**操作系统教学实践云平台**

**操作系统教学实践云平台方案设计**

**北京航空航天大学**

**操作系统教学实践云平台方案设计**

**文档变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **时间** | **版本** | **变更内容简述** | **变更人** | **审核人** |
| 1 | 2022/2/15 | V0.1 | 文档初稿撰写 | 王廉杰 | 王雷 |
| 2 | 2022/9/5 | V0.1 | 完善设计方案 | 王廉杰 | 王雷 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1. 引言 4](#_Toc114425227)

[1.1. 文档概述 4](#_Toc114425228)

[1.2. 文档适用范围 4](#_Toc114425229)

[2. 总体设计 4](#_Toc114425230)

[2.1. 操作系统实践教学云平台概述 4](#_Toc114425231)

[2.2. 操作系统实践教学云平台架构设计 5](#_Toc114425232)

[3. 操作系统实践教学云平台子系统模块设计 5](#_Toc114425233)

[3.1. 操作系统实践教学云平台功能模块设计 5](#_Toc114425234)

[4. 操作系统实践教学云平台子系统功能逻辑 6](#_Toc114425235)

[4.1. 学生实验环境 6](#_Toc114425236)

[4.1.1. 学生实验环境镜像设计 6](#_Toc114425237)

[4.1.2. 学生实验环境用户管理 7](#_Toc114425238)

[4.1.3. 学生实验环境Web IDE实现 8](#_Toc114425239)

[4.2. 代码托管平台 9](#_Toc114425240)

[4.2.1. 学生实验仓库管理 9](#_Toc114425241)

[4.2.2. 实验分支开发模型 9](#_Toc114425242)

[4.3. 实验评测机制 10](#_Toc114425243)

[4.3.1. 自动评测脚本部署 11](#_Toc114425244)

[4.3.2. 评测服务容器实现 11](#_Toc114425245)

[4.3.3. 信息展示平台实现 12](#_Toc114425246)

# 引言

本项目应用容器化技术，在arm64架构的华为TaiShan 200服务器之上设计了一个更加通用、便捷的操作系统实践教学平台，针对现有操作系统实验集成环境缺少图形化界面的困扰，以及教学团队部署实验内容与评测过程中的不便等问题做出了改进，能够支持学生通过Web IDE或SSH连接到私有的Linux实验容器，进行操作系统实验的开发学习和调试运行，为学生提供了一个低成本且便捷的实验环境，同时提高了教学团队的教学效率，为师生提供了良好的操作系统实验教学体验。

## 文档概述

本文档描述了操作系统实践教学平台的概要设计，为具体实现提供指导。

## 文档适用范围

本文档适用范围待定。

# 总体设计

## 操作系统实践教学云平台概述

本项目的目标是基于Docker容器技术、GitLab代码托管服务以及Web IDE开发环境，在arm64架构的openEuler环境下设计实现一个独立且统一的操作系统实践教学平台，支持在同一物理服务器上为大量学生提供独立隔离的Linux集成实验环境，并允许学生使用浏览器接入Web IDE，使学生能够将精力集中于实验内容中，同时允许课程团队基于Git远程仓库分发实验代码和相关文件，并为各个实验部署评测脚本，对学生的代码提交进行实时评测和反馈，同时能够通过网页端查看学生的提交历史和评测成绩。

为了达到上述目标，操作系统实践教学平台需要实现以下功能：

1. 提供容器化的学生实验环境

* 实现不依赖于本地的云端实践环境，支持操作系统实验的完整流程；
* 在arm64平台下提供不同架构的交叉编译、调试和仿真环境；
* 实现各学生环境间的隔离和独立性，保证学生之间的代码和文件互不可见；
* 为学生提供图形化界面的开发环境，优化学生体验；
* 支持进行代码的提交和评测反馈。

1. 提供代码托管平台

* 提供基于Git的学生代码托管服务，允许学生提交和管理所有实验分支的代码；
* 支持学生通过网页实时浏览实验代码和评测结果；
* 便于教学团队查看学生仓库的提交情况。

