Socket programming
Software Technologies - Lecture 6

Saurabh Barjatiya

International Institute Of Information Technology, Hyderabad

22 February, 2010

Contents

Introduction

- TCP is connection oriented, reliable transport layer protocol.
- There are two types of TCP sockets server socket and client socket.
- Server socket wait for client to connect.

Important functions

socket

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

Header files: sys/types.h, sys/socket.h

socket() is used to create a OS socket. This returns a file descriptor on which we can use standard read/write functions after connection is established. This function is called by both TCP server and client as both need socket for communication.

Important functions

htonl

uint32_t htonl(uint32_t hostlong);

Header files: arpa/inet.h

htonl() is used to convert from host specific storage format (little endian or big endian) to network byte order. This is very important so that hosts manufactured by different vendors and having different architectures can communicate without any problem.

Important functions

htons

uint16_t htons(uint16_t hostshort);

Header files: arpa/inet.h

htons() is used to convert from host specific storage format (little endian or big endian) to network byte order. This is very important so that hosts manufactured by different vendors and having different architectures can communicate without any problem.

Important functions

bind

int bind(int sockfd, const struct sockaddr *my_addr, socklen_t addrlen);

Header files: sys/types.h, sys/socket.h

bind() is used to indicate that the socket would listen on particular address and particular port. All server sockets must first bind to a interface address and port number. This is not required for client sockets.

Important functions

listen

int listen(int sockfd, int backlog);

Header files: sys/socket.h

listen() is used to start listening on already binded server socket. With the help of listen we can define queue length (backlog) on how many client connections should be kept queued for accept.

Important functions

accept

int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);

Header files: sys/types.h, sys/socket.h

accept() function call can be used to receive file descriptor for client which is connected to server socket on which the program is already listening. In case there is no client in queue the accept() function call will sleep() till the client connects to the socket.

Important functions

inet_pton

int inet_pton(int af, const char *src, void *dst);

Header files: sys/types.h, sys/socket.h, arpa/inet.h

inet_pton() converts address in C style string (char *) to struct in_addr format, so that we can assign it to sockaddr_in.sin_addr.

Important functions

connect

int connect(int sockfd, const struct sockaddr *serv_addr, socklen_t addrlen);

Header files: sys/types.h, sys/socket.h

connect() function can be used to connect socket, created with socket() function to server and port specified in serv_addr structure.

Important structures

struct sockaddr_in

```
struct sockaddr_in
{
    sa_family_t     sin_family;
    in_port_t       sin_port;
    struct in_addr  sin_addr;
};
```

Important structures

struct in_addr

```
typedef uint32_t in_addr_t;
struct in_addr
{
        in_addr_t s_addr;
};
```

Important structures

struct sockaddr

```
struct sockaddr
{
    sa_family_t  sa_family;
    char         sa_data[];
};
```

Contents

Introduction

- UDP is connection-less oriented, unreliable transport layer protocol.
- In UDP we have only one type of socket and we can send/receive messages using it.
- UDP send will succeed even if there is no server listening on other end.

Important functions

recvfrom

ssize_t recvfrom(int s, void *buf, size_t len, int flags,
struct sockaddr *from, socklen_t *fromlen);

Header files: sys/types.h, sys/socket.h

recvfrom() can be used to receive data from both connection oriented and connectionless sockets.

Important functions

sendto

ssize_t sendto(int s, const void *buf, size_t len,
int flags, const struct sockaddr *to, socklen_ttolen);

Header files: sys/types.h, sys/socket.h

sendto() can used to send messages on both connection oriented and connection less sockets. In connection oriented sockets the to parameters will get ignored.

Contents

Introduction

- Unix supports filesystem sockets for communication within machine.
- Filesystem sockets can be created and deleted in locations where we can create and delete files.
- Filesystem sockets are used according to normal file protections hence provide secure means of interprocess communication for processes belonging to same user without any kind of authentication.
- Unix sockets cannot be accessed from different machine, in this aspect they are very different from traditional TCP/IP sockets.

Important structures

struct sockaddr_un

```
struct sockaddr_un
{
    short sun_family;
    char  sun_PATH[108];
};
```