

CLIENT STUFF

socket()

- Creates a Socket Descriptor

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

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

Parameter	Values
domain	•PF_INET for IPv4 •PF_INET6 for IPv6
type	•SOCK_STREAM •SOCK_DGRAM •SOCK_SEQPACKET •SOCK_RAW
protocol	IPPROTO_IP (defined as 0)

connect()

- Connect to a server located at some address and port

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

```
memset(&my_addr, 0, sizeof(struct sockaddr));
addr.sin_family = AF_INET;
addr.sin_port = htons(PORT);
addr.sin_addr.s_addr = net_addr("127.0.0.1");
```

send() and recv()

```
int send(int sockfd, const void *msg, int len,
int flags);
```

```
int recv(int sockfd, void *buf, int len,
unsigned int flags);
```

CONNECTIONLESS COMMUNICATION

Datagram Send and Receive

```
int sendto(int sockfd, const void *msg, int
len, unsigned int flags, const struct
sockaddr *to, socklen_t tolen);
```

```
int recvfrom(int sockfd, void *buf, int len,
unsigned int flags, struct sockaddr *from,
int *fromlen);
```

DNS

- Domain Name Server
- Resolve a name to an IP address:

<http://www.cs.pitt.edu> -> 130.49.220.23

DNS

```
#include <netdb.h>
struct hostent *gethostbyname(const char *name);

struct hostent {
    char    *h_name;        /* official name of host */
    char    **h_aliases;   /* alias list */
    int     h_addrtype;    /* host address type */
    int     h_length;      /* length of address */
    char    **h_addr_list; /* list of addresses */
}
#define h_addr h_addr_list[0] /* for backward compatibility */
```