1. 提供代码评测服务

* 支持通过代码托管平台的服务端Hook自动触发各实验分支的评测脚本；
* 在学生提交代码时实时反馈评测信息；
* 将评测结果保存在学生仓库和服务后台，便于学生查看评测历史；
* 允许教学团队查看所有学生的评测结果和实验得分。

## 操作系统实践教学云平台架构设计

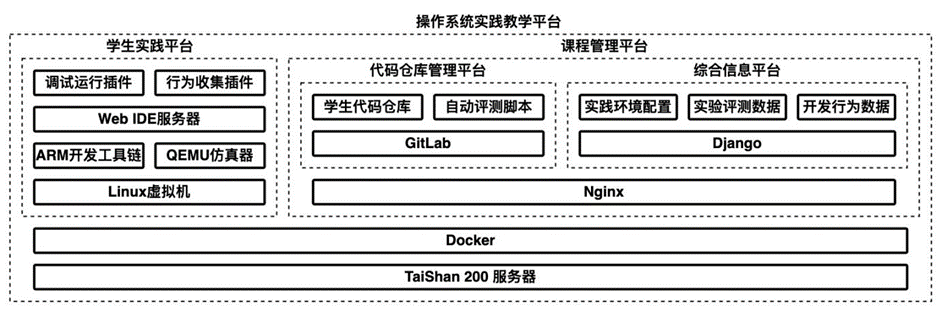


图 2.1 操作系统实践教学云平台整体架构

为满足学生和教学团队的不同需求，本文将操作系统实践教学平台分为学生实践平台和课程管理平台。平台在华为TaiShan 200以及华为云提供的弹性云服务器上测试部署，支持arm64架构的Linux平台，并通过Docker容器技术部署不同服务，整体架构如图 2.1 所示。

# 操作系统实践教学云平台子系统模块设计

## 操作系统实践教学云平台功能模块设计

本平台的功能模块可以依照架构设计划分为三个部分：

1. 学生实践平台面向学生提供实验代码的云端开发环境，并基于统一的Docker镜像进行部署。该镜像在openEuler 22.03发行版镜像的基础上，统一提供了arm64平台下的交叉开发工具链、QEMU仿真器、Web IDE服务器code-server、OpenSSH服务端、Git版本控制工具以及其他必要的基础包，并支持浏览器端Web IDE以及SSH终端两种接入方式，学生接入环境后可直接在个人的代码仓库中开发并提交实验代码。
2. 代码仓库管理平台使用容器化的GitLab实现，通过为学生与教学团队配置不同的权限，可实现实验仓库的访问控制与统一管理，并可通过Git Hook部署自动评测脚本，实现学生代码的自动评测。通过网页端，学生可查看自己的代码仓库、提交历史和评测结果，也可供教学团队查看所有学生的实验完成情况。
3. 综合信息平台基于Django和Vue框架开发，为教学团队提供了前端查询页面，可实时查看所有学生在实验中的评测结果，便于统计学生的实验完成情况和在各实验中的评分。

# 操作系统实践教学云平台子系统功能逻辑

## 学生实验环境

基于能够快速部署独立且统一的学生实验开发环境的目标，平台使用Docker容器技术进行学生实验环境的配置与部署。为了满足操作系统实验教学活动中学生的开发编译、调试仿真和代码提交等需求，实验环境基于openEuler 22.03发行版镜像进行配置，在基础发行版基础上，统一安装了开发所必要的软件包和工具链。

### 学生实验环境镜像设计

通过在openEuler镜像上预装统一的软件包，平台能够为所有学生提供统一且隔离的操作系统实验开发环境，提升学生的开发效率和体验，并增进学生对现代操作系统的理解和应用能力。在openEuler发行版源的基础上，为了满足操作系统实验的跨架构内核开发需求，平台提供了在arm64架构下构建的支持多种架构的gdb调试器和QEMU仿真器，并为MIPS架构构建了完整的gcc交叉编译工具链，包含链接器ld、汇编器as和反汇编器objdump等内核开发中使用的工具。

