**操作系统教学实践云平台**

**操作系统教学实践云平台指导文档**

**北京航空航天大学**

**操作系统教学实践云平台指导文档**

**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 | 2022/8/7 | V0.1 | 文档初稿撰写 | 陈伟明 | 王雷 |
| 2 | 2022/9/7 | V0.2 | 补充用户测试和多种语言测试 | 陈伟明 | 王雷 |
| 3 | 2022/9/10 | V0.3 | 补充教学端使用指南 | 王廉杰 | 王雷 |
| 4 | 2022/9/15 | V0.4 | 增加学习行为 | 王廉杰 | 王雷 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1. 学生端使用指南 4](#_Toc114425120)

[1.1. 学生登录实验环境 4](#_Toc114425121)

[1.2. 操作系统代码获取和测试样例的添加 6](#_Toc114425122)

[1.3. 操作系统运行及调试流程 9](#_Toc114425123)

[1.4. 操作系统内核实验提交评测 11](#_Toc114425124)

[1.5. 用户态实验流程 11](#_Toc114425125)

[1.6. 常见语言支持情况 13](#_Toc114425126)

[2. 教学端使用指南 19](#_Toc114425127)

[2.1 实验文档的发布 19](#_Toc114425128)

[2.2 学生成绩查询 23](#_Toc114425129)

# 学生端使用指南

## 学生登录实验环境

操作系统课程实验平台部署在服务器上，学生主要通过SSH连接和WEB IDE两种方式远程访问服务器。首先，课程组会根据学生的学号生成对应的实验账号信息，如图1所示，包括了SSH连接的端口，WEB IDE的访问端口以及学生用户的口令。

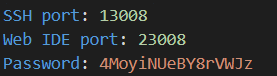


图1 分配给学生使用的端口信息

学生可以通过SSH连接方式接入到远程开发机器上。如图 2所示，学生可以使用如PowerShell等终端进行SSH连接：

1. 输入”ssh [git@122.9.36.214](mailto:git@122.9.36.214) -p SSH\_port”，其中SSH\_port即为上述分配给学生使用的SSH的连接端口，122.9.36.214为平台的域名或IP地址（此处为华为云上测试环境的IP），按照提示输入口令即可成功连接到远程开发机器上。
2. 登录到 SSH 后，可以使用“cd Student\_ID”命令切换到学生仓库中，其中Student\_ID为学生的学号。学生使用各种系统工具以及进行实验代码的编写都需要通过命令的方式进行交互。

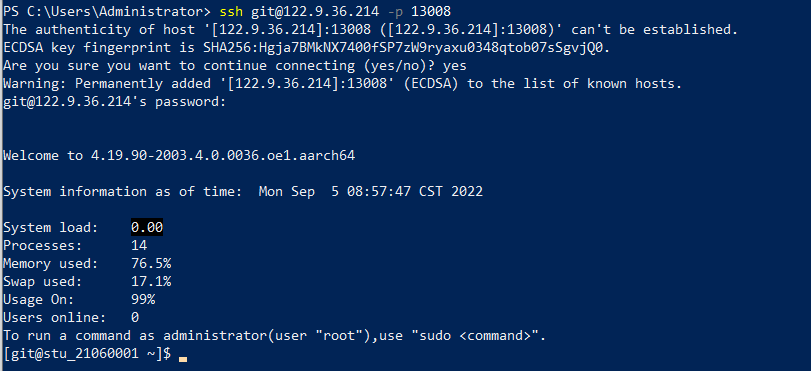


图2 SSH方式登录示意

此外，学生可以使用提供了图形化界面以及集成开发环境的Web IDE进行操作系统实验的开发。如图3所示，学生可以使用如Chrome等浏览器：

1. 输入地址”http://122.9.36.214:Web\_IDE\_port”即可跳转到Web IDE的页面进行开发，其中Web\_IDE\_port为上述分配个学生使用的Web IDE连接端口，
2. 按照提示输入口令进行相关的身份认证即可成功登录到Web IDE的开发平台上。
3. 后续的代码编写等流程，均可参考VS Code的使用方法。

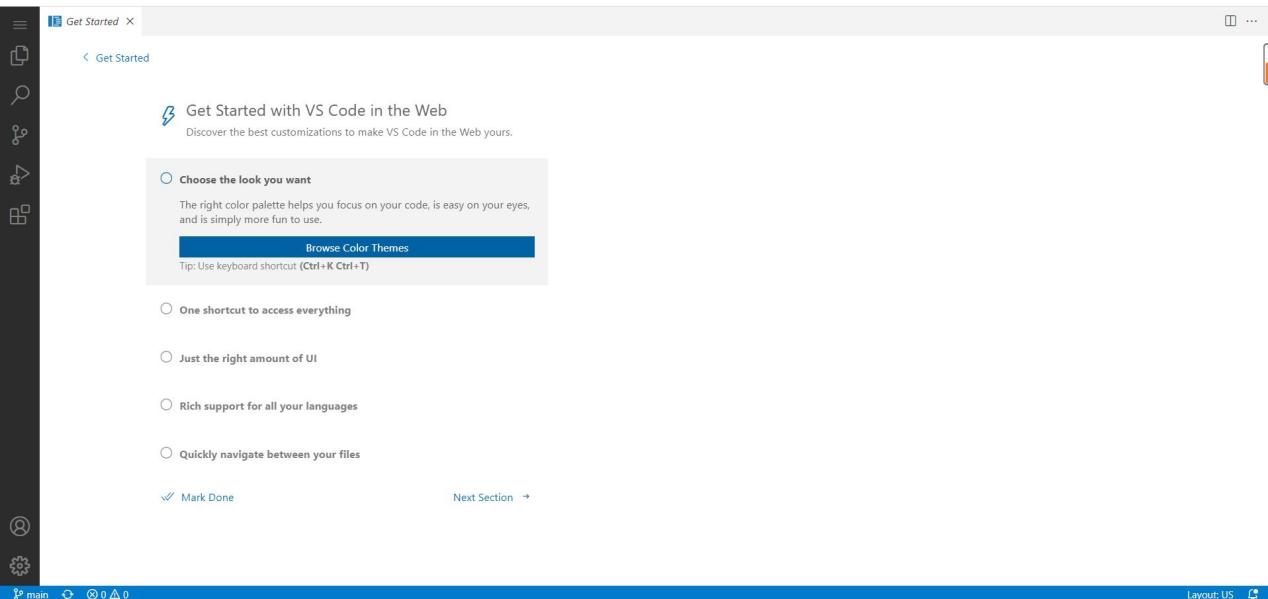


图3 Web IDE登录页面

## 操作系统代码获取和测试样例的添加

学生可按照实验课程要求完成操作系统实验的代码编写，完成编写之后学生可以在本地进行自主的样例测试。学生通过SSH方式登录之后，可以直接通过命令行进行交互，而使用Web IDE的学生则需要在编写完代码之后，在IDE中开启终端，具体步骤为”Application Menu ->Terminal ->New Terminal”，在终端中进行命令行交互，完成工程的构建。

学生在进入到个人仓库之后，代码获取流程如下：

1）在终端中执行”git fetch public”从公共仓库来获得各个实验分支下发的初始代码和文档：

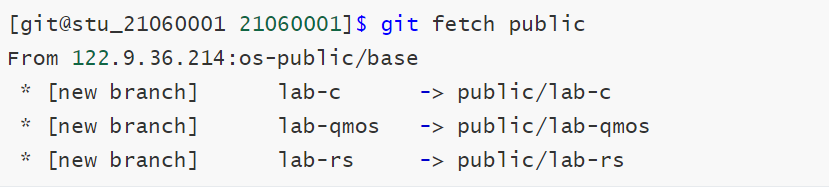


图4 学生实验仓库代码拉取

2）在获取代码之后，学生使用”git checkout branch\_name”指令来切换到所需分支下完成操作系统实验，其中branch\_name为对应分支名，如lab-qmos即为对应的操作系统实验分支。

3）如图5所示，学生获得代码之后便可以通过在线阅读文档进行相关实验代码的撰写和测试。此外，学生也可以登录gitlab仓库：http://122.9.36.214/，进入到对应的实验分支下，打开对应的文档进行阅读，如图6所示，其中登录所需的用户名为学生的学号，如21060001，密码在图1已给出。

