- Protocol - SOCK_STREAM



[2] <https://www.javatpoint.com/socket-programming>

Fig: SOCKET API [2]

Server

1 . Socket creation:

```
server_fd = socket(AF_UNIX, SOCK_STREAM, 0);
```

2 . Binding to pathname:

```
server_address.sun_family = AF_UNIX;  
strncpy(server_address.sun_path, SOCKET_NAME, sizeof(server_address.sun_path)-1);  
bind(server_fd, (const struct sockaddr *)&server_address, sizeof(server_address));
```

Server

3. Listen:

```
listen(server_fd, 1);
```

4 accept:

```
client_fd = accept(server_fd, NULL, NULL);
```

5 Read/Write:

```
write(client_fd, msg, strlen(msg));  
Read(client_fd, msg, strlen(msg));
```

6 Close:

```
Close(server_fd);
```

Client

1 Socket Creation:

```
fd = socket(AF_UNIX, SOCK_STREAM, 0);
```

2 connect:

```
connect(fd, (struct sockaddr *) &server_address, sizeof(server_address));
```

3 Read/Write:

```
write(fd, msg, strlen(msg));
```

```
Read(fd, msg, strlen(msg));
```

6 Close:

```
Close(fd);
```

Examples

1. <https://man7.org/linux/man-pages/man7/unix.7.html>
2. <https://github.com/praveensankar/IN3230>
3. Linux man page for the system calls and commands
4. Maybe some random online tutorials and blogs !



Thank You