针对用户态的操作系统实验，镜像内还部署了GCC、Rustup、OpenJDK、Python、Golang和PHP等多种语言的编译工具链，便于学生在用户态下进行多语言的编译构建和调试运行。通过在学生容器内启动code-server和sshd服务进程，可为学生提供浏览器端Web IDE和SSH终端两种远程接入方式，从而直接在云端进行操作系统实验代码的开发。

平台提供的镜像能够通过Docker容器技术为大量学生批量部署，并支持操作系统实验开发的完整流程。针对不同体系结构的特定实验课程，教学团队还可方便地在该镜像的基础上进行迭代和扩展，配置不同的开发软件包和工具链，并为学生创建新的实验容器，以快速支持特殊的操作系统实验。

### 学生实验环境用户管理

为了分配并管理各学生对应的实验环境容器、GitLab用户和仓库，平台需要按以下步骤维护新增的学生信息，流程如图4.1：

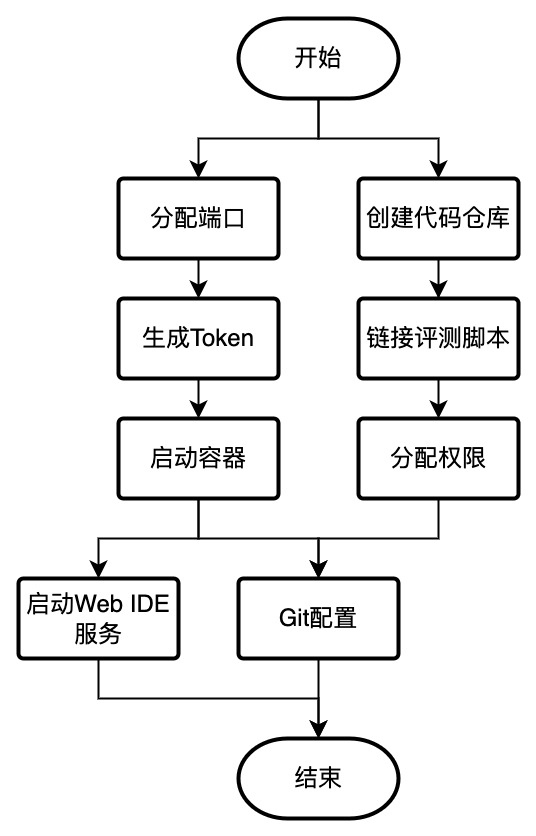


图 4. 1 学生实验环境自动部署流程图

1. 为学生分配一个SSH端口和一个Web IDE端口，并生成随机口令，保证每个学生分配的端口不重复
2. 根据统一的实验环境镜像，为学生创建实验容器
   1. 配置重启策略为always，以便容器关闭或出错时自动重置
   2. 容器中的SSH端口22以及Web IDE端口需要映射到分配的外部端口
   3. 容器以 stu\_<学号> 命名，便于主机进行识别和管理
   4. 需要将外部的公共工具链目录以只读方式挂载到容器中
3. 容器创建后，自动设置容器用户和Web IDE的登录密码，为学生随机生成的SSH密钥对，并启动SSH服务端sshd和Web IDE服务端
4. 在GitLab中创建学生对应的用户和实验仓库，并为用户添加容器的SSH公钥
5. 在学生仓库中创建main分支，并为学生对应的用户配置开发权限
6. 将Git服务器生成的known\_hosts文件拷贝到容器中，在容器中使用Git克隆学生仓库，并自动使用git config命令设置学生的身份信息
7. 更新服务端数据目录，记录学生的学号、SSH端口、Web IDE端口和登录口令

### 学生实验环境Web IDE实现

平台提供的Web IDE基于VS Code集成开发环境，使用code-server作为服务端实现，并提供了容器化的自动部署和基于密钥的身份认证。因此我们通过code-server能快速实现一个基于VS Code的Web IDE。学生实验环境完成创建后，学生只需通过浏览器访问分配的对应端口，在认证页面提供口令后即可开始使用VS Code进行实验代码的开发。