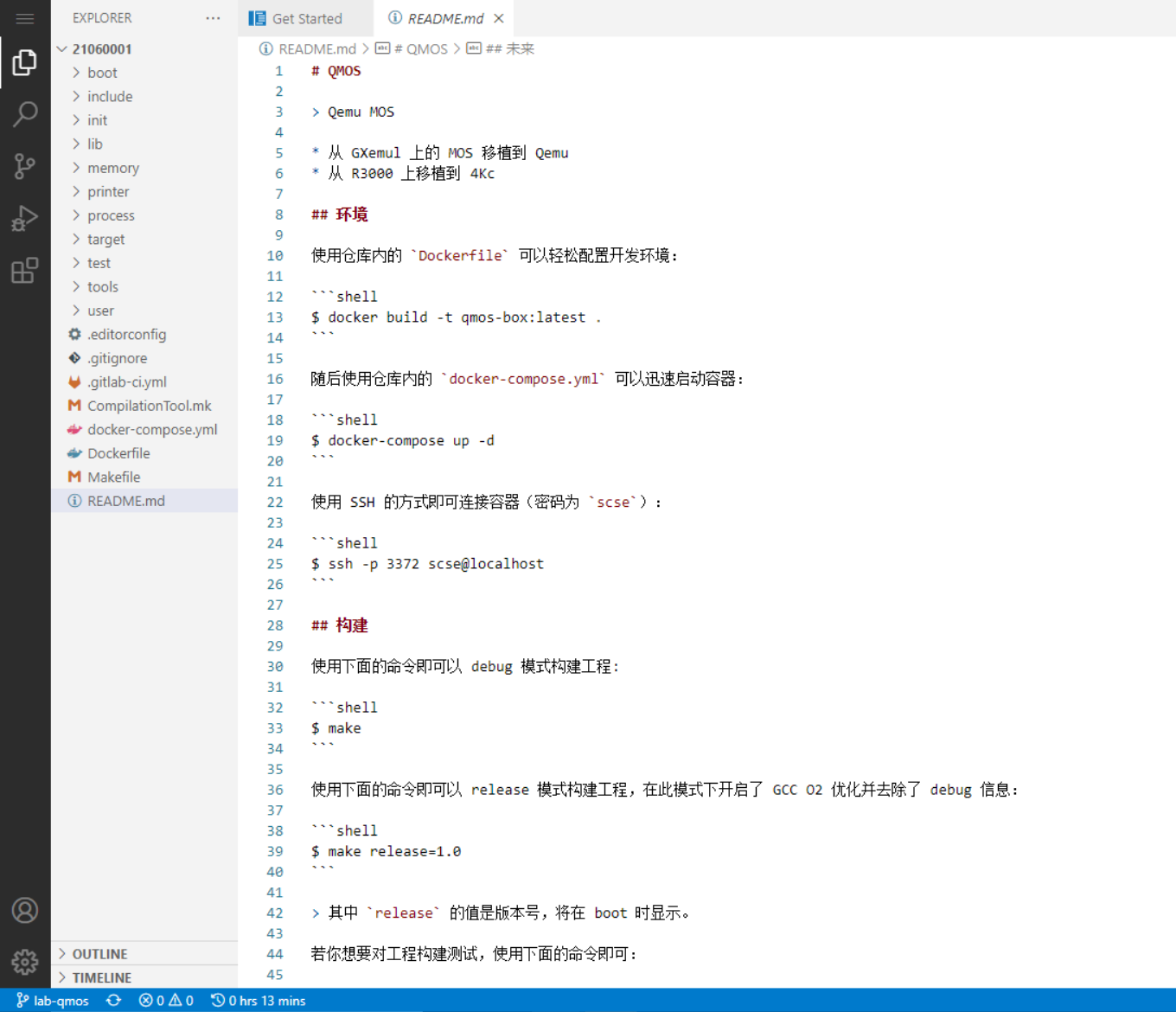


图5 实验任务文档在Web IDE上在线阅读

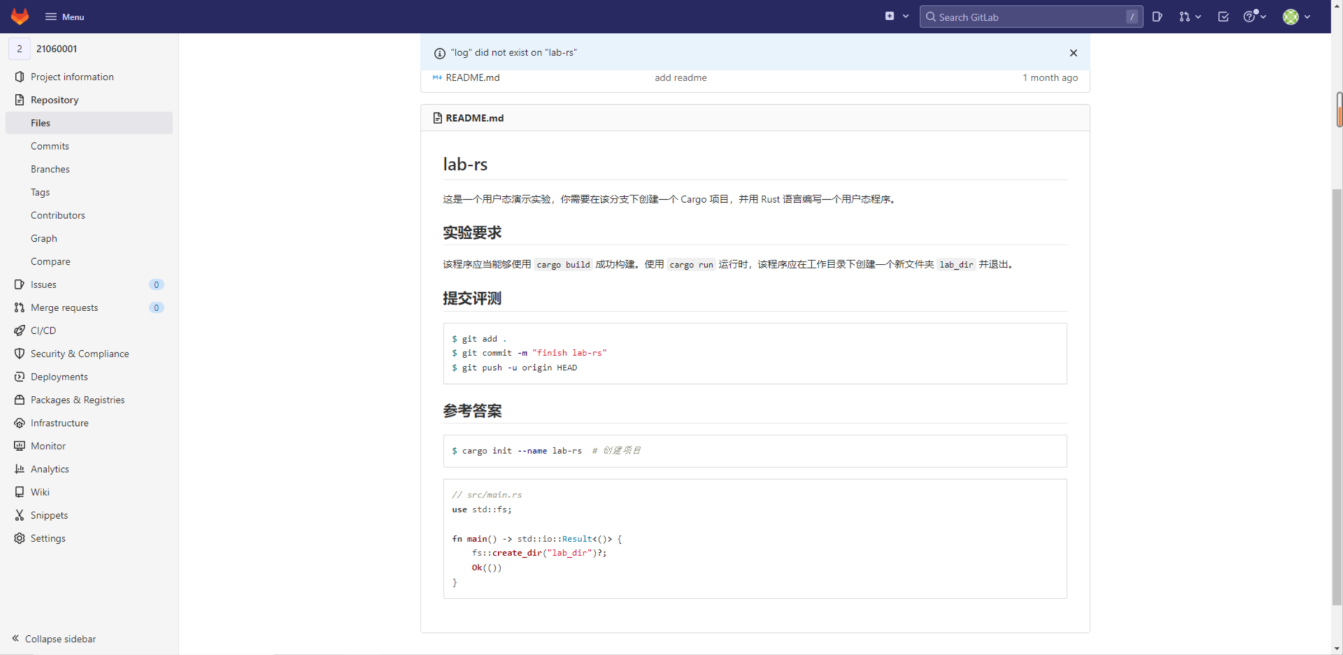


图6 Gitlab上阅读实验文档

对于内核态程序测试，流程如下：

1. 学生在init/init.c中添加相关的测试函数或引入相关函数进行调用；
2. 在终端主目录下使用”make”命令即可构建对应的工程。

对于用户态程序测试，流程有所不同：

1. 首先需要在test目录下添加相关的测试函数，模仿已有的用户态测试程序，以libMain为主函数名，书写用户态测试程序，假设文件名称为Demo.c。其中，需要确保libMain的原型为”void libMain(void);”。
2. 在test/Makefile中进行一定修改：
3. 在test\_list中添加测试程序索引，其中索引必须为数字，例如为上述Demo测试程序添加索引1952；
4. 在对应的test\_索引号系列中添加新增的测试程序声明，定义为学生所编写的用户态程序组，例如：”test\_1952 := Demo.x”。整体修改思路可以参考图4



图7 test/Makefile修改示意

1. 在init/init.c中添加相关的宏定义：
2. 在”#if TEST == \*”、”#elif TEST == \*”系列判定之后增加对应索引的判断，如”elif TEST == 1952”；

2、在新增的预处理作用域内使用PROCESS\_CREATE\_PRIORITY函数创建初始进程，其中程序名为 TestDemo，优先级由学生自行决定。整体修改思路如图5所示

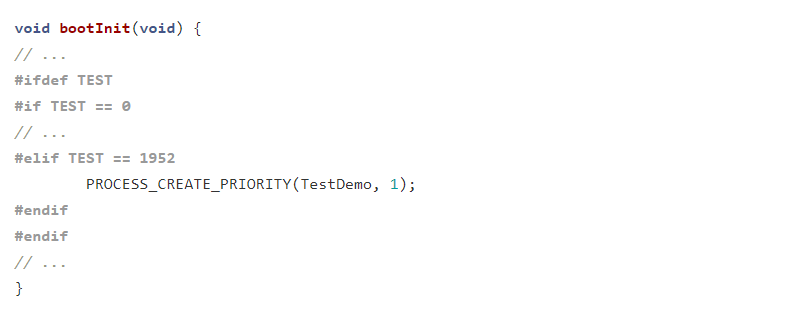


图8 init.c修改示意

4）在终端使用”make test=n”命令即可对测试程序构建相应的工程，其中n为所编写测试程序对应的索引。

## 操作系统运行及调试流程

操作系统运行流程如下：

1. 在主目录下make或make test=<test-case-index> 构建操作系统内核
2. 在终端中使用”make run”命令来运行QMOS内核。

如图6所示，成功启动一个不含测试测序的内核。为应对调试的需求，目前可使用”make debug”来运行QMOS内核，此命令将调用Qemu的int和mmu调试参数（每当中断和异常发生时，显示CP0相关寄存器的值；每当填写TLB时，显示相应TLB表项被填写的内容）。学生可灵活使用C文件中的DEBUG宏，帮助其查看内核运行过程中的重要变更信息。

