

*code/netp/tiny/tiny.c*

```
1  /*
2   * tiny.c - A simple, iterative HTTP/1.0 Web server that uses the
3   *         GET method to serve static and dynamic content
4   */
5  #include "csapp.h"
6
7  void doit(int fd);
8  void read_requesthdrs(rio_t *rp);
9  int parse_uri(char *uri, char *filename, char *cgiargs);
10 void serve_static(int fd, char *filename, int filesize);
11 void get_filetype(char *filename, char *filetype);
12 void serve_dynamic(int fd, char *filename, char *cgiargs);
13 void clienterror(int fd, char *cause, char *errnum,
14                 char *shortmsg, char *longmsg);
15
16 int main(int argc, char **argv)
17 {
18     int listenfd, connfd;
19     char hostname[MAXLINE], port[MAXLINE];
20     socklen_t clientlen;
21     struct sockaddr_storage clientaddr;
22
23     /* Check command-line args */
24     if (argc != 2) {
25         fprintf(stderr, "usage: %s <port>\n", argv[0]);
26         exit(1);
27     }
28
29     listenfd = Open_listenfd(argv[1]);
30     while (1) {
31         clientlen = sizeof(clientaddr);
32         connfd = Accept(listenfd, (SA *)&clientaddr, &clientlen);
33         Getnameinfo((SA *)&clientaddr, clientlen, hostname, MAXLINE,
34                     port, MAXLINE, 0);
35         printf("Accepted connection from (%s, %s)\n", hostname, port);
36         doit(connfd);
37         Close(connfd);
38     }
39 }
```

*code/netp/tiny/tiny.c*

图 11-29 TINY Web 服务器

然后，我们将 URI 解析为一个文件名和一个可能为空的 CGI 参数字符串，并且设置一个标志，表明请求的是静态内容还是动态内容（第 23 行）。如果文件在磁盘上不存在，我们立即发送一个错误信息给客户端并返回。

最后，如果请求的是静态内容，我们就验证该文件是一个普通文件，而我们是有限权限的（第 31 行）。如果是这样，我们就向客户端提供静态内容（第 36 行）。相似地，如果请求的是动态内容，我们就验证该文件是可执行文件（第 39 行），如果是这样，我们就继续，并且提供动态内容（第 44 行）。