

函数不是线程安全的，这就要求 Stevens 引入一个不同的线程安全的版本，称为 `readline_r`。我们已经在 `rio_readlineb` 和 `rio_readnb` 函数中修改了这两个缺陷，使得这两个函数是相互兼容和线程安全的。

```

code/src/csapp.c
1  ssize_t rio_readlineb(rio_t *rp, void *usrbuf, size_t maxlen)
2  {
3      int n, rc;
4      char c, *bufp = usrbuf;
5
6      for (n = 1; n < maxlen; n++) {
7          if ((rc = rio_read(rp, &c, 1)) == 1) {
8              *bufp++ = c;
9              if (c == '\n') {
10                 n++;
11                 break;
12             }
13         } else if (rc == 0) {
14             if (n == 1)
15                 return 0; /* EOF, no data read */
16             else
17                 break; /* EOF, some data was read */
18         } else
19             return -1; /* Error */
20     }
21     *bufp = 0;
22     return n-1;
23 }

```

```

code/src/csapp.c
code/src/csapp.c
1  ssize_t rio_readnb(rio_t *rp, void *usrbuf, size_t n)
2  {
3      size_t nleft = n;
4      ssize_t nread;
5      char *bufp = usrbuf;
6
7      while (nleft > 0) {
8          if ((nread = rio_read(rp, bufp, nleft)) < 0)
9              return -1; /* errno set by read() */
10         else if (nread == 0)
11             break; /* EOF */
12         nleft -= nread;
13         bufp += nread;
14     }
15     return (n - nleft); /* Return >= 0 */
16 }

```

code/src/csapp.c

图 10-8 `rio_readlineb` 和 `rio_readnb` 函数