基于VS Code具有的能力，Web IDE能够完善支持操作系统实验代码的编辑和调试，其提供了多语言的语法高亮、语法检查、代码补全和Git版本控制等功能，能够直接集成到平台提供的代码托管服务，其界面如图4.2所示。

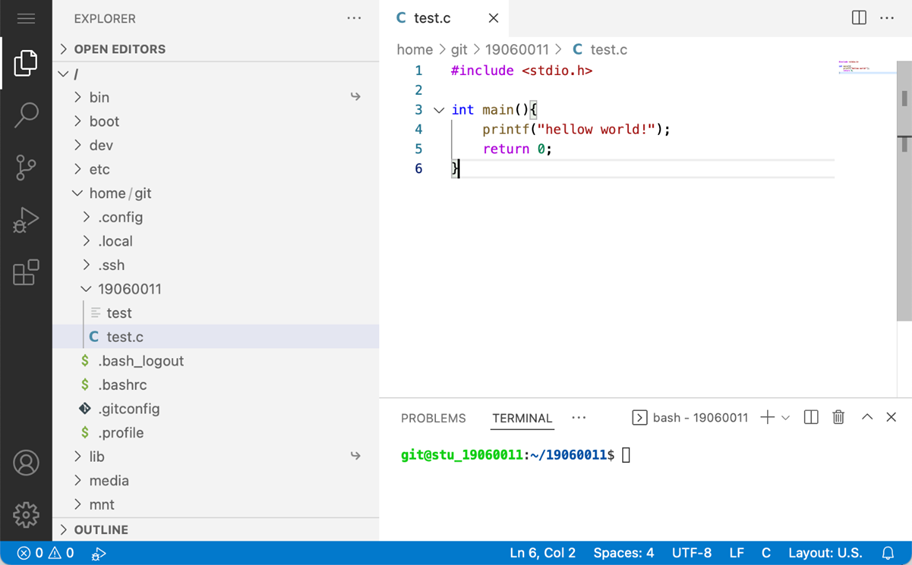


图 4. 2 Web IDE 代码编辑

## 代码托管平台

### 学生实验仓库管理

Git是目前广泛使用的版本控制系统，能够高效地处理代码文件的版本管理。平台提供了基于GitLab的代码托管服务，以提供学生实验代码的可视化管理和自动评测服务。在为学生创建实验环境的同时，平台自动为学生在GitLab上创建学生用户，以及对应的实验仓库，在统一的项目组中对所有学生的实验仓库进行管理，并自动为学生对应的GitLab用户分配权限。如图4.3所示，每位学生仅具有个人实验仓库中的开发权限，其他同学的代码对其而言是隐藏的，而教学团队可通过管理员用户，对所有仓库进行管理，能够看到每个学生代码仓库的情况，作为学生实验情况的反映。

