

北京邮电大学

课程指导手册



课程：编译原理与技术课程设计

学期：2023-2024 学年春季学期

学院：计算机学院（国家示范性软件学院）

目录

一、实验平台	3
1.课程链接	3
2.运行环境	3
二、平台操作	3
1.实践平台页面	3
2.代码上传	5
三、评测	11
1.评测流程	11
2.评测要求	11
四、网络资料推荐	12

一、实验平台

1.课程链接

<https://www.educoder.net/classrooms/ixfbck2m?code=5FWNE>

2.运行环境

- Debian GNU/Linux 9
- Flex 2.6.1
- Bison 3.0.4
- cmake 3.28.2
- Python 3.7.5
- c++ 7.3.0

二、平台操作

【注】以实践项目“Pascal-S 语言编译程序”为例

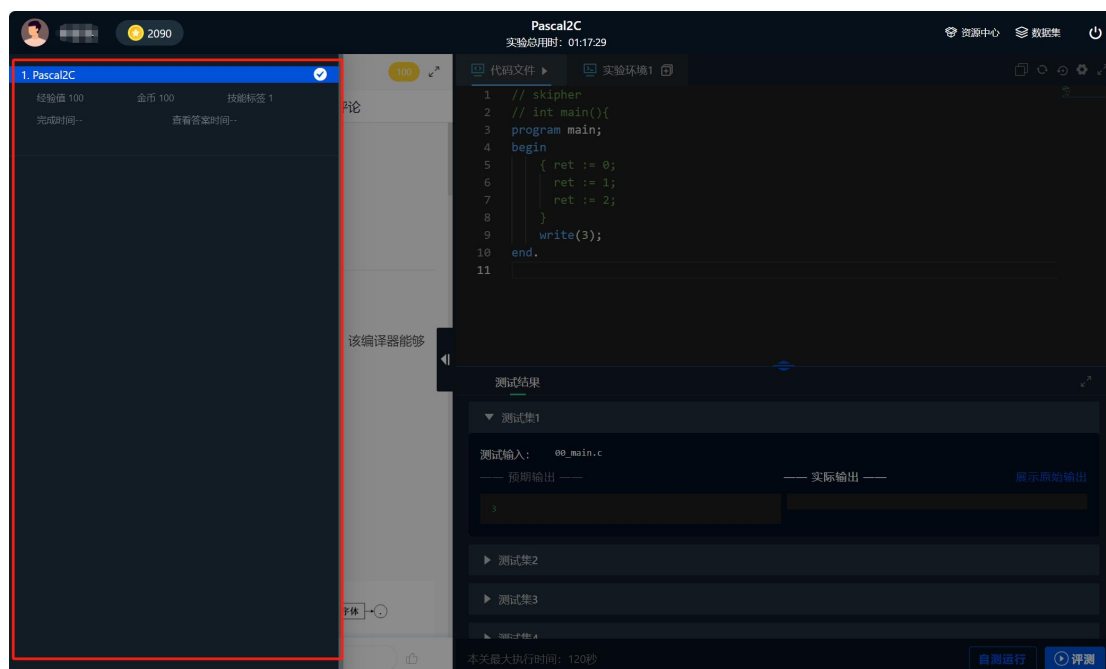
1.实践平台页面

The screenshot displays the Educoder platform interface for the 'Pascal2C' task. The interface is divided into two main panels. The left panel, labeled '第1关: Pascal2C', contains a '任务要求' (Task Requirements) section with a '任务描述' (Task Description) and '相关知识' (Relevant Knowledge) section. The right panel, labeled 'Pascal2C', shows a code editor with a Pascal program, a '测试结果' (Test Results) section, and a '评测' (Evaluation) button. Red boxes and numbers 1-8 highlight specific UI elements:

- 1: User profile icon and score (2090)
- 2: Task requirements section
- 3: Code file dropdown menu
- 4: Experiment environment dropdown menu
- 5: Run button
- 6: Code editor area
- 7: Test results section
- 8: Evaluation button

