

**<< Nginx测试与开发>>**

**软件开发计划书**



北京航空航天大学

2016-3-16

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1.0 | 2016-3-16 | 黄新越、杨云 | 余锋伟、李晓聪 | 初始版本，尚未完善 |
| 1.1 | 2016-3-23 | 余锋伟、黄新越 | 杨云、李晓聪 | 问题细化，完善时间表 |
| 2.0 | 2016-3-30 | 余锋伟 | 黄新越、杨云 | 添加计划分析 |
| 2.1 | 2016-4-1 | 余锋伟 | 黄新越、李晓聪 | 细化修改 |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[1引言 1](#_Toc447106952)

[1.1编写目的 1](#_Toc447106953)

[1.2 背景 1](#_Toc447106954)

[1.3特点 1](#_Toc447106955)

[1.4参考资料 2](#_Toc447106956)

[2 项目概述 2](#_Toc447106957)

[2.1工作内容 2](#_Toc447106958)

[2.2主要参加人员 3](#_Toc447106959)

[2.3产品 3](#_Toc447106960)

[2.3.1程序 3](#_Toc447106961)

[2.3.2文件 3](#_Toc447106962)

[2.4验收标准 3](#_Toc447106963)

[2.5完成项目的最迟期限 3](#_Toc447106964)

[3实施计划 4](#_Toc447106965)

[3.1计划分析 4](#_Toc447106966)

[3.1.1了解软件 4](#_Toc447106967)

[3.1.2 需求分析 4](#_Toc447106968)

[3.1.3测试分析 4](#_Toc447106969)

[3.1.4功能实现 4](#_Toc447106970)

[3.1.4 其他软件管理工具学习 4](#_Toc447106971)

[3.2 组织形式 5](#_Toc447106972)

[3.3任务分解与计划进度 5](#_Toc447106973)

[4支持条件 6](#_Toc447106974)

[4.1计算机系统支持 6](#_Toc447106975)

**软件开发计划书**

# 1引言

## 1.1编写目的

Nginx是一款功能强大且高性能的Web和反向代理服务器，其特点是占有内存少，并发能力强，稳定性好。Nginx可以在大多数Unix或类Unix系统上编译运行，如FreeBSD、Solaris、Linux等，并且官方还提供了Windows下的可执行版本。

Nginx目前已经成为市场份额仅次于Apache的Web服务器，被全球1.2亿个网站采用，而且在规模较大和对性能要求较高的网站、互联网服务中更为流行，几乎有成为标配的趋势。它也很好地适应了目前移动化、系统内外通过API集成的需求。目前，Nginx在Netflix、Wordpress.com、百度、新浪、网易、腾讯等国内外众多知名网站中应用。

编写此计划书，一是为了理解Nginx服务器工作过程，以及对Nginx需求进行探究，充分的了解其工作原理，并进行二次开发。二是为了管理项目进度，降低项目失败风险，并提供项目验收的依据，最终完成软件工程的八个实验。

## 1.2 背景

* 作者 ：Igor Sysoev
* 国籍： 俄罗斯
* 商业版本：Nginx, Inc.维护。商业版本与免费的社区版相比主要增加了高级负载均衡、应用健康检查、HTTP直播流和动态流、活动监控、远程日志等高级功能与技术支持（包括配置与调优、性能优化等）。
* 项目进展： 开发中，目前版本号1.9.12
* 联系方式：nginx@nginx.org

## 1.3特点

1. **热部署**

在master管理进程与worker工作进程的分离设计，使的Nginx具有热部署的功能，即在7×24小时不间断服务的前提下，升级Nginx的可执行文件。也可以在不停止服务的情况下修改配置文件，更换日志文件等功能。

