5.4 TCP Echo Client: `main` Function

Figure 5.4 shows the TCP client `main` function.

Figure 5.3 `str_echo` function: echoes data on a socket.

lib/str_echo.c

```
 1 #include    "unp.h"

 2 void
 3 str_echo(int sockfd)
 4 {
 5     ssize_t n;
 6     char    buf[MAXLINE];

 7   again:
 8     while ( (n = read(sockfd, buf, MAXLINE)) > 0)
 9         Writen(sockfd, buf, n);

10     if (n < 0 && errno == EINTR)
11         goto again;
12     else if (n < 0)
13         err_sys("str_echo: read error");
14 }
```

Figure 5.4 TCP echo client.

tcpcliserv/tcpli01.c

```
 1 #include    "unp.h"

 2 int
 3 main(int argc, char **argv)
 4 {
 5     int     sockfd;
 6     struct sockaddr_in servaddr;

 7     if (argc != 2)
 8         err_quit("usage: tcpcli <IPaddress>");

 9     sockfd = Socket(AF_INET, SOCK_STREAM, 0);

10     bzero(&servaddr, sizeof(servaddr));
11     servaddr.sin_family = AF_INET;
12     servaddr.sin_port = htons(SERV_PORT);
13     Inet_pton(AF_INET, argv[1], &servaddr.sin_addr);

14     Connect(sockfd, (SA *) &servaddr, sizeof(servaddr));

15     str_cli(stdin, sockfd);     /* do it all */

16     exit(0);
17 }
```

Create socket, fill in Internet socket address structure

9–13 A TCP socket is created and an Internet socket address structure is filled in with the server's IP address and port number. We take the server's IP address from the command-line argument and the server's well-known port (`SERV_PORT`) is from our `unp.h` header.

Connect to server

14–15 `connect` establishes the connection with the server. The function `str_cli` (Figure 5.5) handles the rest of the client processing.