SYNOPSIS

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

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

If successful, `socket` returns a nonnegative integer corresponding to a socket file
descriptor. If unsuccessful, `socket` returns -1 and sets `errno`. The following table lists the
mandatory errors for `socket`.

<table>
<thead>
<tr>
<th>errno</th>
<th>cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAFNOSUPPORT</td>
<td>implementation does not support specified address family</td>
</tr>
<tr>
<td>EMFILE</td>
<td>no more file descriptors available for process</td>
</tr>
<tr>
<td>ENFILE</td>
<td>no more file descriptors available for system</td>
</tr>
<tr>
<td>EPROTOTONOSUPPORT</td>
<td>protocol not supported by address family or by implementation</td>
</tr>
<tr>
<td>EPROTOTYPE</td>
<td>socket type not supported by protocol</td>
</tr>
</tbody>
</table>

Example 18.22

The following code segment sets up a socket communication endpoint for Internet
communication, using a connection-oriented protocol.

```c
int sock;

if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    perror("Failed to create socket");
```

18.7.2 The bind function

The `bind` function associates the handle for a socket communication endpoint with a spec-
cific logical network connection. Internet domain protocols specify the logical connection
by a port number. The first parameter to `bind`, `socket`, is the file descriptor returned by
a previous call to the `socket` function. The *address structure contains a family name
and protocol-specific information. The `address_len` parameter is the number of bytes in
the *address structure.

SYNOPSIS

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

int bind(int socket, const struct sockaddr *address,
          socklen_t address_len);
```

If successful, `bind` returns 0. If unsuccessful, `bind` returns -1 and sets `errno`. The fol-
lowing table lists the mandatory errors for `bind` that are applicable to all address families.