1. **支持高并发连接**

在这一个互联网快速发展，互联网用户数量不断增加的今天，一些大公司、网站都需要面对高并发请求。理论上，Nginx支持的并发连接上限取决于你的内存。

1. **低的内存消耗**

在一般的情况下，10000个非活跃的HTTP Keep-Alive 连接在Nginx中仅消耗2.5M的内存，这也是Nginx支持高并发连接的基础。

1. **处理响应请求很快**

在正常的情况下，单次请求会得到更快的响应。在高峰期，Nginx可以比其他的Web服务器更快的响应请求。

1. **具有很高的可靠性**

Nginx是一个高可靠性的Web服务器，现在很多的网站都在使用Nginx，足以说明Nginx的可靠性。高可靠性来自其核心框架代码的优秀设计、模块设计的简单性；并且这些模块都非常的稳定。

## 1.4参考资料

《深入剖析Nginx》

《Nginx权威指南》

《Nginx中文文档》

# 2 项目概述

## 2.1工作内容

首先，了解Nginx的编译，安装与使用。

然后，利用源代码、开发文档等资料，分析Nginx的需求，细致划分，得到需求文档。通过阅读源代码，从整体上把握Nginx的设计思路，并将需求对应到源代码相应模块。

针对对Nginx提到的各项优异的性能，对其性能指标进行独立的测试，产出测试报告。

最后，根据小组成员对Nginx的理解，针对其配置繁杂（nginx.conf配置文件、监控命令等）等问题，开发图形化的配置、监控工具。产出可用的配置监控软件

## 2.2主要参加人员

黄新越：北航软件工程专业研一学生，主要研究方向为数据挖掘与软件需求。 李晓聪：北航软件工程专业研一学生，主要研究方向为需求工程。

杨云：北航软件工程专业研一学生，主要研究方向为数据挖掘。

余锋伟：北航软件工程专业研一学生，主要研究方向为编译优化。

## 2.3产品

### 2.3.1程序

Nginx代码。

图形化监控程序。

### 2.3.2文件

1. 项目计划书。

2. 需求文档。

3. 测试文档。

4. 项目总结报告。

## 2.4验收标准

Nginx代码、实验所需的相关文档和测试用例等等。

## 2.5完成项目的最迟期限

本学期第十七周。

# 3实施计划

## 3.1计划分析

### 3.1.1了解软件

在初步了解的基础上，尽可能的全面了解该软件，熟悉该软件的一些常用实例，分析软件的源代码。尽可能的从开发思想上去了解开发者的意图。

同时还要在网上收集一些使用者的反馈，了解软件的版本更迭过程。

### 3.1.2 需求分析

确定对系统的综合要求 。分析系统的数据要求，导出系统的逻辑模型，修正系统的开发计划。主要难点包括需求分析和建模。 时还需要了解一些评审指标和制作评审文档等。

### 3.1.3测试分析

对其进行测试，检查软件有没有错误，决定软件是否具有稳定性，写出相应的测试规范和测试用例。 和分析需求时一样，我们还要了解测试评审指标及制定评审文档。

### 3.1.4功能实现

实现Nginx配置的监控程序，需要学习和使用一种可视化的编程语言，我们计划使用开源软件QT作为可视化的手段。

### 3.1.5 其他软件管理工具学习

在研究过程中，我们会用到项目管理和项目配置工具，还有建模工具等，在整个的开发的过程中，我们都将伴随开发工具的学习，不再单独为研究工具而花费时间，可以通过其他阶段的时间剩余来合理安排。

