
编译原理实验平台 学生使用手册

Student Manual of NJU Compiler OJ

Version 1.1.0

目录

1	前言	1
1.1	平台介绍	1
1.2	编译原理实验注意事项	1
1.2.1	学生实验环境.....	1
1.2.2	实验编译方式.....	1
1.2.3	实验输出格式.....	1
2	用户模块.....	2
2.1	登录.....	2
3	实验模块.....	2
3.1	实验列表	2
3.2	实验详情页	3
3.3	提交代码	4

[illegible]

1 前言

1.1 平台介绍

暂略。

1.2 编译原理实验注意事项

1.2.1 学生实验环境

学生的实验环境务必与如下所述环境相同，因为 OJ 会在该环境下进行测试。若使用不同的实验环境造成问题，学生请自行解决。

硬件环境：

- (1) CPU: AMD x86_64 架构
- (2) 内存: 大于等于 1GB

软件环境：

- (1) 操作系统, Ubuntu 20.04 64bit (<https://ubuntu.com>)
- (2) C 语言编译器, GCC 7.5.0 (<https://gcc.gnu.org>)
- (3) 词法分析器生成器, Flex 2.6.4 (<https://github.com/westes/flex>)
- (4) 语法分析器生成器, Bison 3.5.1 (<https://www.gnu.org/software/bison/>)
- (5) MIPS 模拟器, SPIM 8.0 (<http://spimsimulator.sourceforge.net/>)
- (6) Python2 环境, Python 2.7 (<https://python.org>)

以下提供一份在 Ubuntu 20.04 下安装对应实验环境的命令行脚本：

```
# 换镜像源，加速下载
cp /etc/apt/sources.list /etc/apt/sources.list.back
sed -i 's/archive.ubuntu.com/mirrors.nju.edu.cn/g' /etc/apt/sources.list
sed -i 's/security.ubuntu.com/mirrors.nju.edu.cn/g' /etc/apt/sources.list
# 下载相关软件，并设置使用 gcc-7 版本
apt update
apt install -qq -y build-essential flex bison libbison-dev gcc-7
update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 100
```

1.2.2 实验编译方式

以课程主页中提供的 Makefile 文件（**不可修改**）为主，以 C 语言实现，不可依赖除 Flex 和 Bison 外的其他库。

1.2.3 实验输出格式

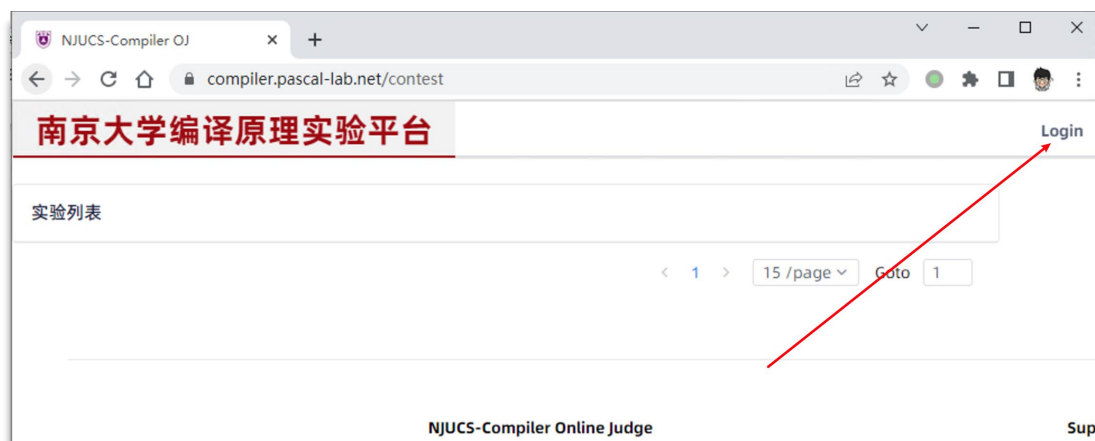
关于实验一、实验二，错误输出和语法树输出都**必须**通过标准输出（stdout）进行输出，而不是标准错误（stderr）。

