文件操作 都是导入fs模块

const fs = require('fs')

fs.readFile('ts.txt', (err,data) => {

    if(err) {

        console.log();

    }

    console.log(data.toString());

})

改写

const fs = require('fs')

fs.writeFile('./ts.txt','hellobyd',() => {

    console.log('file is written');

})

**引入模块**

在 Node.js 中，引入一个模块非常简单，如下我们创建一个 **main.js** 文件并引入 hello 模块，代码如下:

var hello = require('./hello');

hello.world();

以上实例中，代码 require('./hello') 引入了当前目录下的 hello.js 文件（./ 为当前目录，node.js 默认后缀为 js）。

从流中读取数据 好处是可以一边加载一边读取 不用一次性把水全部倒进桶里

var fs = require('fs')

var data = ''

var readerStream = fs.createReadStream('ts.txt')

readerStream.setEncoding('utf-8')

readerStream.on('data',(chunk)=>{

        console.log(chunk);

})

**端口号被占用解决方案：**

1. 在cmd输入netstat -aon|findstr "8080"查看8080端口的所有进程。
2. 输入 taskkill -PID 12396 -F

安装socket

npm i socket.io