Socket prog

- Struct sockaddr{
    unsigned short sa_family;
    sa_data[14];
  }

- Struct sockaddr_in{
    short sin_family;
  us short sin_port; //port #
  Struct in_addr sin_addr; // IP addr
    char sin_zero[8];
  }

- Struct in_addr{
    unsigned long s_addr; //4B
  }

- Network data sent in bigendian. This is called network byte order.
- Host can have any endiness & is called Host byte order. (~~HOD~~) (HBO)
- We convert Host Byte Order to N/w Byte order before sending. (NBO)

HBO to NBO : htons() (short), htonl() (long)
NBO to HBO : ntohs(), ntohl()

- IP address: ASCII dotted to Binary (network): inet_aton()
  Binary (network) to ASCII dotted : inet_ntoa()

PAGE NO.
DATE: / /

socket()

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

domain ⇒ PF_INET (IPV4)
type ⇒ SOCK_STREAM (TCP), SOCK_DGRAM (UDP)
protocol ⇒ 0

eg: Socket (PF_INET, SOCK_STREAM, 0);
return val: socket descriptor, -1 if error

Bind() : attatch port #, IP addr

int Bind(int sockfd, struct sockaddr *my_addr,
sock_len addrlen)
sockfd → socket descriptor ret by socket()
my_addr → pointer to sock_addr pointer. It is easy to create sockaddr_in pointer & cast it to sockaddr type.
addrlen → length of sockaddr_in structure.

eg
```
struct sockaddr_in my;
my.sin_family = PF_INET;
my.sin_port = htons(80)
my.sin_addr.s_addr = INADDR_ANY;
bzero(&my, 8)        → padding

bind(sock, (struct sockaddr *) &my, sizeof(my));
```
(socket desc)

Listen() – initialize wait queue of connections.

int Listen (int sock, int backlog)
↓
sockfd

backlog → length of pending connection queue.
eg Listen (sock, 10)
↳ max 10 pending conn.

Accept() – Used to accept conn req by client

int Accept (int socket, (struct sockaddr *) &client,
socklen_t *client_len)

socket → socket descriptor
client → Passed as empty struct of type sockaddr & accept fills it with client info.
client_len → pointer to size of client structure.

eg struct sockaddr_in client
int len = sizeof (client);

Accept (sock, (struct sockaddr *) &client, &len);

Connect() – Used by client to connect to server.

int Connect (int sockfd, (struct sockaddr *) & server_addr
, socklen_t len)

server_addr → a sockaddr_in struct having details of server.
len → size of server_addr struct.
eg: connect (sockfd, (struct sockaddr*) server_addr, len);

PAGE NO.
DATE: / /

send/recv

int send (int sockfd, void *msg, size_t len, int flags)
int recv (int sockfd, void *msg, size_t len, int flags)

msg → Pointer to buffer to send/rcv message to/from
len → Size of msg buffer.
flags → 0

return val → size of ~~data~~ bytes sent/rcv.

eg char send_buf[1024];
char recv_buf[1024];
int sent_size, rcv_size;

send_bytes = send (sockfd, send_buf, 1024, 0)
rcv_bytes = ~~send~~ recv (" " " ")

close () — close conn

int close (int sockfd)

---

pjoin