1) 查看关卡（框图 1）

可以查看本实验的关卡。每个关卡题目、评测数据、评测要求不同。根据设置，关卡可以被跳过。遇到困难关卡，可尝试跳过，优先解决能力范围内的题目。



2) 查看关卡题目（框图 2）

展示了本关卡相关信息，包括题目和要求等。

3) 编辑器（框图 3、框图 6）

可通过“代码文件”右侧的倒三角，查看框图 6 对应的哪个文件。通过在框图 6 内的编辑器，可以对对应文件进行修改。

4) git 仓库（框图 5）

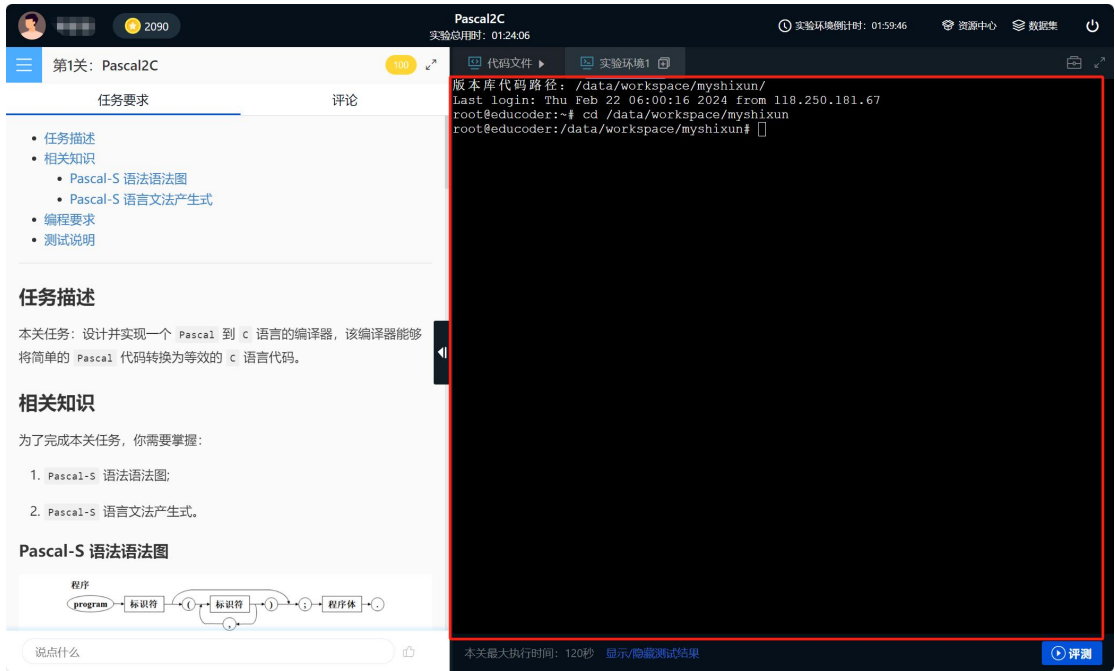
点击此按钮，可以直接赋值本实践项目对应的 git 仓库。可以通过 git 仓库管理，来上传、修改实践项目的文件。每个账号都有一个独立的仓库。具体操作见后文。

5) 评测（框图 7、框图 8）

框图 7 展示了测试用例的预期输出和实际输出，用户可以通过此处来比对。框图 8 中的按钮运行测试脚本。自测不上传测试数据，评测上传。

6) 命令行模式（框图 4）

可以连接项目仓库，以命令行的方式展示。需要使用命令进入项目：
cd /data/workspace/myshixun



2.代码上传

“代码文件”只能修改文件，所以需要创建新的文件时，需要使用命令行窗口、git 仓库操作或者平台组件。

1) 命令行模式

可以使用 linux 的文件系统处理命令完成操作。下面给出常用命令：

命令	格式	选项	解释
ls	ls [--options] [file/directory]	-a:显示所有 -l:长格式显示 -R:递归展开	展示路径下文件
touch	touch [file..]	*	创建新文件 更新文件时间
cp	cp [--options] src dst	-r:递归复制	复制文件
rm	rm [--options] file/directory	-r:递归删除	删除文件
cat	cat file	*	打印文件
mv	mv file	*	移动文件
chmod	chmod rights file	*	修改文件权限
mkdir	mkdir dir_name	*	新建文件夹

可以使用 vi 编辑器进行文件编辑。下面给出常用命令：

命令	解释
vi file	打开文件
:q	退出编辑器
:q!	强制退出编辑器
:wq/:x	保存修改并退出编辑器
h,i,k,l	左、下、上、右移光标
0	移到当前行首
\$	移到当前行尾
gg	移到文件开头
G	移到文件结尾
Ctrl+f	向前翻页
Ctrl+b	向后翻页
i	在光标前插入文本
a	在光标后插入文本
o	在当前行下方插入新行
dd	删除当前行
yy	复制当前行
p	复制粘贴的内容
/keyword	向下搜索关键词
?keyword	向上搜索关键词
:s/old/new/g	替换当前行的所有 old 为 new
:%s/old/new/g	替换文件所有 old 为 new

