|  |
| --- |
|  |
| 软件项目计划书 |
| 基于Node.js的研究与应用 |
|  |
| **温元祯** |
| **2017/3/22** |

|  |
| --- |
|  |
|  |

**文档更改历史记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **初始信息** | | | |
| **文件名称** | 项目计划书 | **编写人** | 温元祯 |
| **初始版本号** | V1.0 | **发布日期** | 2017/3/22 |
| **更改记录** | | | |
| **版本号** | **更改要点** | **修改人** | **修改日期** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 3](#_Toc477974556)

[1.1 目的 3](#_Toc477974557)

[1.2 术语 3](#_Toc477974558)

[1.3 参考资料 3](#_Toc477974559)

[2. 项目概述 3](#_Toc477974560)

[2.1 项目简介 3](#_Toc477974561)

[2.2 Node.js架构简介 4](#_Toc477974562)

[2.3 Node.js核心模块简介 4](#_Toc477974563)

[2.4 产品目标与范围 5](#_Toc477974564)

[2.5 应交付成果 5](#_Toc477974565)

[2.6 项目开发环境 6](#_Toc477974566)

[2.6.1 软件环境 6](#_Toc477974567)

[2.6.2 硬件环境 6](#_Toc477974568)

[2.6.3 开发工具 6](#_Toc477974569)

[2.7 项目验收方式和依据 6](#_Toc477974570)

[2.7.1 验收方式 6](#_Toc477974571)

[2.7.2 验收依据 6](#_Toc477974572)

[3. 项目团队组织 6](#_Toc477974573)

[3.1 组织结构 6](#_Toc477974574)

[3.2 项目分工 6](#_Toc477974575)

[3.3 协作与沟通 6](#_Toc477974576)

[4. 实施计划 7](#_Toc477974577)

[4.1 工作流程 7](#_Toc477974578)

[4.2 总体进度计划 7](#_Toc477974579)

[4.3 项目控制计划 7](#_Toc477974580)

1. **引言**
   1. **目的**

本文档的编写目的是为了协调组内成员开展后期的工作，对项目的进度进行合理的安排和调控。

* 1. **术语**

关于本文档中出现的一些专业术语、缩略语如表1所示。

|  |  |
| --- | --- |
| 术语/缩略语 | 描述 |
| JavaScript | JavaScript一种直译式脚本语言 |
| API | Application Programming Interface，应用程序编程接口 |
| HTTP | HyperText Transfer Protocol，[超文本传输协议](http://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE) |
| libev | 一个[事件驱动](http://baike.baidu.com/subview/536048/536048.htm)的编程框架 |
| libeio | 提供全套一部文件操作的接口 |
| libuv | libuv 是 Node 的新跨平台抽象层 |
| epoll | epoll是[Linux内核](http://baike.baidu.com/item/Linux%E5%86%85%E6%A0%B8)为处理大批量[文件描述符](http://baike.baidu.com/item/%E6%96%87%E4%BB%B6%E6%8F%8F%E8%BF%B0%E7%AC%A6)而作了改进的poll，能显著提高程序在大量[并发连接](http://baike.baidu.com/item/%E5%B9%B6%E5%8F%91%E8%BF%9E%E6%8E%A5)中只有少量活跃的情况下的系统CPU利用率。 |
| kqueue | Kqueue功能同epoll，存在于许多unix系统中。 |
| POSIX | Portable Operating System Interface，一套操作系统API规范 |
| IOCP | Input/Output Completion Port，输入输出完成端口 |

表 1专业术语/缩略图描述表

* 1. **参考资料**

【1】Node.js v6.10.0文档

【2】Node.js开发指南

1. **项目概述**
   1. **项目简介**

Node.js是一个可以让JavaScript运行在服务器端的平台，它可以让JavaScript脱离浏览器的束缚运行在一般的服务器环境下。

Node.js在诞生之初就充分考虑了在事实响应、超大规模数据要求下架构的可扩展性，实现了诸如文件系统、模块、包、操作系统API、网络通信等Core JavaScript没有或者不完善的功能。历史上将JavaScript移植到浏览器外的计划不止一个，但Node.js是最出色的一个。

Node.js的JavaScript引擎是V8,来自Google Chrome项目。V8号称是目前世界上最快的JavaScript引擎，它的即时编译执行速度已经快到了接近本地代码的执行速度。Node.js不运行在浏览器中，所以也不存在JavaScript的浏览器兼容性问题。