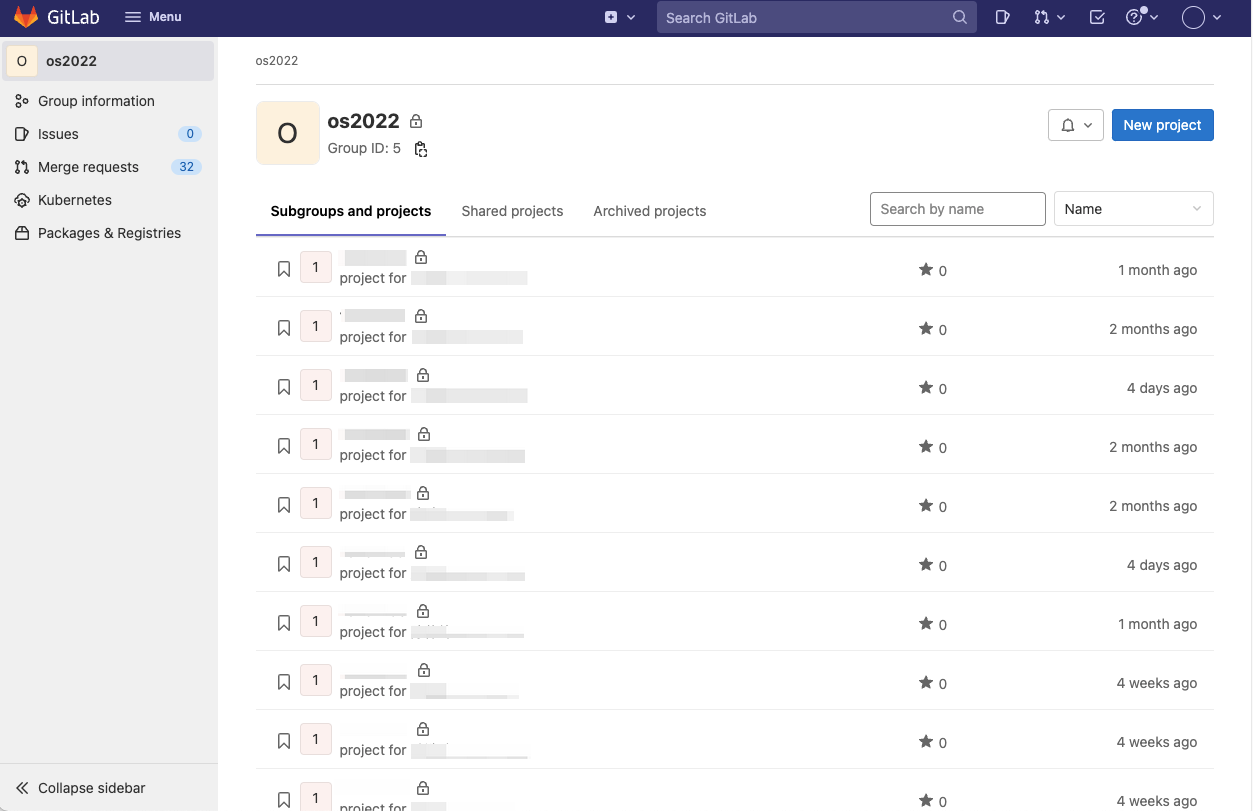


图 4. 3 GitLab项目组

### 实验分支开发模型

基于Git的版本控制模型，学生可以在自己的实验仓库中的不同分支下独立进行不同的操作系统实验，如图4.4，并在提交的同时触发不同分支下的评测脚本，实现评测结果的实时反馈。通过拉取公共的远端仓库，学生能够方便地获取实验团队下发的实验初始代码和文档，并自动建立本地实验分支开始开发。

学生进行实验代码开发的时序图如图4.5所示，学生可通过Web IDE提供的版本控制功能，或直接使用Git命令与Git服务器进行交互。针对每个实验，学生首先从公共仓库拉取实验初始代码和文档，并在实验仓库中自动创建分支，在实验环境中完成代码开发和版本管理。在开发完成后，使用Git命令即可将代码推送到远程服务器，此时服务器会自动触发对应实验分支的自动评测脚本，同时为学生给出完整的评测反馈，并将评测结果保存在服务器中。