2) Git 仓库模式

1. Git 的安装和配置可以参考网络资料
2. 在上述框图 5 的位置复制 git 仓库地址（每个用户的仓库地址不同）
3. 在本地选中新建仓库的位置打开命令行窗口，然后使用命令拉取仓库：
git clone your_repository_url

首次拉取需要输入账密，对应头歌平台的账密。

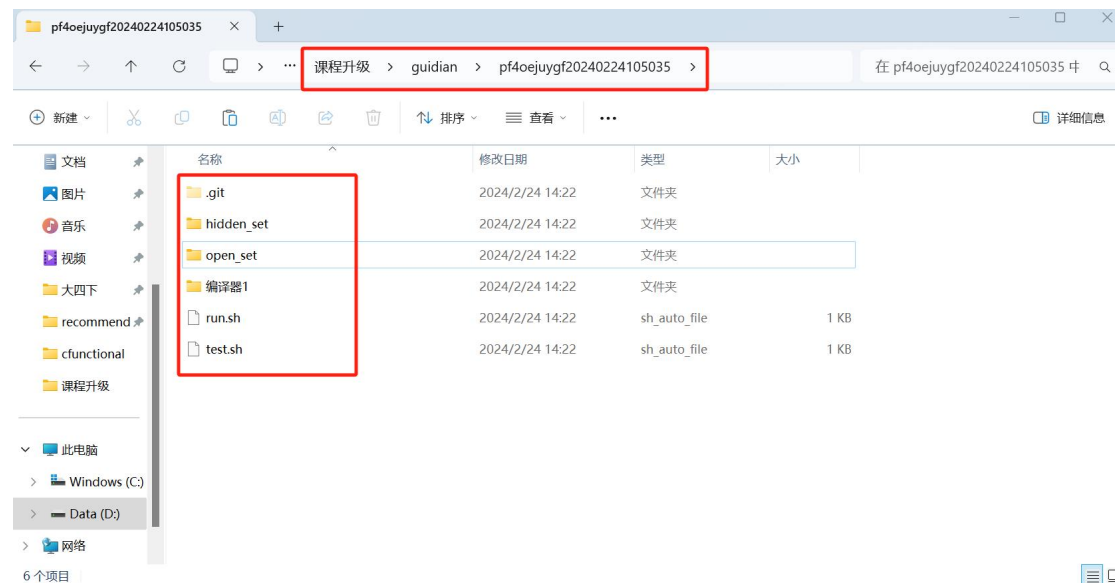
```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [版本 10.0.22621.3007]
(c) Microsoft Corporation. 保留所有权利。

D:\AA-school\Postgraduate\课程升级\guidian>git --version
git version 2.42.0.windows.2

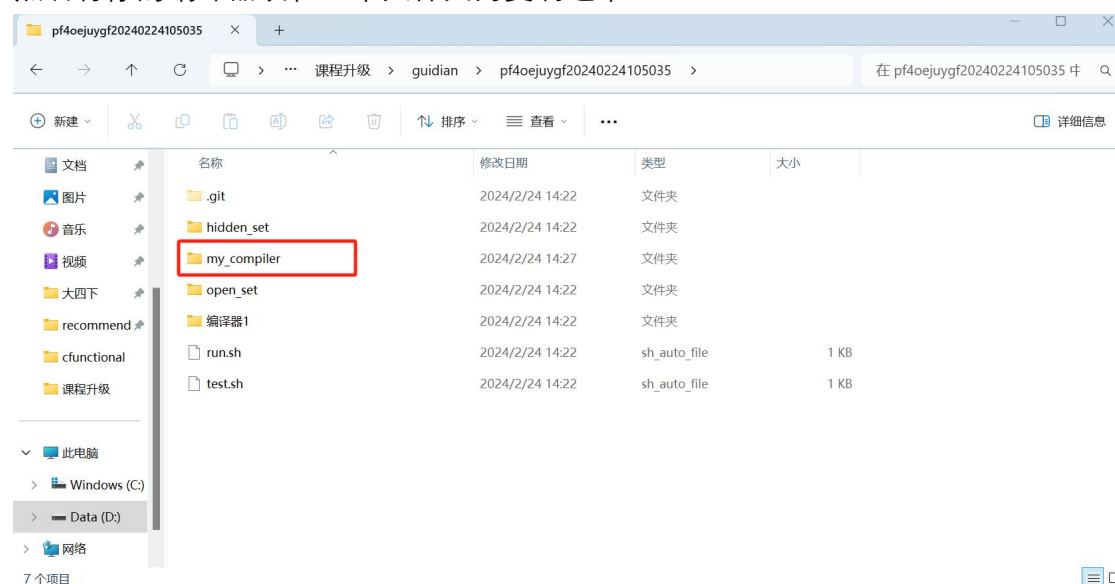
D:\AA-school\Postgraduate\课程升级\guidian>git clone https://git.educoder.net/pfrj7kmfh/pf4oejuygf20240224105035.git
Cloning into 'pf4oejuygf20240224105035'...
remote: Enumerating objects: 486, done.
remote: Counting objects: 100% (486/486), done.
remote: Compressing objects: 100% (389/389), done.
remote: Total 486 (delta 66), reused 350 (delta 4), pack-reused 0
Receiving objects: 100% (486/486), 8.43 MiB | 9.22 MiB/s, done.
Resolving deltas: 100% (66/66), done.

D:\AA-school\Postgraduate\课程升级\guidian>
```