系统在运行时，内核会陷入一段循环中，学生欲退出运行：

1. 同时按Ctrl+a键，松开之后再按c键即可调出QEMU的monitor
2. 输入q/quit即可退出。

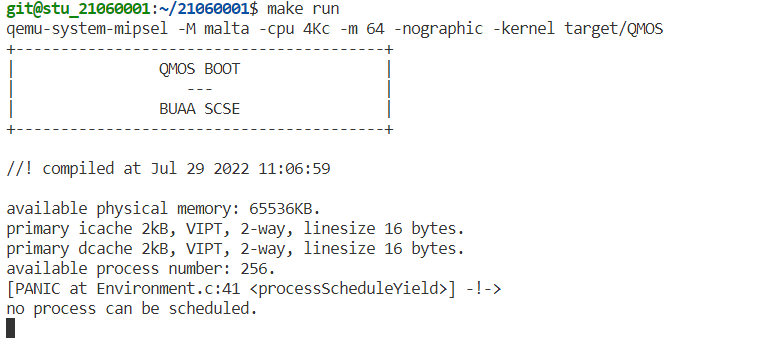


图9 内核运行

同时，本平台支持GDB调试：

1. 使用make或make test=<test-case-index> 构建操作系统内核
2. 在终端输入”make socket”命令，即可让内核进入debug模式
3. 新建终端，输入”gdb --eval-command "target remote :1234" ./target/QMOS”即可启动GDB进行调试，如果需要调试时动态加载的用户程序（例如test/Demo.c），应当在进入GDB后通过”add-symbol-file ./test/Demo.b”命令加载测试程序的符号表，方可调试用户程序。如图7所示，成功运行GDB对内核进行调试。

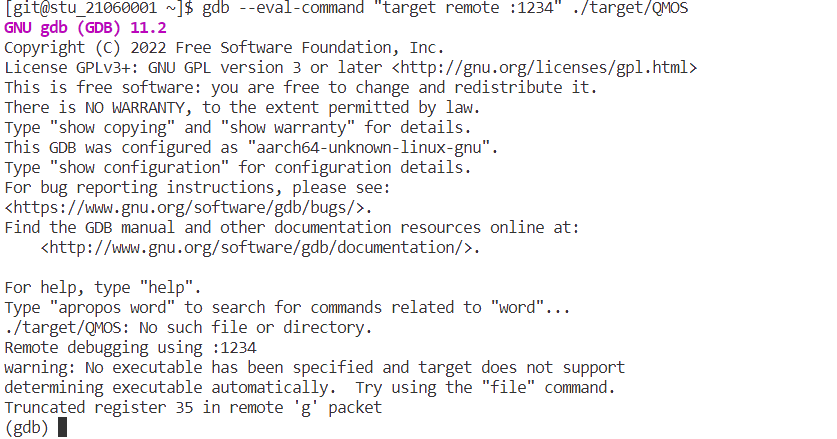


图10 GDB运行

## 操作系统内核实验提交评测

学生在本地完成代码的编写以及测试之后，可以将代码提交到git仓库，同时触发评测机制。在学生主目录下，使用以下命令进行提交:

$ git add .

$ git commit -m "commit\_info"

$ git push -u origin HEAD

其中commit\_info为学生自行编写的提交信息，善用提交信息可以方便学生对实验错误的溯源。

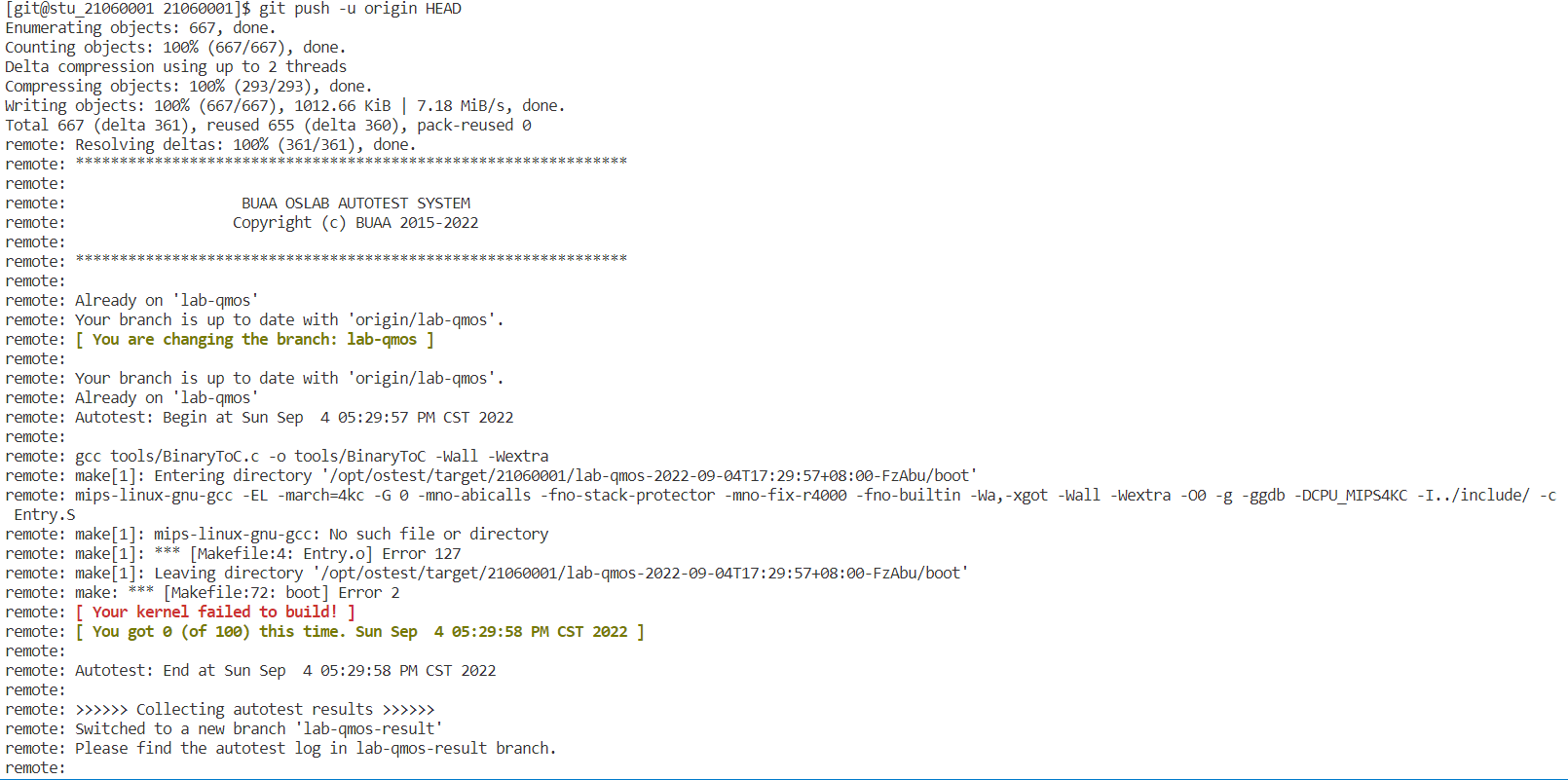


图11 提交评测示意

如图9所示，为对某实验的提交评测结果，系统会返回相应的评测信息和对应的分数情况，上图红色部分提醒学生创建内核失败，用户可以根据该信息进行查询。

## 用户态实验流程

学生在获取代码时，不仅可以获取到对应的操作系统实验代码分支“lab-qmos”，还可以切换到其他用户态实验分支，如“lab-c”和“lab-rs”。以Rust 语言用户态实验为例，讲述用户态实验的流程：

1. 在Web IDE 终端上，输入“git checkout lab-rs”
2. 通过阅读文档，获得对应实验的任务：“学生需要在该分支下创建一个 Cargo 项目，并用 Rust 语言编写一个用户态程序”以及要求：“该程序应当能够使用‘cargo build’成功构建。使用’cargo run’运行时，该程序应在工作目录下创建一个新文件夹‘lab\_dir’并退出”。
3. 本地测试根据语言特性，进行不同的测试方式。
4. 提交评测流程同操作系统实验：

$ git add .

