socket()

- Creates a Socket Descriptor

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>PF_INET for IPv4</td>
</tr>
<tr>
<td></td>
<td>PF_INET6 for IPv6</td>
</tr>
<tr>
<td>type</td>
<td>SOCK_STREAM</td>
</tr>
<tr>
<td></td>
<td>SOCK_DGRAM</td>
</tr>
<tr>
<td></td>
<td>SOCK_SEQPACKET</td>
</tr>
<tr>
<td></td>
<td>SOCK_RAW</td>
</tr>
<tr>
<td>protocol</td>
<td>IPPROTO_IP (defined as 0)</td>
</tr>
</tbody>
</table>

connect()

- Connect to a server located at some address and port

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

```c
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()

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

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

Datagram Send and Receive

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

```c
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 \rightarrow 130.49.220.23

```c
#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 */
```