之后就能在文件管理器页面，查看仓库代码了



然后将你的编译器装在一个文件夹内复制过来



然后再该目录下的命令行界面使用 git 命令上传文件：

git add .

git commit -m 'upload'

git push

```
D:\AA-school\Postgraduate\课程升级\guidian\pf4oejuygf20240224105035>git add .
D:\AA-school\Postgraduate\课程升级\guidian\pf4oejuygf20240224105035>git commit -m 'upload'
[master e832d07] 'upload'
68 files changed, 98093 insertions(+)
create mode 100644 my_compiler/.github/workflows/pytest.yml
create mode 100644 my_compiler/CMakeLists.txt
create mode 100644 my_compiler/LICENSE
create mode 100644 my_compiler/document/Wirth-PascalS.pdf
create mode 100644 my_compiler/document/google-style.jpg
create mode 100644 my_compiler/document/zh-google-styleguide-20220529.pdf
create mode 100644 "my_compiler/document/\343\200\212\347\274\226\350\257\221\345\216\237\347\220\206\344\270\216\346\2
12\200\346\234\257\350\257\276\347\250\213\350\256\276\350\256\241\343\200\213-202302.pdf"
create mode 100644 my_compiler/include/ast.h
create mode 100644 my_compiler/include/compiler.h
create mode 100644 my_compiler/include/parser.h
create mode 100644 my_compiler/include/parser.tab.h
```

```
D:\AA-school\Postgraduate\课程升级\guidian\pf4oejuygf20240224105035>git push
Enumerating objects: 86, done.
Counting objects: 100% (86/86), done.
Delta compression using up to 8 threads
Compressing objects: 100% (71/71), done.
Writing objects: 100% (85/85), 8.14 MiB | 2.63 MiB/s, done.
Total 85 (delta 2), reused 83 (delta 1), pack-reused 0
To https://git.educoder.net/pfrj7kmfh/pf4oejuygf20240224105035.git
8df4966..e832d07  master -> master
```

然后在平台刷新，就能看见上传的文件了

```
root@educoder:/data/workspace/myshixun# ls
hidden_set my_compiler open_set run.sh test.sh 编译器1
root@educoder:/data/workspace/myshixun#
```