Node.js内建了HTTP服务器支持，用户可以轻松地实现一个网站和服务器的组合。这个服务器不仅可以用来调试代码，而且它本身就可以部署到产品环境。Node.js还可以部署到非网络应用的环境下，比如一个命令行工具。Node.js还可以调用C/C++的代码，这样可以充分利用已有的诸多函数库，也可以将对性能要求非常高的部分用C/C++来实现。

* 1. **Node.js架构简介**

Node.js用异步式I/O和事件驱动代替多线程，带来了可观的性能提升。Node.js除了使用V8作为JavaScript引擎之外，还使用了高效的libev和libeio库支持事件驱动和异步式I/O。图1是Node.js架构的示意图。



图 1 Node.js架构图

Node.js的开发者在libev和libeio的基础上还抽象出了层libuv。对于POSIX操作系统，libuv通过封装libev和libeio来利用epoll或kqueue。而在windows下，libuv使用了windows的IOCP机制，以在不同平台下实现同样的高性能。

* 1. **Node.js核心模块简介**

Node.js的核心模块由一些精简而高效的库组成，为Node.js提供了基本的API。下面简要介绍几个最常用的核心模块。

（1）全局对象

全局对象及其所有属性全局变量都可以在程序的任何地方被访问。Node.js中的全局对象是global。常用的全局变量有process、console等。因此在代码的编写过程中要使用var定义变量以避免引入全局变量。

（2）Util

Util提供常用函数的集合，用来弥补核心JavaScript的功能过于精简的不足。常用的函数有util.inherits、util.inspect等。

（3）Events

Events是Node.js最重要的模块，因为Node.js本身架构就是事件式的，而它提供了唯一的接口。Events模块不仅用于用户代码与Node.js下层事件循环的交互，还几乎被所有的模块依赖。

（4）Fs

Fs模块是文件操作的封装，它提供了文件的读取、写入、更名、删除、遍历目录等文件系统操作。而且fs模块中所有的操作都提供了异步和同步两个版本。

* 1. **产品目标与范围**

（1）对Node.js做需求分析，编写需求说明书。

（2）对Node.js做测试需求分析，编写需求报告。

（3）对Node.js进行模块测试，编写测试报告。

（4）在Node.js的基础上做模块扩展。

* 1. **应交付成果**

项目应交付的产品如表1所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 交付时间 | 要求 | 备注 |
| No.1 | 项目计划书 | 2016/3/22 | 完整、明确 | 暂无 |
| No.2 | 项目进度表 | 2016/3/22 | 完整、详细 | 暂无 |
| No.3 | 项目工作日志 | 待定 | 暂无 | 暂无 |
| No.4 | 软件需求说明书 | 待定 | 暂无 | 暂无 |
| No.5 | 需求检查单 | 待定 | 暂无 | 暂无 |
| No.6 | 测试需求说明书 | 待定 | 暂无 | 暂无 |
| No.7 | 软件测试分析报告 | 待定 | 暂无 | 暂无 |
| No.8 | 软件测试评审报告 | 待定 | 暂无 | 暂无 |
| No.9 | 项目工作量统计报告 | 待定 | 暂无 | 暂无 |
| No.10 | 项目方法总结报告 | 待定 | 暂无 | 暂无 |
| No.11 | 项目个人总结 | 待定 | 暂无 | 暂无 |

表 1 应交付成果列表

* 1. **项目开发环境**
     1. 软件环境

Windows 10

* + 1. 硬件环境

4G内存 + 250G硬盘 +百兆网卡 + 百兆局域网

* + 1. 开发工具

webstorm

* 1. **项目验收方式和依据**
     1. 验收方式

验收方式为每周上课汇报、项目扩展代码以及实验验收文档。

* + 1. 验收依据

验收依据“应交付成果”清单进行验收。

1. **项目团队组织**
   1. **组织结构**

项目团队包括一名组长，三名组员。组长和组员协调合作。组长对项目进度、工作计划进行合理安排并分配相应的任务。组长和组员需及时完成任务。

* 1. **项目分工**

项目具体分工详见《node.js项目规划书v1.0版》。

* 1. **协作与沟通**

小组成员利用微信、邮件等通讯工具及时交流，并一周举行1-2次面谈会议。小组通过github进行文档的提交和更新。

所有沟通协作都要进行记录。

1. **实施计划**
   1. **工作流程**

工作流程参照《node.js项目规划书v1.0版》。

* 1. **总体进度计划**

总体进度计划参照《node.js项目规划书v1.0版》。

* 1. **项目控制计划**

暂无 。