图形用户界面, 应用程序

描述已自动生成

图 4. 4 学生仓库中的不同实验分支

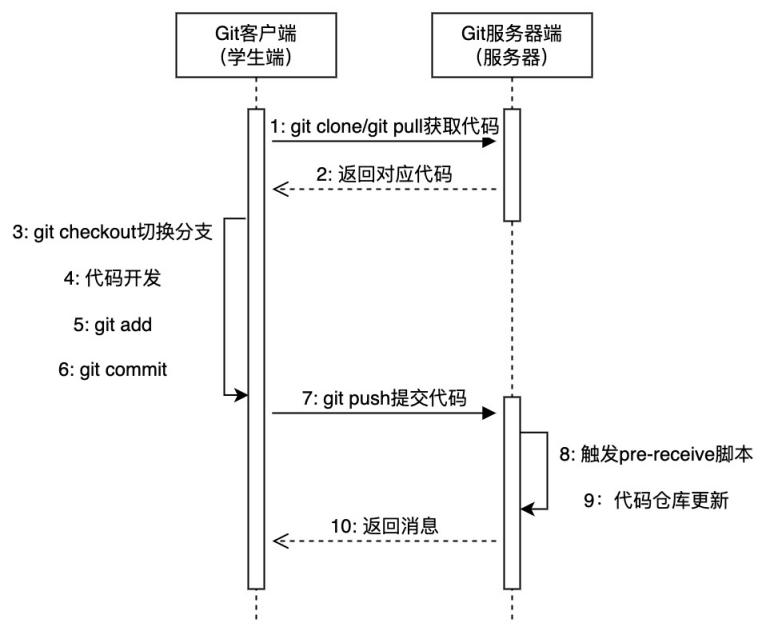


图 4. 5 学生实验代码开发时序图

## 实验评测机制

### 自动评测脚本部署

借助Git服务器端的Hook机制，平台能够在学生将对应分支的实验代码提交到仓库后，自动触发评测程序，从而实现了实验代码的自动评测与反馈。创建学生仓库时，平台将正确命名的可执行文件链接到Git仓库的.git/hooks目录下，即可挂载对应的钩子程序，在处理相应事件时被Git服务器所调用，从而为项目配置各种类型的强制性策略。

在评测环境部署的过程中，平台需要将Git服务端中学生实验仓库下的post-receive钩子程序配置为评测入口，从而在学生提交实验代码时自动分发并执行对应实验分支下的评测脚本。评测入口程序使用SSH协议，与评测服务容器进行通信，并将评测服务的输出实时反馈到用户端Git的远程输出中。评测服务首先获取学生本次提交的实验分支，对于开启评测的实验分支，若已为该实验部署可执行的评测脚本，则将本次提交的代码从远程仓库克隆至评测容器中的临时目录，再以该目录路径为参数执行评测目录下对应实验的评测脚本，将其返回值作为评测的得分。实验评测结束后，评测服务将本次评测的用户、时间和得分信息存储在服务器的数据卷中，并将评测结果上传到学生仓库中的评测结果分支，学生可通过拉取对应分支或从GitLab网页端查看历史的评测结果。

评测脚本默认存储在服务主机的/opt/ostest目录，其中每个目录都对应一个开放评测的实验分支名称，教学团队只需在对应目录下放置命名为judge.sh的可执行脚本，即可为对应实验配置评测逻辑。

### 评测服务容器实现

为了在与学生环境相同的独立环境中进行实验评测，并与Git服务端实时进行评测Hook触发和实验仓库的克隆两种双向通信，平台使用了独立的评测服务容器osjudge，在Git服务端的post-receive钩子程序中使用SSH协议与评测服务通信。在评测服务启动时，平台需要自动读取Git服务端的SSH公钥，将其存储在评测服务和学生环境容器的known\_hosts历史公钥文件中，并为评测服务创建SSH密钥对，将其公钥添加到GitLab的特权评测用户中，以允许评测程序读取学生仓库。同时，为了允许Git服务器向评测服务发起SSH连接，还需要将评测私钥添加到Git服务器中，在发起连接时用于认证。为了允许评测服务调用交叉编译工具链和公共Docker镜像，需要将工具链目录以及Docker服务监听的Unix域套接字挂载到评测容器中，为评测脚本提供容器级别的虚拟化评测能力，以避免直接运行学生提交的不可信代码。

### 信息展示平台实现

为了允许教学团队查看所有学生的代码提交和评测历史，了解各实验的得分和完成情况，平台提供了基于Django和Vue框架的信息查询前端，以对评测服务记录的数据进行读取和解析，允许管理员在系统中为教学团队配置教师用户，并对教师用户进行认证，允许经授权的教师用户实时查看所有学生在实验中的提交评测记录和各实验的得分，并对学生的成绩排名等统计数据进行可视化的呈现。