- 如果没有更新，在平台命令行重新 clone 即可。

3) 平台组件模式

按照如下图片操作，即可通过平台组件，上传相应的文件。默认保存在 /data/workspace/userfiles 目录下：

The screenshot displays the Educoder platform interface for the 'Pascal2C' task. The left sidebar shows the task requirements and related knowledge. The main area shows a terminal window with the following commands and output:

```
root@educoder:/data/workspace# ls
myshixun
root@educoder:/data/workspace# git clone https://git.educoder.net/pfrj7kmfh/pf4oejuygf20240224105035.git
Cloning into 'pf4oejuygf20240224105035'...
Username for 'https://git.educoder.net': 13452669329
Password for 'https://13452669329@git.educoder.net':
remote: Enumerating objects: 488, done.
remote: Counting objects: 100% (488/488), done.
remote: Compressing objects: 100% (391/391), done.
remote: Total 488 (delta 65), reused 350 (delta 4), pack-reused 0
Receiving objects: 100% (488/488), 8.43 MiB | 10.25 MiB/s, done.
Resolving deltas: 100% (65/65), done.
Checking out files: 100% (370/370), done.
root@educoder:/data/workspace# ls
myshixun pf4oejuygf20240224105035
root@educoder:/data/workspace# cd pf4oejuygf20240224105035
root@educoder:/data/workspace/pf4oejuygf20240224105035# ls
hidden_set my_compiler open_set run.sh test.sh 编译器1
root@educoder:/data/workspace/pf4oejuygf20240224105035# cd ..
root@educoder:/data/workspace# rm -rf myshixun
root@educoder:/data/workspace# ls
pf4oejuygf20240224105035
root@educoder:/data/workspace# cp -r pf4oejuygf20240224105035 myshixun
root@educoder:/data/workspace# ls
myshixun pf4oejuygf20240224105035
root@educoder:/data/workspace# cd myshixun
root@educoder:/data/workspace/myshixun# ls
hidden_set my_compiler open_set run.sh test.sh 编译器1
root@educoder:/data/workspace/myshixun# ls
hidden_set my_compiler open_set run.sh test.sh 编译器1
root@educoder:/data/workspace/myshixun# cd ..
root@educoder:/data/workspace# ls
myshixun pf4oejuygf20240224105035 userfiles
root@educoder:/data/workspace# cd userfiles
root@educoder:/data/workspace/userfiles# ls
使用说明.txt
root@educoder:/data/workspace/userfiles#
```


李彦言2090

Pascal2C

实验总用时: 01:14:40

实验环境倒计时: 01:52:14

功能

第1关: Pascal2C

任务要求

评论

任务描述

相关知识

编程要求

测试说明

任务描述

相关知识

Pascal-S 语法语法图

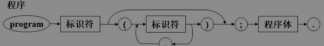
本关任务: 设计并实现一个 Pascal 到 C 语言的编译器, 该编译器能够将简单的 Pascal 代码转换为等效的 C 语言代码。

为了完成本关任务, 你需要掌握:

1. Pascal-S 语法语法图;

2. Pascal-S 语言文法产生式。

Pascal-S 语法语法图



root@educoder:/data/workspace# ls

myshixun

root@educoder:/data/workspace# git clone https://git.educoder.net/pf4oejuygf20240224105035.git

Cloning into 'pf4oejuygf20240224105035'...

Username for 'https://git.educoder.net': 13452669329

Password for 'https://13452669329@git.educoder.net':

remote: Enumerating objects: 488, done.

remote: Counting objects: 100% (488/488), done.

remote: Compressing objects: 100% (391/391), done.

remote: Total 488 (delta 65), reused 350 (delta 4), pack-reused 0

Resolving deltas: 100% (65/65), done.

Checking out files: 100% (370/370), done.

root@educoder:/data/workspace# ls

myshixun pf4oejuygf20240224105035

root@educoder:/data/workspace# cd pf4oejuygf20240224105035

root@educoder:/data/workspace/pf4oejuygf20240224105035# ls

hidden_set my_compiler open_set run.sh test.sh 编译器1

root@educoder:/data/workspace/pf4oejuygf20240224105035# cd ..

root@educoder:/data/workspace# ls

myshixun pf4oejuygf20240224105035

root@educoder:/data/workspace# cp -r pf4oejuygf20240224105035 myshixun

root@educoder:/data/workspace# cd myshixun

root@educoder:/data/workspace/myshixun# ls

hidden_set my_compiler open_set run.sh test.sh 编译器1

root@educoder:/data/workspace/myshixun# cd ..

root@educoder:/data/workspace# ls

myshixun pf4oejuygf20240224105035 userfiles

root@educoder:/data/workspace# cd userfiles

root@educoder:/data/workspace/userfiles# ls

使用说明.txt

root@educoder:/data/workspace/userfiles#

重置环境

上传文件

下载文件

服务预览

复制版本库地址

SSH直连

关卡配置信息

闯关

测试集解锁

实训基本信息

实训难易度

实验环境

实验容器网络

120秒 显示/隐藏测试结果

李彦言2090

Pascal2C

实验总用时: 01:14:57

实验环境倒计时: 01:52:14

资源中心

数据集

电源

第1关: Pascal2C

任务要求

评论

任务描述

相关知识

编程要求

测试说明

任务描述

相关知识

Pascal-S 语法语法图

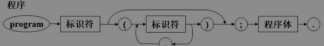
本关任务: 设计并实现一个 Pascal 到 C 语言的编译器, 该编译器能够将简单的 Pascal 代码转换为等效的 C 语言代码。

