

Problem 1

What is TLS? Please describe its usage.

Problem 2

Refer to the I/O multiplexing based concurrent echo server in Figure 12.8, indicate the pros and cons of this event-driven programming approach based on I/O multiplexing.

Problem 3

Modify Tiny so that when it serves static content, it copies the requested file to the connected descriptor using `calloc`, `rio_readn`, `rio_writen`, instead of `mmap` and `rio_writen`.

Problem 4

Consider the code in Figure 12.5:

```

1  #include "csapp.h"
2  void echo(int connfd);
3
4  void sigchld_handler(int sig) {
5      while(waitpid(-1, 0, WNOHANG) > 0);
6      return;
7  }
8
9  int main(int argc, char **argv) {
10     int listenfd, connfd, port;
11     socklen_t clientlen = sizeof(struct sockaddr_in);
12     struct sockaddr_in clientaddr;
13
14     if(argc != 2) {
15         fprintf(stderr, "usage: %s <port>\n", argv[0]);
16         exit(0);
17     }
18     port = atoi(argv[1]);
19
20     Signal(SIGCHLD, sigchld_handler);
21     listenfd = Open_listenfd(port);
22     while(1) {
23         connfd = Accept(listenfd, (SA *) &clientaddr, &clientlen);
24         if (Fork() == 0) {
25             Close(listenfd);
26             echo(connfd);
27             Close(connfd);
28             exit(0);
29         }
30         Close(connfd);
31     }
32 }

```

1. After the parent closes the connected descriptor in line 30 of the concurrent server, the child is still able to communicate with the client using its copy of the descriptor. Why?
2. Assume that 12306.cn has used this design to provide services, what will happen when *Chunyun* begins?