

需要初始化 `byte_cnt` 计数器和 `mutex` 信号量。一个方法是我们为 `SBUF` 和 `RIO` 程序包使用过的，它要求主线程显式地调用一个初始化函数。另外一个方法，在此显示的，是当第一次有某个线程调用 `echo_cnt` 函数时，使用 `pthread_once` 函数(第 19 行)去调用初始化函数。这个方法的优点是它使程序包的使用更加容易。这种方法的缺点是每一次调用 `echo_cnt` 都会导致调用 `pthread_once` 函数，而在大多数时候它没有做什么有用的事。

code/conc/echoserv-pre.c

```

1  #include "csapp.h"
2  #include "sbuf.h"
3  #define NTHREADS 4
4  #define SBUFSIZE 16
5
6  void echo_cnt(int connfd);
7  void *thread(void *vargp);
8
9  sbuf_t sbuf; /* Shared buffer of connected descriptors */
10
11 int main(int argc, char **argv)
12 {
13     int i, listenfd, connfd;
14     socklen_t clientlen;
15     struct sockaddr_storage clientaddr;
16     pthread_t tid;
17
18     if (argc != 2) {
19         fprintf(stderr, "usage: %s <port>\n", argv[0]);
20         exit(0);
21     }
22     listenfd = Open_listenfd(argv[1]);
23
24     sbuf_init(&sbuf, SBUFSIZE);
25     for (i = 0; i < NTHREADS; i++) /* Create worker threads */
26         Pthread_create(&tid, NULL, thread, NULL);
27
28     while (1) {
29         clientlen = sizeof(struct sockaddr_storage);
30         connfd = Accept(listenfd, (SA *) &clientaddr, &clientlen);
31         sbuf_insert(&sbuf, connfd); /* Insert connfd in buffer */
32     }
33 }
34
35 void *thread(void *vargp)
36 {
37     Pthread_detach(pthread_self());
38     while (1) {
39         int connfd = sbuf_remove(&sbuf); /* Remove connfd from buffer */
40         echo_cnt(connfd);                /* Service client */
41         Close(connfd);
42     }
43 }

```

code/conc/echoserv-pre.c

图 12-28 一个预线程化的并发 echo 服务器。这个服务器使用的是有一个生产者或多个消费者的生产者-消费者模型