为了完成本关任务, 你需要掌握:

1. Pascal-S 语法语法图;

2. Pascal-S 语言文法产生式。

Pascal-S 语法语法图



root@educoder:/data/workspace# ls

myshixun

root@educoder:/data/workspace# git clone https://git.educoder.net/pf4oejuygf20240224105035.git

Cloning into 'pf4oejuygf20240224105035'...

Username for 'https://git.educoder.net': 13452669329

Password for 'https://13452669329@git.educoder.net':

remote: Enumerating objects: 488, done.

remote: Counting objects: 100% (488/488), done.

remote: Compressing objects: 100% (391/391), done.

remote: Total 488 (delta 65), reused 350 (delta 4), pack-reused 0

Resolving deltas: 100% (65/65), done.

Checking out files: 100% (370/370), done.

root@educoder:/data/workspace# ls

myshixun pf4oejuygf20240224105035

root@educoder:/data/workspace# cd pf4oejuygf20240224105035

root@educoder:/data/workspace/pf4oejuygf20240224105035# ls

hidden_set my_compiler open_set run.sh test.sh 编译器1

root@educoder:/data/workspace/pf4oejuygf20240224105035# cd ..

root@educoder:/data/workspace# ls

myshixun pf4oejuygf20240224105035 userfiles

root@educoder:/data/workspace# cd userfiles

root@educoder:/data/workspace/userfiles# ls

使用说明.txt

root@educoder:/data/workspace/userfiles#

文件上传

点击上传图标, 选择要上传的文件或将文件拖拽到此 (单个文件最大限制为20MB)

默认存放目录路径为 "/data/workspace/userfiles"