$ git commit -m "commit\_info"

$ git push -u origin HEAD

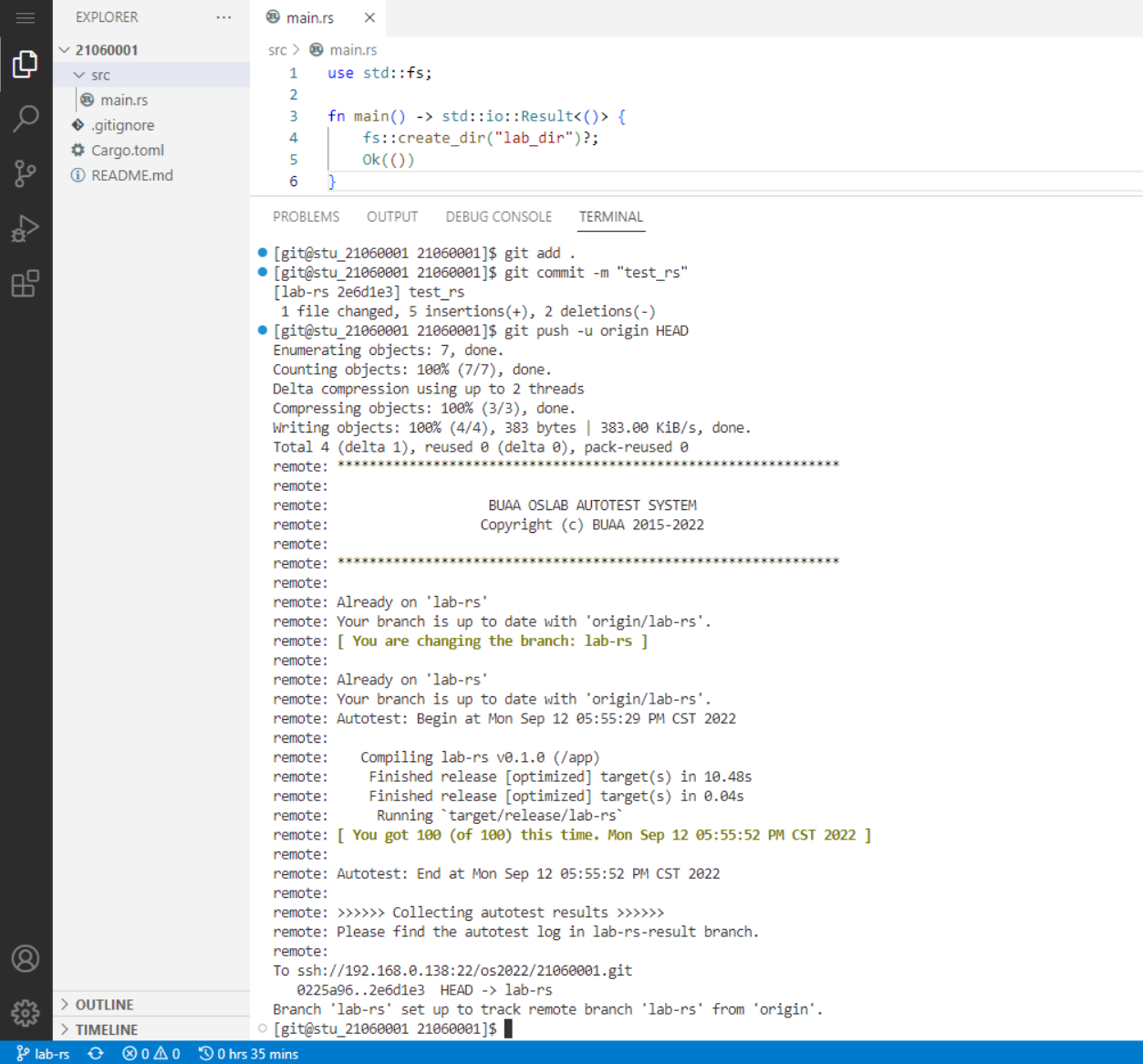
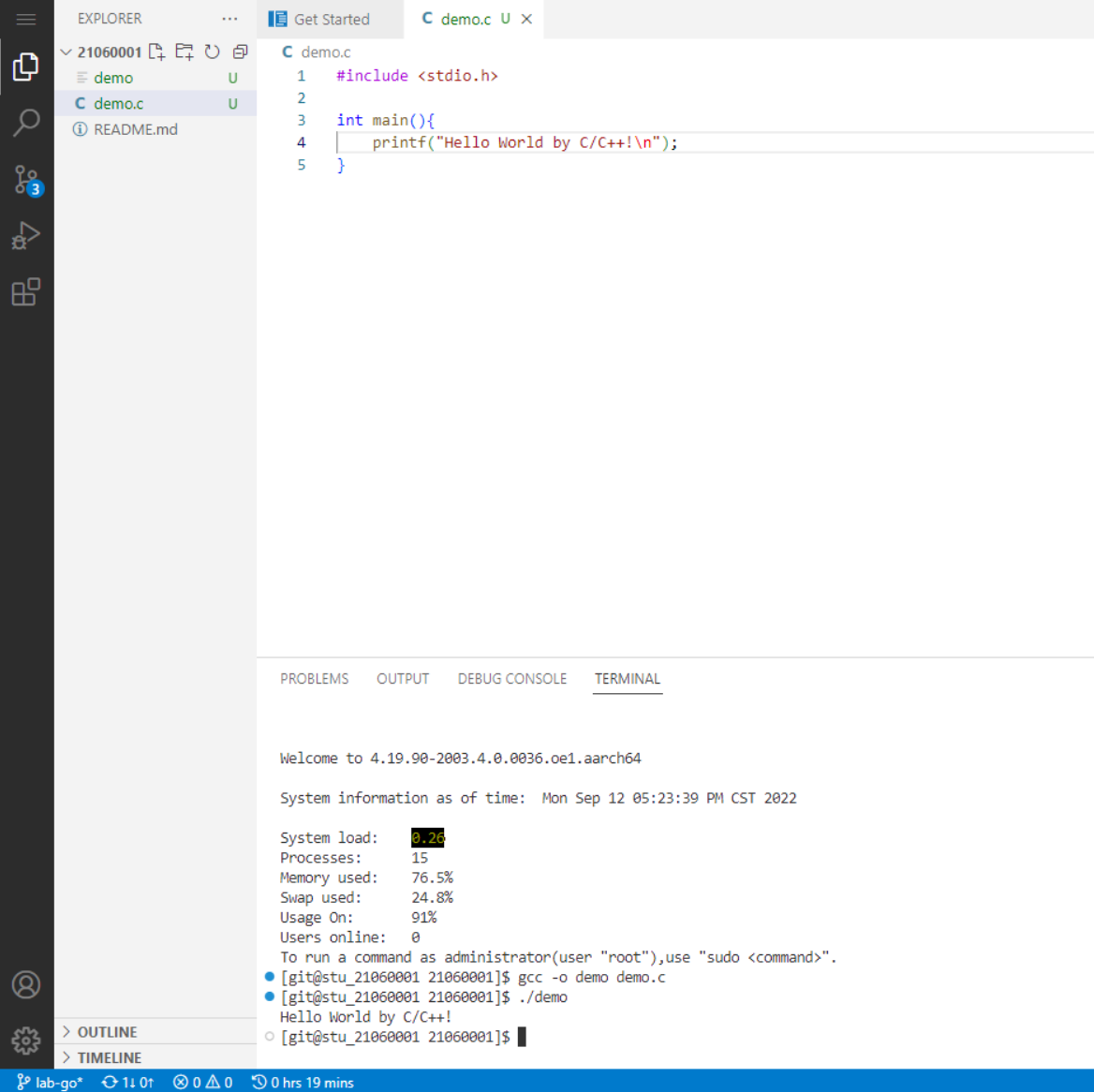