## 3.2 组织形式

每周至少一次线下会议讨论项目进度，需要时可召开多次会议，线上会议不限。

QQ讨论组、微信讨论组用于即时的讨论与相关资源的分享。

## 3.3任务分解与计划进度

|  |  |  |  |
| --- | --- | --- | --- |
| 任务名称 | 工期 | 开始时间 | 完成时间 |
| **项目确定以及介绍** | **2 个工作日** | **2016年3月12日** | **2016年3月13日** |
| 项目调研 | 5 工时 | 2016年3月12日 | 2016年3月12日 |
| 选题会议 | 8 工时 | 2016年3月12日 | 2016年3月12日 |
| **项目计划书** | **3 个工作日** | **2016年3月14日** | **2016年3月16日** |
| 撰写引言 | 12 工时 | 2016年3月14日 | 2016年3月15日 |
| 项目概述 | 12 工时 | 2016年3月14日 | 2016年3月15日 |
| 目的及成果 | 12 工时 | 2016年3月14日 | 2016年3月15日 |
| mpp学习与使用 | 12 工时 | 2016年3月15日 | 2016年3月16日 |
| **软件需求分析** | **5 个工作日** | **2016年3月19日** | **2016年3月24日** |
| 软件模块分析 | 3 个工作日 | 2016年3月19日 | 2016年3月22日 |
| 测试需求分析 | 3 个工作日 | 2016年3月19日 | 2016年3月22日 |
| 软工规范分析 | 3 个工作日 | 2016年3月19日 | 2016年3月22日 |
| 扩展功能分析 | 3 个工作日 | 2016年3月19日 | 2016年3月22日 |
| 需求分析会议 | 6 工时 | 2016年3月23日 | 2016年3月23日 |
| **软件需求评审** | **2 个工作日** | **2016年3月25日** | **2016年3月26日** |
| 修改软件模块需求 | 2 个工作日 | 2016年3月25日 | 2016年3月26日 |
| 修改测试需求 | 2 个工作日 | 2016年3月25日 | 2016年3月26日 |
| 修改软工规范需求 | 2 个工作日 | 2016年3月25日 | 2016年3月26日 |
| 修改扩展功能需求 | 2 个工作日 | 2016年3月25日 | 2016年3月26日 |
| 需求评审会议 | 6 工时 | 2016年3月26日 | 2016年3月26日 |
| **软件需求复评审** | **2 个工作日** | **2016年3月27日** | **2016年3月28日** |
| 确定软件模块需求 | 2 个工作日 | 2016年3月27日 | 2016年3月28日 |
| 确定测试需求 | 2 个工作日 | 2016年3月27日 | 2016年3月28日 |
| 确定软工规范需求 | 2 个工作日 | 2016年3月27日 | 2016年3月28日 |
| 确定扩展功能需求 | 2 个工作日 | 2016年3月27日 | 2016年3月28日 |
| **软件产品改进与展示** | **10 个工作日** | **2016年3月29日** | **2016年4月10日** |
| 软件改进 | 9 个工作日 | 2016年3月29日 | 2016年4月8日 |
| 软件展示 | 6 工时 | 2016年4月10日 | 2016年4月10日 |
| **软件测试分析** | **6 个工作日** | **2016年4月11日** | **2016年4月16日** |
| 测试需求分析 | 6 个工作日 | 2016年4月11日 | 2016年4月16日 |
| **软件测试评审** | **2 个工作日** | **2016年4月17日** | **2016年4月18日** |
| 修改测试需求 | 2 个工作日 | 2016年4月17日 | 2016年4月18日 |
| **软件测试复评审** | **2 个工作日** | **2016年4月19日** | **2016年4月20日** |
| 确定测试需求 | 2 个工作日 | 2016年4月19日 | 2016年4月20日 |
| 软件演示与测评 | 4 个工作日 | 2016年4月21日 | 2016年4月26日 |
| 综合实验分析 | 4 个工作日 | 2016年4月27日 | 2016年4月30日 |
| 综合实验总结 | 4 个工作日 | 2016年5月4日 | 2016年5月8日 |
| **软件进度计划与控制** | **76 个工作日** | **2016年3月12日** | **2016年6月24日** |
| 每周组会 | 76 个工作日 | 2016年3月12日 | 2016年6月24日 |
| 每日更新github | 76 个工作日 | 2016年3月12日 | 2016年6月24日 |
| 工作量估计与统计分析 | 76 个工作日 | 2016年3月12日 | 2016年6月24日 |
| 配置管理 | 76 个工作日 | 2016年3月12日 | 2016年6月24日 |

# 4支持条件

## 4.1计算机系统支持

Linux/Windows/OS X，gcc