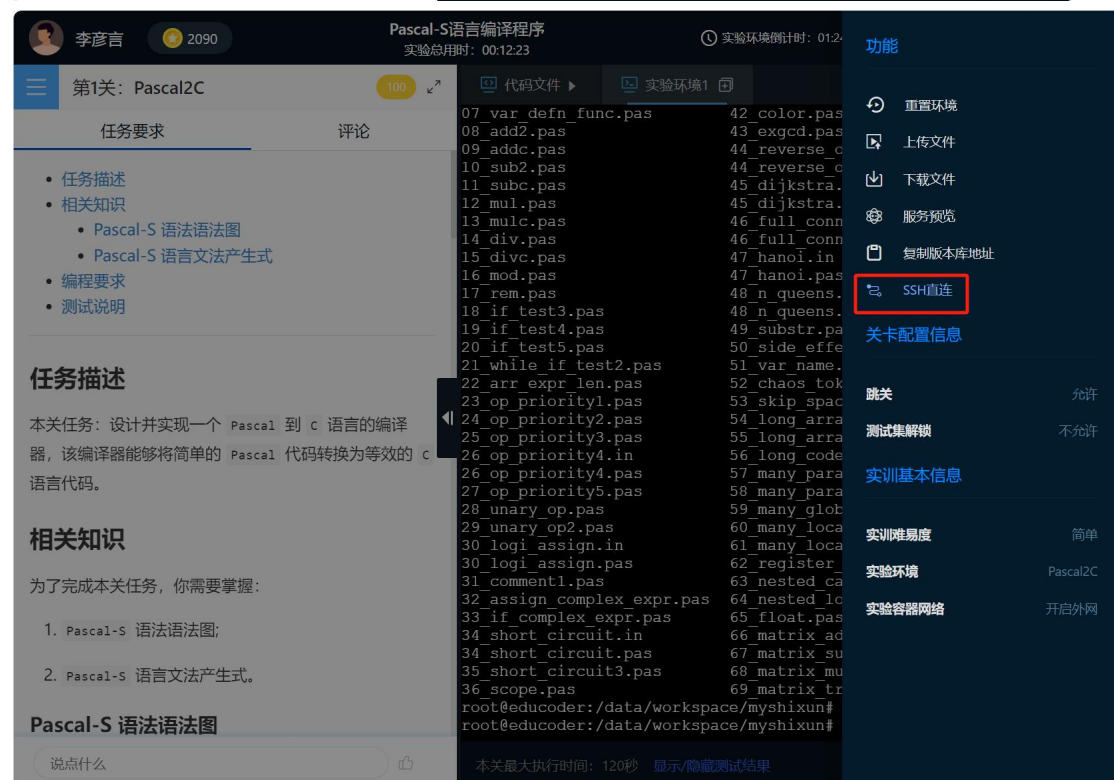
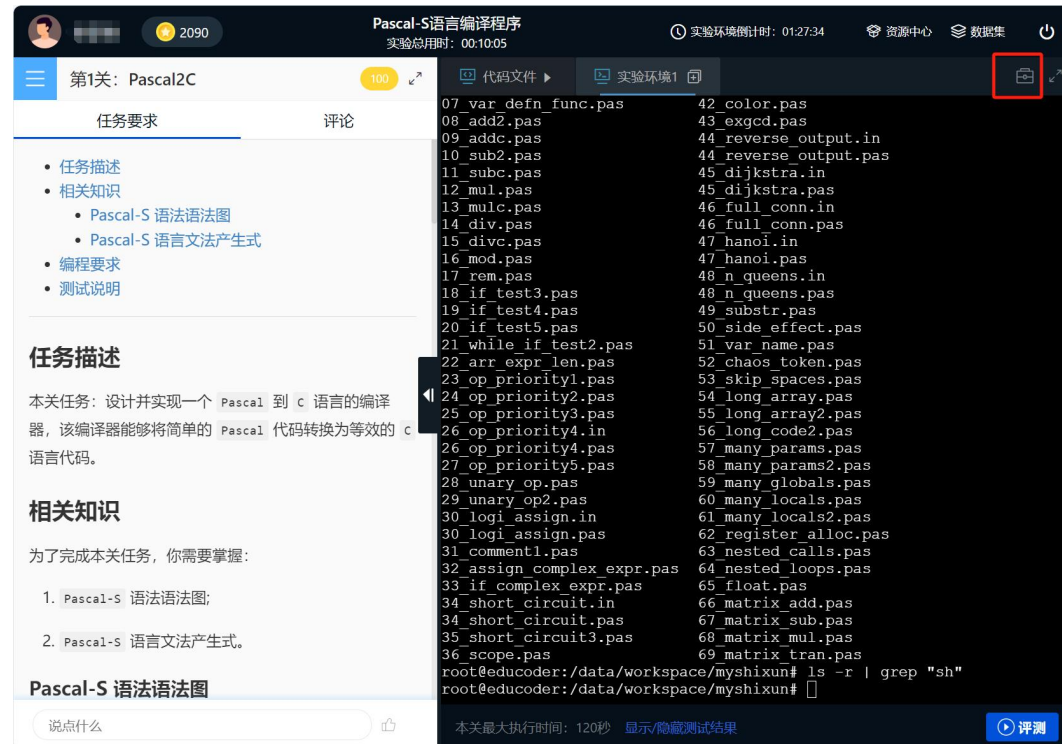
关闭