图 12 Rust 语言用户态实验

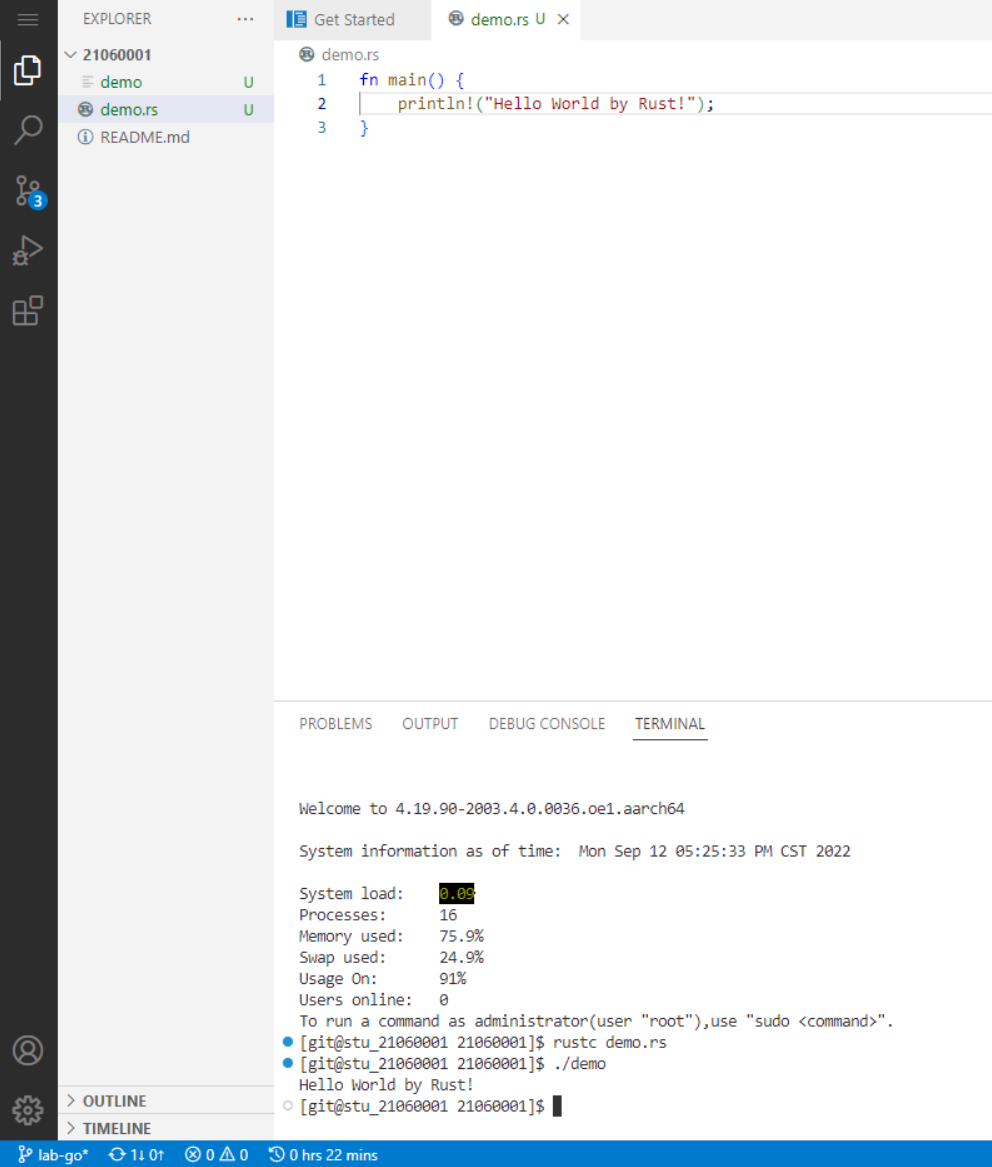
## 常见语言支持情况

本WEB IDE支持常见的编程语言，如C/C++、rust、go、python、shell、java、php语言。学生可以编写相关的程序，并通过终端来进行相应的程序运行。本指南以输出文本”Hello world by Language\_name”为功能构造demo程序，对常见语言支持情况进行测试。

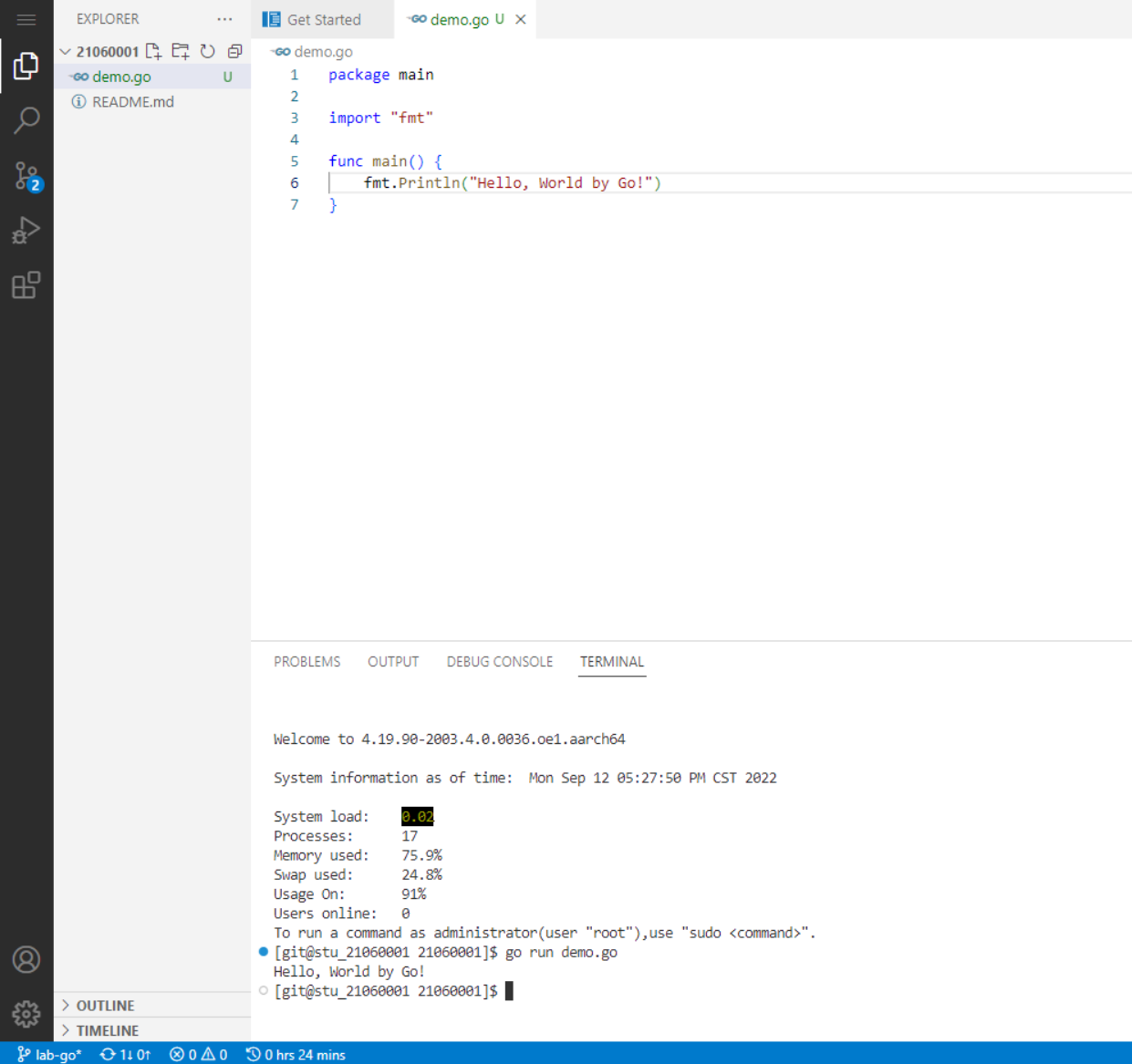
C/C++:



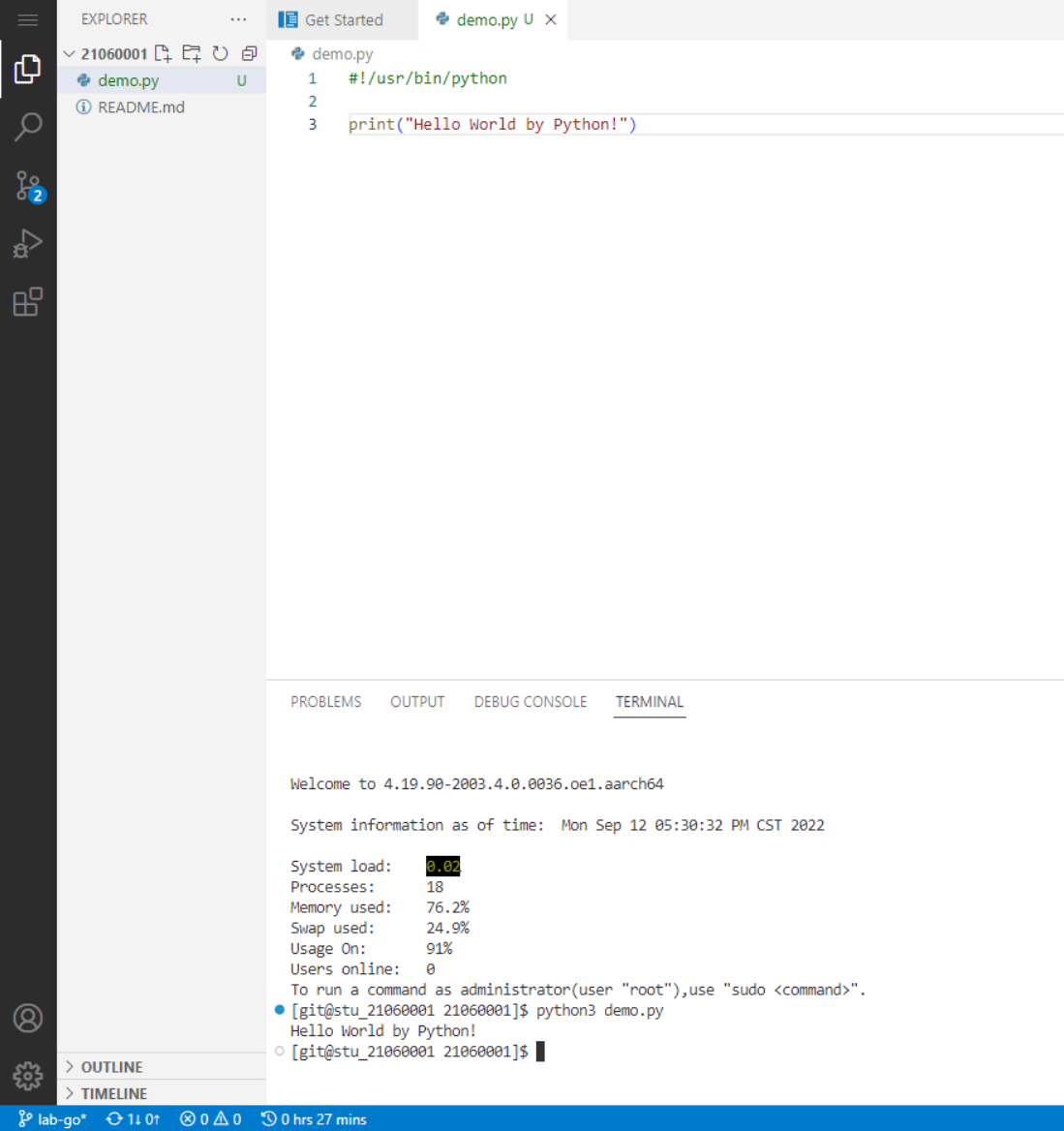
Rust:



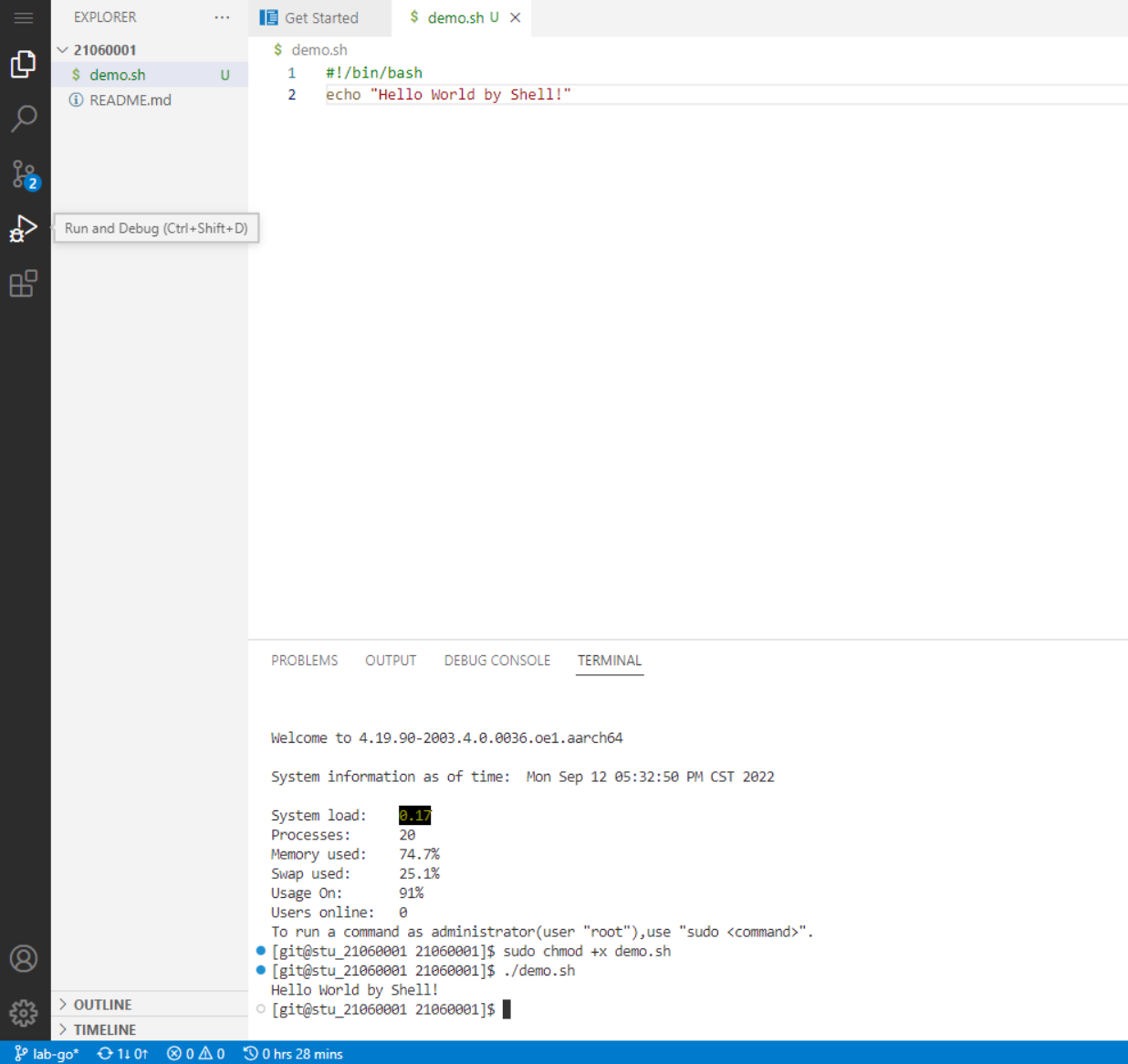
Go:



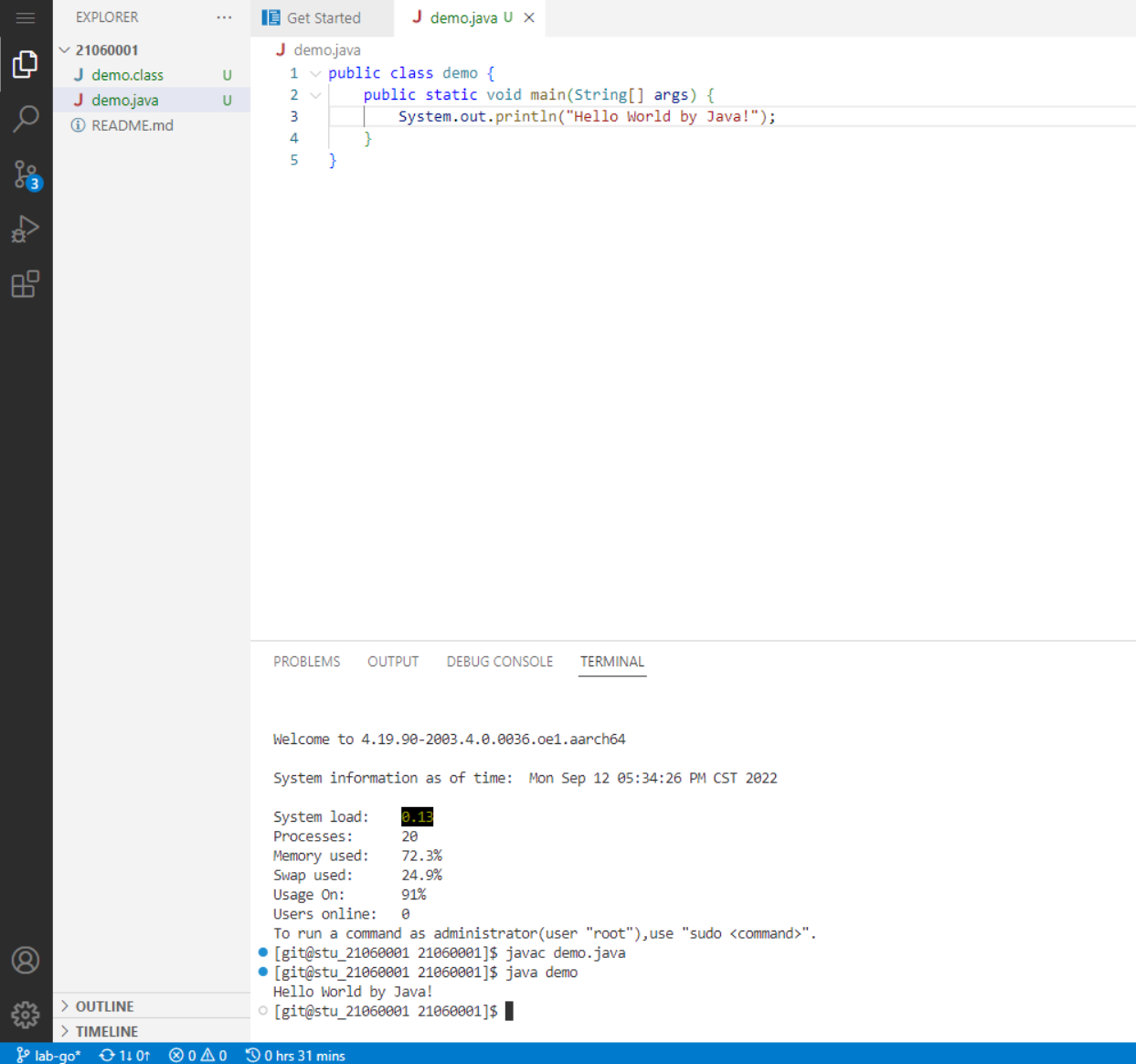
Python:



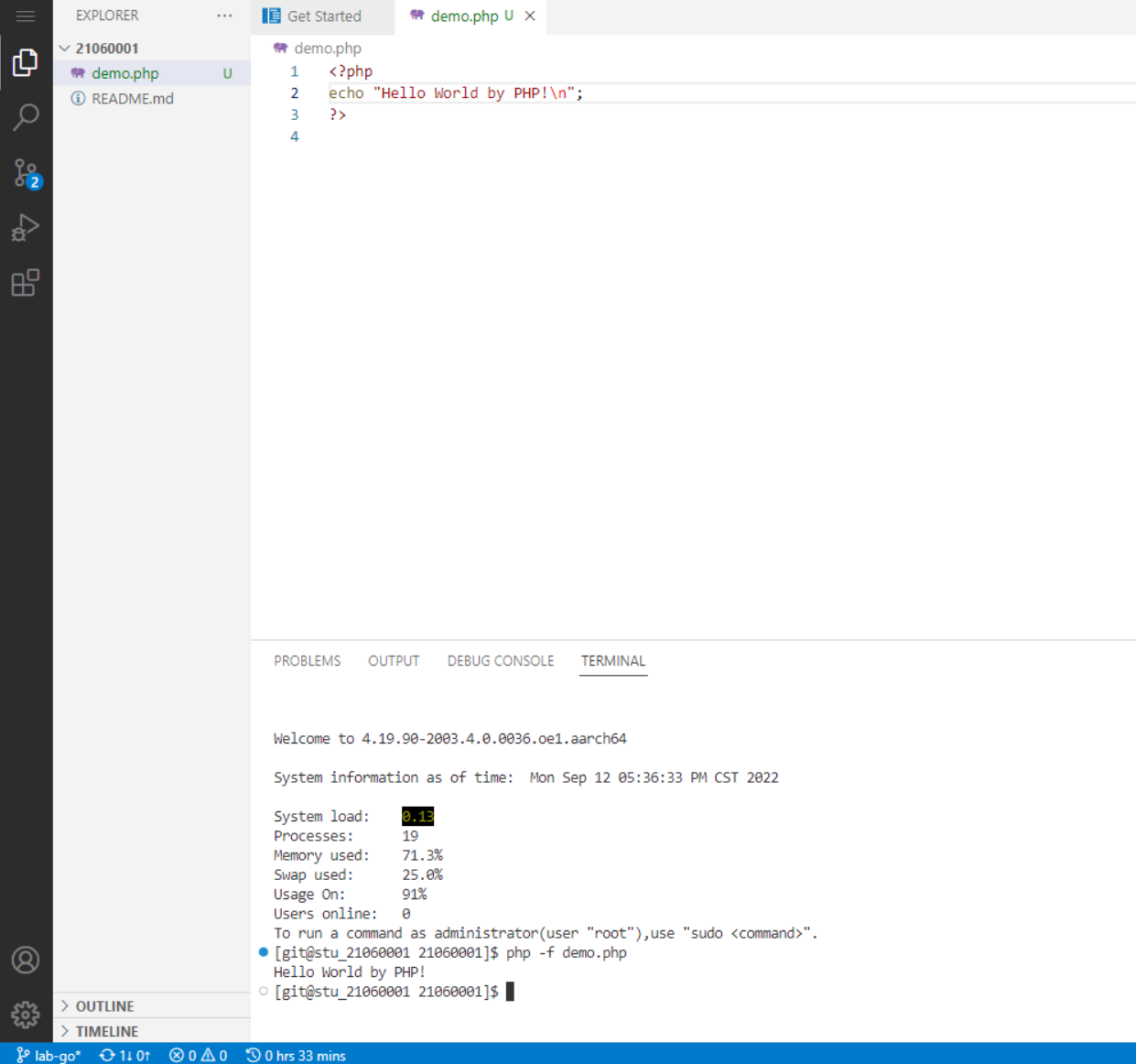
Shell:



Java:



PHP:



# 教学端使用指南

## 实验文档的发布

教学团队可以通过两种方式下发实验文档到学生端上：

一是可以通过Gtilab，在Web上直接进行文件的上传2操作：

1. 登录Gitlab仓库：http://122.9.36.214，使用教学团队的账号进行登录，在此提供相应教学团队的管理用户名以及密码：用户名：`root`；密码：`Kn8iQUuqYFjP0BIxVUe+s01s4L8qfylw808FlHZOLt8=`。
2. 选择对应的项目：os-public/base，该项目即为学生可以获取的源仓库。如图13所示，选择对应的实验分支，并点击“+”号，选择“上传文件”。

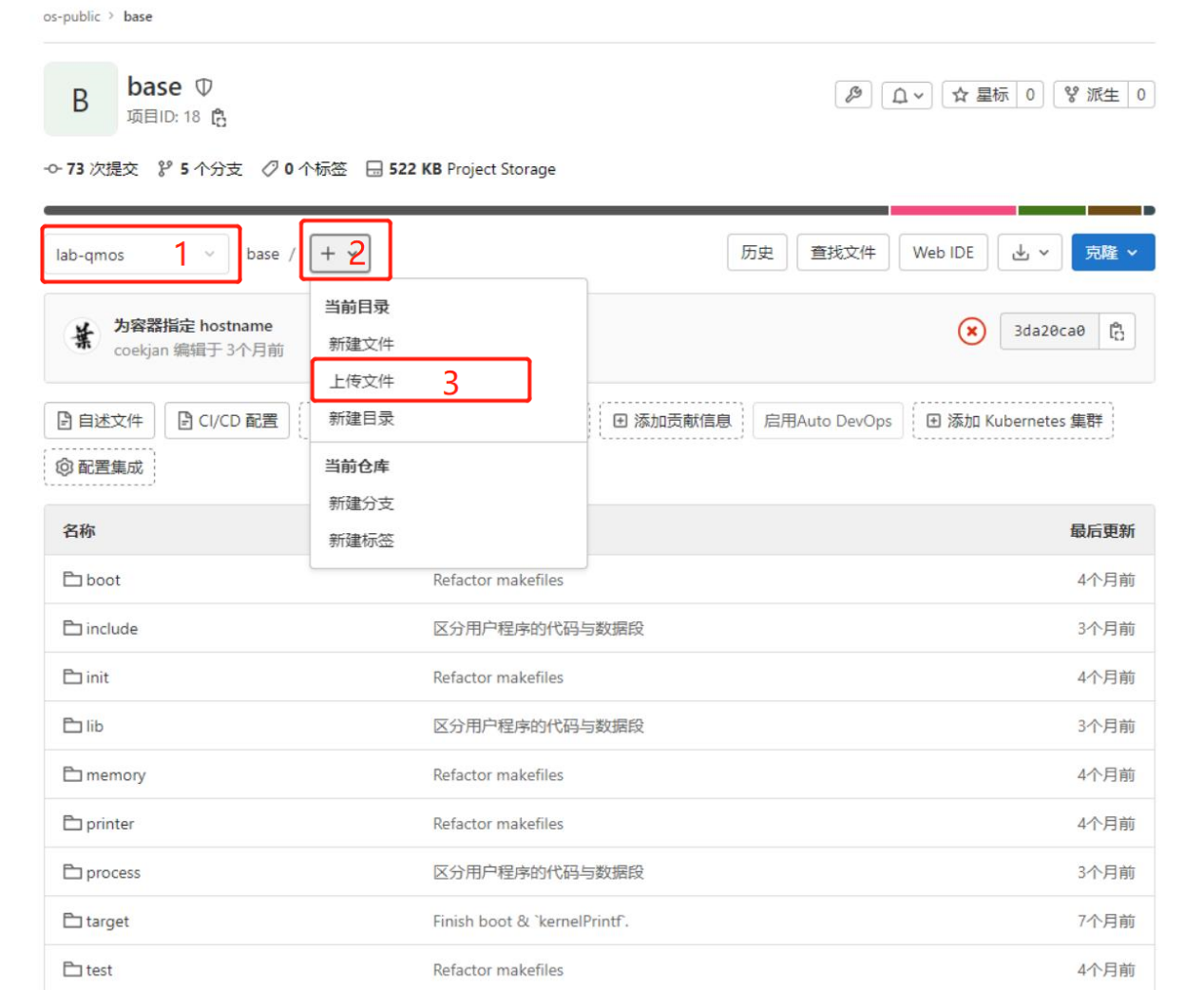


图13 在Gitlab仓库下上传文档

1. 如图14所示，选择需要上传的文档，编辑好上传时的提交信息，确认目标分支准确无误。

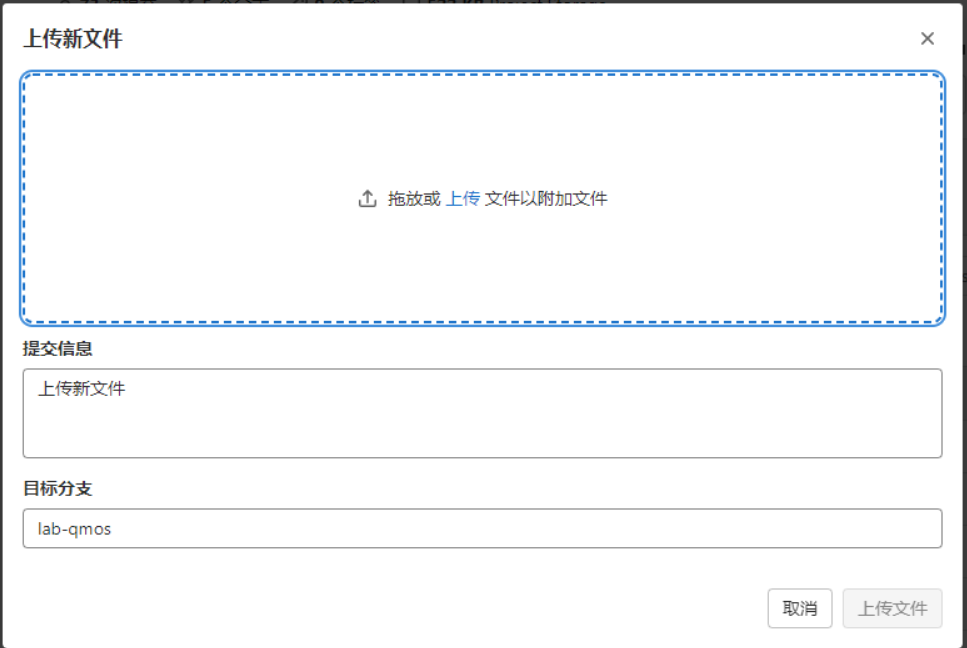


图14 Gitlab最终上传文档

二是教学团队可以通过与远程仓库创建连接，在本地上进行更新：

1. 可以使用如PowerShell、Ubuntu终端等终端，使用”git clone git@122.9.36.214:os-public/base.git”来与远程仓库建立联系。如果是第一次提交还需要额外进行密钥添加的操作：
   1. 如图 15所示，先在终端中输入“ssh-keygen”，产生所有动作都可以按回车跳过。再输入“cat ~/.ssh/id\_rsa.pub”输出自己的密钥，将密钥进行复制。

