[Team LiB]

5.5 TCP Echo Client: `str_cli` Function

This function, shown in Figure 5.5, handles the client processing loop: It reads a line of text from standard input, writes it to the server, reads back the server's echo of the line, and outputs the echoed line to standard output.

Figure 5.5 `str_cli` function: client processing loop.

lib/str_cli.c

```
 1 #include    "unp.h"

 2 void
 3 str_cli(FILE *fp, int sockfd)
 4 {
 5     char    sendline[MAXLINE], recvline[MAXLINE];

 6     while (Fgets(sendline, MAXLINE, fp) != NULL) {

 7         Writen(sockfd, sendline, strlen (sendline));

 8         if (Readline(sockfd, recvline, MAXLINE) == 0)
 9             err_quit("str_cli: server terminated prematurely");

10         Fputs(recvline, stdout);
11     }
12 }
```

Read a line, write to server

6–7 `fgets` reads a line of text and `writen` sends the line to the server.

Read echoed line from server, write to standard output

8–10 `readline` reads the line echoed back from the server and `fputs` writes it to standard output.

Return to `main`

11–12 The loop terminates when `fgets` returns a null pointer, which occurs when it encounters either an end-of-file (EOF) or an error. Our `Fgets` wrapper function checks for an error and aborts if one occurs, so `Fgets` returns a null pointer only when an end-of-file is encountered.

[Team LiB]