Sockect:

SYNOPSIS

#include <sys/types.h

#include <sys/socket.h>

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

The socket has the indicated type, which specifies the communication semantics. Currently defined types are:

SOCK_STREAM Provides sequenced, reliable, two-way, connection-based byte streams. An out-of-band data transmission mechanism may be supported.

SOCK_DGRAM Supports datagrams (connectionless, unreliable messages of a fixed maximum length).

SOCK_SEQPACKET Provides a sequenced, reliable, two-way connection-based data transmission path for datagrams of fixed maximum length; a consumer is required to read an entire packet with each input system call.

SOCK_RAW Provides raw network protocol access.

SOCK_RDM Provides a reliable datagram layer that does not guarantee ordering.

SOCK_PACKET Obsolete and should not be used in new programs

Bind

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

When a socket is created with socket(2), it exists in a name space (address family) but has no address assigned to it. bind() assigns the address specified by addr to the socket referred to by the file descriptor sockfd.

The actual structure passed for the addr argument will depend on the address family. The sockaddr structure is defined as something like:

```
struct sockaddr {
    sa_family_t sa_family;
    char        sa_data[14];
}
```

RETURN VALUE

On success, zero is returned. On error, -1 is returned

Listen

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
int listen(int sockfd, int backlog);
```

RETURN VALUE

On success, zero is returned. On error, -1 is returned

listen() marks the socket referred to by sockfd as a passive socket, that is, as a socket that will be used to accept incoming connection requests using accept

accept

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);

#define _GNU_SOURCE             /* See feature_test_macros(7) */
#include <sys/socket.h>
int accept4(int sockfd, struct sockaddr *addr, socklen_t *addrlen, int flags);
```

The following values can be bitwise ORed in flags to obtain different behavior:

SOCK_NONBLOCK Set the O_NONBLOCK file status flag on the new open file description. Using this flag saves extra calls to fcntl(2) to achieve the same result.

SOCK_CLOEXEC Set the close-on-exec (FD_CLOEXEC) flag on the new file descriptor. See the description of the O_CLOEXEC flag in open(2) for reasons why this may be useful.

RETURN VALUE

On success, these system calls return a nonnegative integer that is a file descriptor for the accepted socket. On error, -1 is returned

connect

SYNOPSIS

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

#include <sys/socket.h>


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

The connect() system call connects the socket referred to by the file descriptor sockfd to the address specified by addr.

RETURN VALUE

If the connection or binding succeeds, zero is returned. On error, -1