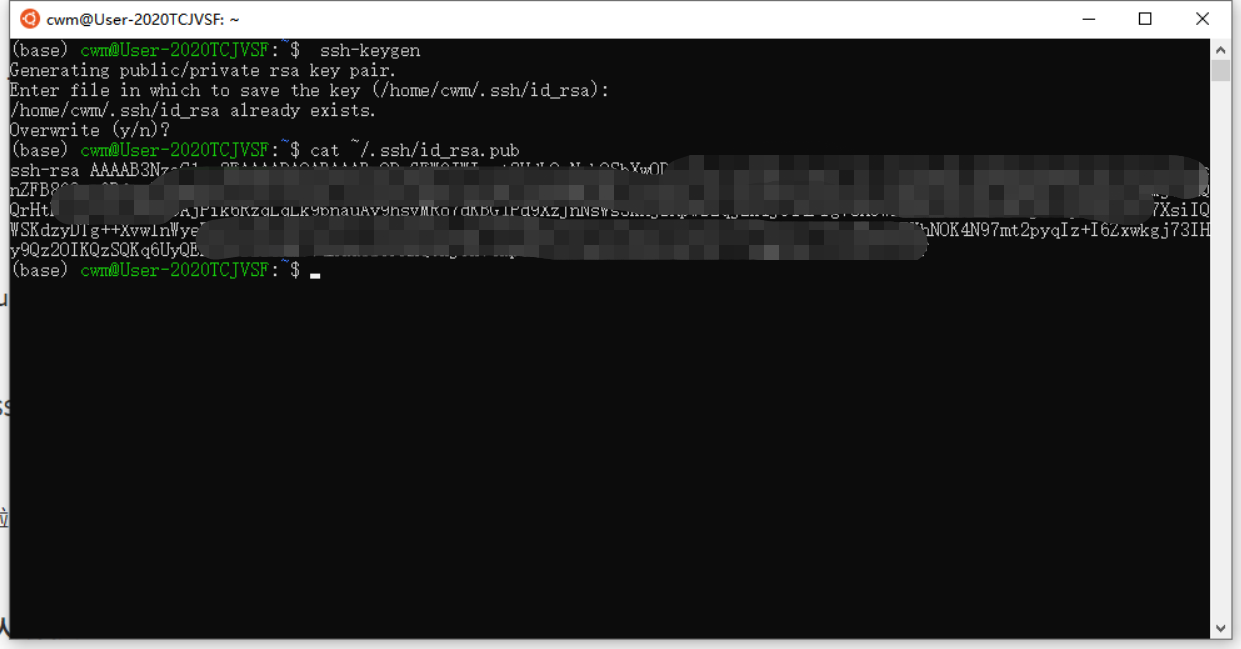


图 15 获取本机密钥

* 1. 然后使用教学团队用户的账号，如上述提供的root账号，登录到Gitlab仓库，如图 16所示，通过点击用户头像 >> 偏好设置 >> SSH密钥，进入到SSH密钥添加页面，在框中输入上述复制的个人密钥完成密钥添加的操作。

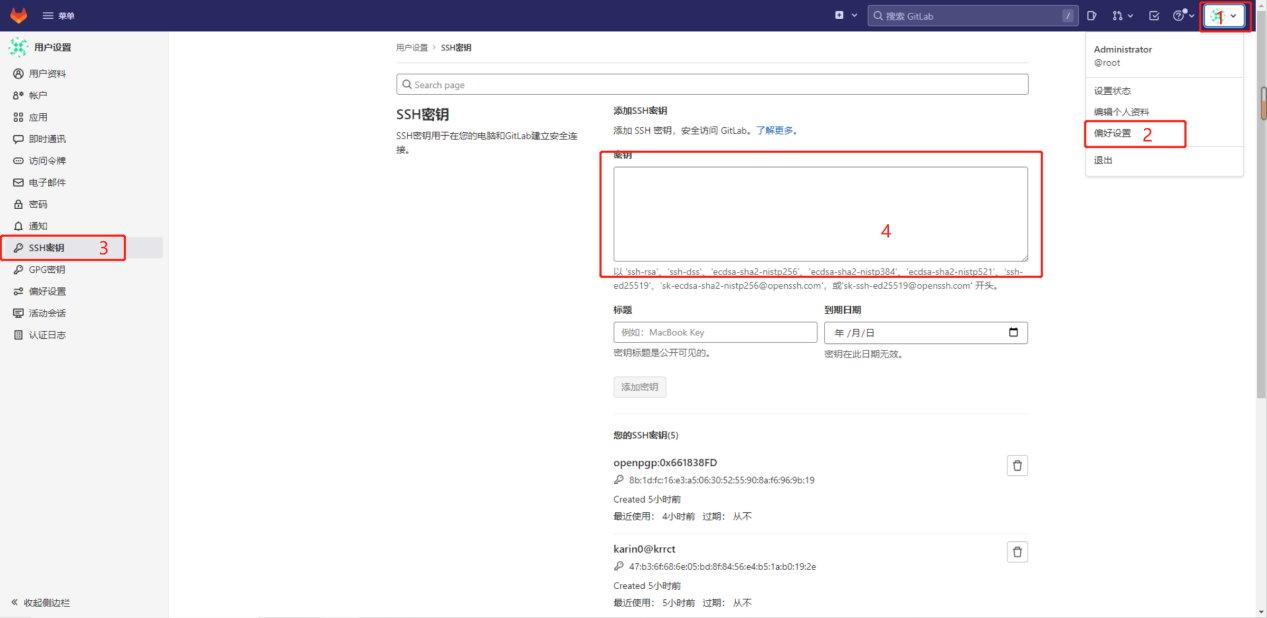


图 16 Gitlab上密钥添加流程

1. 建立联系后，可以在本地将所需文件复制到clone的仓库之中。
2. 通过Git进行提交：
   1. git add .
   2. git commit -m “commit\_info”
   3. git push -u origin HEAD

## 学生成绩查询

教学团队可登录到信息统计平台 http://122.9.36.214:81/ ，查看学生在各实验中的评测得分，以及各学生的实验统计信息。测试环境中的教师用户名为“admin”，密码为“inTDPC8oHCxvK2S7TmJv”。如图17所示，教学团队可以查询到各个实验的总体情况以及各个学生不同实验下的得分情况。



图17 学生成绩统计页面

如图18、19所示，点击一名学生后，教学团队可以具体查看该学生在各实验中的评测记录和得分进度，以及该学生在Web IDE中的开发时长、编辑历史等行为信息。



图18 学生评测记录统计

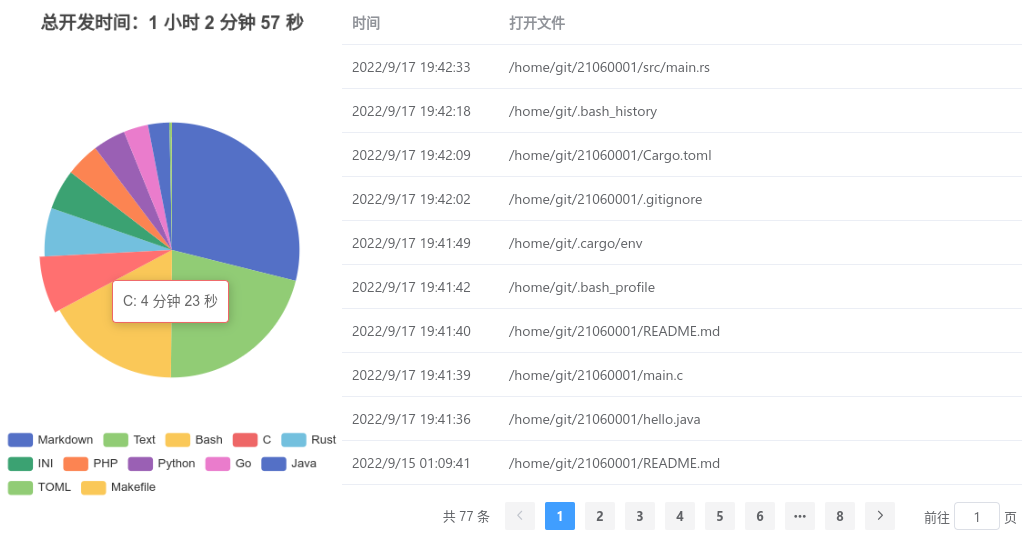


图19 学生学习行为统计