这个项目主要的目的是对**浏览器的链接请求进行解析处理，处理完之后给浏览器客户端返回一个响应，如文字图片视频等**。服务器后端的处理方式**使用socket通信，利用多路IO复用**，可以同时处理多个请求，请求的解析使**用预先准备好的线程池**，使用模拟proactor模式，**主线程负责监听**，监听到有事件之后，从socket中循环读取数据，然后将读取到的数据封装成一个请求对象插入请求队列。睡眠在请求队列上的**工作线程被唤醒进行处理**。处理的方式用**状态机**。

对请求文件的处理：客户端发出链接，到达服务器，服务器这端先用read\_once（）函数一次性把所有请求读到缓冲区。然后process\_read函数分别调用用三个函数对缓冲区的内容进行解析。主状态机主要用于解析客户端请求，从状态机用于解析一行内容并把每一行加\0\0格式化，方便主状态机解析，主状态机调用解析请求行，请求头，请求内容三部分函数进行解析。解析结束后利用do\_request（）函数生成响应报文，该函数会根据不同的网址url产生不同响应体。最后通过write函数里套接字的传输方式把响应体传给客户端。

日志分为**同步日志和异步日志**，异步日志利用阻塞队列，先将日志放入阻塞队列中，然后利用条件变量将日志添加到对应文件中。采用**单例模式**。

日志系统初始化函数中主要要做的事：如果文件名没带路径，直接放到log\_full\_name。如果带路径，把文件名取出来放到log\_name，把路径取出来放到dir\_name，然后把时间+log\_name放到log\_full\_name。

在write\_log（）函数中，这里面有两部分，一部分是对新的文件名，时间，日志名进行再次处理，一部分是时间+新加入的日志参数放入缓冲区。异步加入阻塞队列，同步直接写入日志文件。阻塞队列是用数组模拟的生产者消费者模式。根据初始化函数传入的最后一个参数阻塞队列最大容纳值判断是同步还是异步，异步则其大于等于1。

用**单例模式创建了数据库连接池**。数据库链接池中，提前创建一定量的数据链接，并把他们保存在双向链表中。

**用定时器处理非活动链接，定时器容器利用升序链表进行设计**。