Nodejs 的非阻塞 I/O、异步、事件驱动

主讲教师：（大地）

合作网站：www.itying.com

目录

1、 Nodejs 的单线程 非阻塞 I/O 事件驱动

在 Java、PHP 或者.net 等服务器端语言中，会为每一个客户端连接创建一个新的线程。而每个线程需要耗费大约 2MB 内存。也就是说，理论上，一个 8GB 内存的服务器可以同时连接的最大用户数为 4000 个左右。要让 Web 应用程序支持更多的用户，就需要增加服务器的数量，而 Web 应用程序的硬件成本当然就上升了。

Node.js 不为每个客户连接创建一个新的线程，而仅仅使用一个线程。当有用户连接了，就触发一个内部事件，通过非阻塞 I/O、事件驱动机制，让 Node.js 程序宏观上也是并行的。使用 Node.js，一个 8GB 内存的服务器，可以同时处理超过 4 万用户的连接。

2、 Nodejs 回调处理异步

```
//错误的写法：

function getData(){
    //模拟请求数据
    var result='';
```

```
    setTimeout(function(){
        result='这是请求到的数据'

    },200);
    return result;
}

console.log(getData());/*异步导致请求不到数据*/
```

```
//正确的处理异步:
function getData(callback){
    //模拟请求数据
    var result='';
    setTimeout(function(){
        result='这是请求到的数据';
       callback(result);
    },200);
}
getData(function(data){
        console.log(data);
})
```

3、 Nodejs events 模块处理异步

Node.js 有多个内置的事件，我们可以通过引入 events 模块，并通过实例化 EventEmitter 类来绑定和监听事件。

```
// 引入 events 模块
var events = require('events');

var EventEmitter=new events.EventEmitter();  /*实例化事件对象*/

EventEmitter.on('toparent',function(){

    console.log('接收到了广播事件');

})

setTimeout(function(){
```

```
    console.log('广播');
    EventEmitter.emit('toparent');  /*发送广播*/
},1000)
```