

它使用 Linux `dup2` 函数将标准输出重定向到和客户端相关联的已连接描述符。因此，任何 CGI 程序写到标准输出的东西都会直接到达客户端。

注意，因为父进程不知道子进程生成的内容的类型或大小，所以子进程就要负责生成 `Content-type` 和 `Content-length` 响应报头，以及终止报头的空行。

图 11-27 展示了一个简单的 CGI 程序，它对两个参数求和，并返回带结果的 HTML 文件给客户端。图 11-28 展示了一个 HTTP 事务，它根据 `adder` 程序提供动态内容。

code/netp/tiny/cgi-bin/adder.c

```

1  #include "csapp.h"
2
3  int main(void) {
4      char *buf, *p;
5      char arg1[MAXLINE], arg2[MAXLINE], content[MAXLINE];
6      int n1=0, n2=0;
7
8      /* Extract the two arguments */
9      if ((buf = getenv("QUERY_STRING")) != NULL) {
10         p = strchr(buf, '&');
11         *p = '\0';
12         strcpy(arg1, buf);
13         strcpy(arg2, p+1);
14         n1 = atoi(arg1);
15         n2 = atoi(arg2);
16     }
17
18     /* Make the response body */
19     sprintf(content, "QUERY_STRING=%s", buf);
20     sprintf(content, "Welcome to add.com: ");
21     sprintf(content, "%sTHE Internet addition portal.\r\n<p>", content);
22     sprintf(content, "%sThe answer is: %d + %d = %d\r\n<p>",
23             content, n1, n2, n1 + n2);
24     sprintf(content, "%sThanks for visiting!\r\n", content);
25
26     /* Generate the HTTP response */
27     printf("Connection: close\r\n");
28     printf("Content-length: %d\r\n", (int)strlen(content));
29     printf("Content-type: text/html\r\n\r\n");
30     printf("%s", content);
31     fflush(stdout);
32
33     exit(0);
34 }

```

code/netp/tiny/cgi-bin/adder.c

图 11-27 对两个整数求和的 CGI 程序