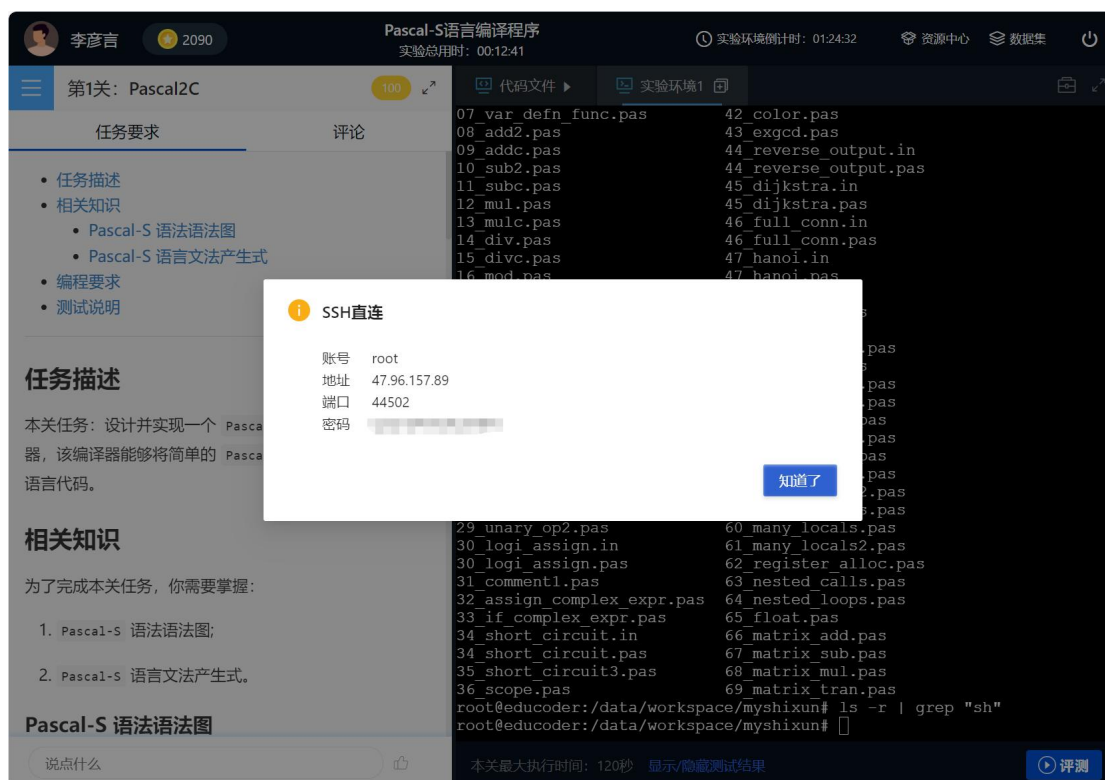
120秒 显示/隐藏测试结果

评测

4) SSH 直连

点击红色框图，选择“SSH 直连”，就可以查看相关的连接信息。然后使用常用的远程连接软件，如 PuTTY、X-Shell、Win SCP，来操作平台文件。建议使用 Win SCP 传输文件，图形化操作更方便。【注意 Win SCP 的传输是不稳定的，传输文件夹可能会漏文件或错码，将文件打包压缩后传输，在平台或其他连接软件上操作解压可解决】





三、评测

1.评测流程

现有两个测试集——open_set 开放测试集和 hidden_set 隐藏测试集。开放测试集总共有 70 个测试用例，隐藏测试集总共有 25 个测试用例。你能在 /data/workspace/myshixun/ open_set 目录下看到开放测试集的相关信息。

测试脚本将自动扫描两个测试集，并用 /data/workspace/myshixun/bin 目录下的编译器将测试集中的每一个测试用例（pas 文件）转为对应的 c 语言程序，然后编译并运行这个 c 文件，并将输入输出自动重定向到指定文件，并将输出同答案对比，来判断是否通过用例。

2.评测要求

1) 将可执行的编译器文件命名为 **pascc**；（注：区分大小写）

2) 将该文件存储在 **/data/workspace/myshixun/bin** 目录下；

3) 须保证编译器能接受命令行参数 `-i filename.pas`, 输出同名 `c` 文件, 并存储在源文件同目录下。

e.g. `pascc -i open_set/00_main.pas` 应该将 `open_set/00_main.pas` 翻译成 `00_main.c`, 然后存在 `open_set` 中。

1) 上传可执行文件

你可以在本地将编译器编译成可执行文件后, 直接上传, 无需使用平台的编译工具。

2) 上传源文件

上传所有源文件后, 使用平台自带的环境 (Debian GNU/Linux 9、Flex 2.6.1、Bison 3.0.4、cmake 3.28.2、Python 3.7.5、c++ 7.3.0) 进行编译。

建议使用 `cmake` 便捷编译。减少编译造成的困扰。

四、网络资料推荐

Mac Git 安装: [Mac 安装 Git mac git-CSDN 博客](#)

Win Git 安装: [如何在在 win10 上安装 Git \(图文并茂\) 超详细 git 安装-CSDN 博客](#)

Git 文档: [Git - 安装 Git \(git-scm.com\)](#)

Git 配置: [超详细的 Git 下载安装、环境变量配置教程 安装好 git 之后如何配置-CSDN 博客](#)

代码风格: [Google 开源项目风格指南——中文版 — Google 开源项目风格指南 \(zh-google-styleguide.readthedocs.io\)](#)

PuTTY: [【工具】putty 下载、安装、使用和常用设置教程-CSDN 博客](#)

X-Shell: [XShell 免费版的安装配置教程以及使用教程 \(超级详细、保姆级\) xshell 安装教程-CSDN 博客](#)

Win SCP: [文件传输工具 WinSCP 下载安装教程 winscp 安装教程-CSDN 博客](#)