**Gitea的分析与应用**

**项目计划**

Version 1.0

小组成员：

SY1906915 麦梓健

SY1906505 孙维华

SY1906306 洪治凑

SY1906432 王子璇

SY1906117 王伟民

SY1906434 郑锋

# 1.项目概述

## 1.1 背景

Linux 内核开源项目有着为数众多的参与者。 绝大多数的 Linux 内核维护工作都花在了提交补丁和保存归档的繁琐事务上）。 到 2002 年，整个项目组开始启用一个专有的分布式版本控制系统 BitKeeper 来管理和维护代码。到了 2005 年，开发 BitKeeper与Linux 内核开源社区的合作关系结束，这就迫使 Linux 开源社区（特别是 Linux 的缔造者 Linus Torvalds）基于使用 BitKeeper 时的经验教训，开发出自己的版本系统。 他们对新的系统制订了若干目标： 速度、简单的设计、对非线性开发模式的强力支持（允许成千上万个并行开发的分支）、完全分布式、有能力高效管理类似 Linux 内核一样的超大规模项目（速度和数据量）。自诞生于 2005 年以来，Git 日臻成熟完善，在高度易用的同时，仍然保留着初期设定的目标。 它的速度飞快，极其适合管理大项目，有着令人难以置信的非线性分支管理系统。

GitHub、GitLab等基于Git的代码托管平台，尽管功能强大，但是过重、过于复杂了，对于许多项目而言并不需要如此庞大的体系，因而轻量级的框架gogs等相继出现。Gogs简单、轻便、运行快，只需要廉价的树莓派即可满足需求。但是gogs由于“独裁”性质的维护，使得由社区维护的gitea出现了。

## 1.2 简介

Gitea由go编写，由社区维护，安装简便，运行需求极低。同时gitea的功能非常强大，具有大量的可扩展组件：

* 支持活动时间线
* 支持SSH以及HTTP/HTTPS协议
* 支持用户、组织和仓库管理系统
* 支持仓库和组织级别Web钩子
* 支持仓库Git钩子和部署密钥
* 支持仓库工单、合并请求以及Wiki
* 支持迁移和镜像仓库以及它的Wiki
* 支持自定义源的Gravatar和Federated Avatar
* 支持MySQL、PostgreSQL、SQLite3、MSSQL和TiDB

## 1.3 术语

|  |  |
| --- | --- |
| Git钩子 | Git钩子是在Git仓库中特定事件发生时自动运行的脚本。可以定制一些钩子，这些钩子可以在特定的情况下被执行，分为Client端的钩子和Server端的钩子。 |
| Wiki | Wiki是一种在网络上开放且可供多人协同创作的超文本系统，由沃德·坎宁安于1995年首先开发，这种超文本系统支持面向社群的协作式写作，同时也包括一组支持这种写作 |
| 镜像仓库 | 用来保存制作好的容器的一个仓库，只不过这里的容器被做成了镜像，所以我们称其为镜像仓库。 |
| SSH | SSH 为建立在应用层基础上的安全协议。SSH 是较可靠，专为远程登录会话和其他网络服务提供安全性的协议 |
|  |  |

# 2 项目内容

## 2.1 项目框架

## 2.2 工作内容

1. 通过阅读代码和文档分析gitea功能和模块结构，
2. 对gitea进行需求分析
3. 对gitea实现进行建模
4. 分析gitea实现的不足和缺陷
5. 对发现的不足与缺陷进行测试修改

# 3 项目计划

## 3.1 人员分工

项目初期：

初期全员进行环境搭建和Git以及Go语言的学习。项目组成员分别在本地搭建Gitea开发环境，熟悉Gitea的功能特性。

王子璇负责收集Go语言安装资料，协助组员完成Go环境的安装；并为组员提供Go语言入门资料。

郑锋负责Git学习资料的收集，并对组员进行Git工作流的培训。

麦梓健负责制定项目工作计划、制作PPT，并进行会议记录。

项目中期：

项目组成员学习Gitea项目的代码结构、实现原理。由洪治凑组织项目组成员进行学习资料和笔记的交流分享。

项目组成员分析Gitea实现的不足和缺陷，讨论优化的方案。麦梓健组织例会进行方案评审

项目后期：

确定Gitea的优化的方案后，麦梓健组织例会进行具体分工，项目组成员通过编写代码进行方案实现。

项目组成员使用Git工作流协作开发，孙维华负责代码评审。王伟民负责系统测试。

## 3.2 工作方向

* 自定义通知特性：
  + 仓库状态发生改变时（比如新的PR、issue），发送自定义通知给项目组成员。
  + 通知类型可以是微信、邮件
* 添加GitLab迁移支持：
  + 从Gitlab进行的自托管或托管迁移添加了新的迁移接口。
  + 在UI上为自托管git服务添加选择
* 支持多语言的README：
  + 代码仓库可能需要针对不同语言的准备多个自述文件。
  + 在github上，一般仓库会在默认README上加入其他语言的链接。
  + 我们打算在gitea中加入原生支持的多语言README的特性

## 3.3 组织形式

1. 会议：每周1次，进行工作总结汇报和检查
2. 群聊：随时进行，进行工作情况的交流和问题的互助。
3. 使用git管理项目进度
   1. 以 master 作为团队主分支，其上所有commit是团队工作合并后的稳定版本
   2. hotfix 分支用于问题反馈的修复，向 master 和 develop 合并
   3. 组长在develop分支上开始新的实验内容
   4. 各个团队成员通过从develop上创建新的个人分支进行自己的工作
   5. 团队成员需要通过pull request将自己的分支合并到develop分支，团队成员审核后组长操作
   6. hotfix 分支合并完成后应被删除
   7. 团队生成的最终文档通过release发布

## 3.4 进度安排

# 参考资料

1. <https://docs.gitea.io/en-us/>
2. https://discourse.gitea.io/