2 用户模块

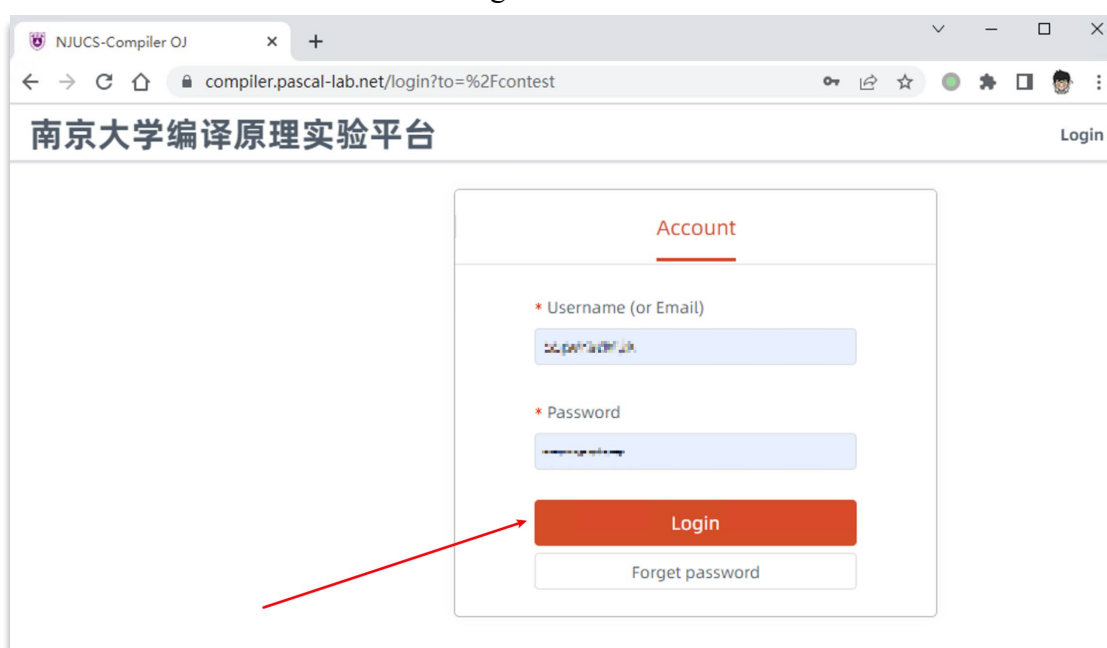
为方便学生使用和助教统一管理，平台账户统一由助教进行导入，并发送账号密码给选课学生。

2.1 登录

(1) 打开网站 <https://compiler.pascal-lab.net/>，点击右上角的“Login”。



(2) 填入账号密码，点击“Login”。



3 实验模块

3.1 实验列表

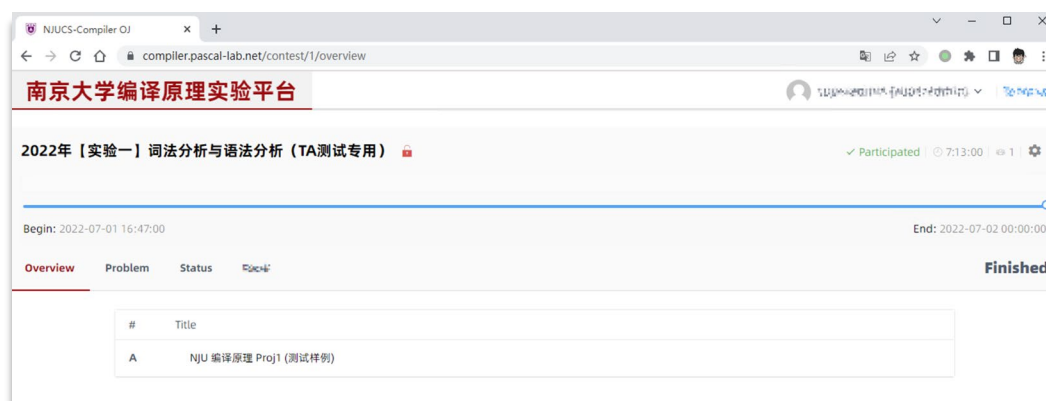
在网站主页的“实验列表”下选择一个要进入的实验，点击进入。



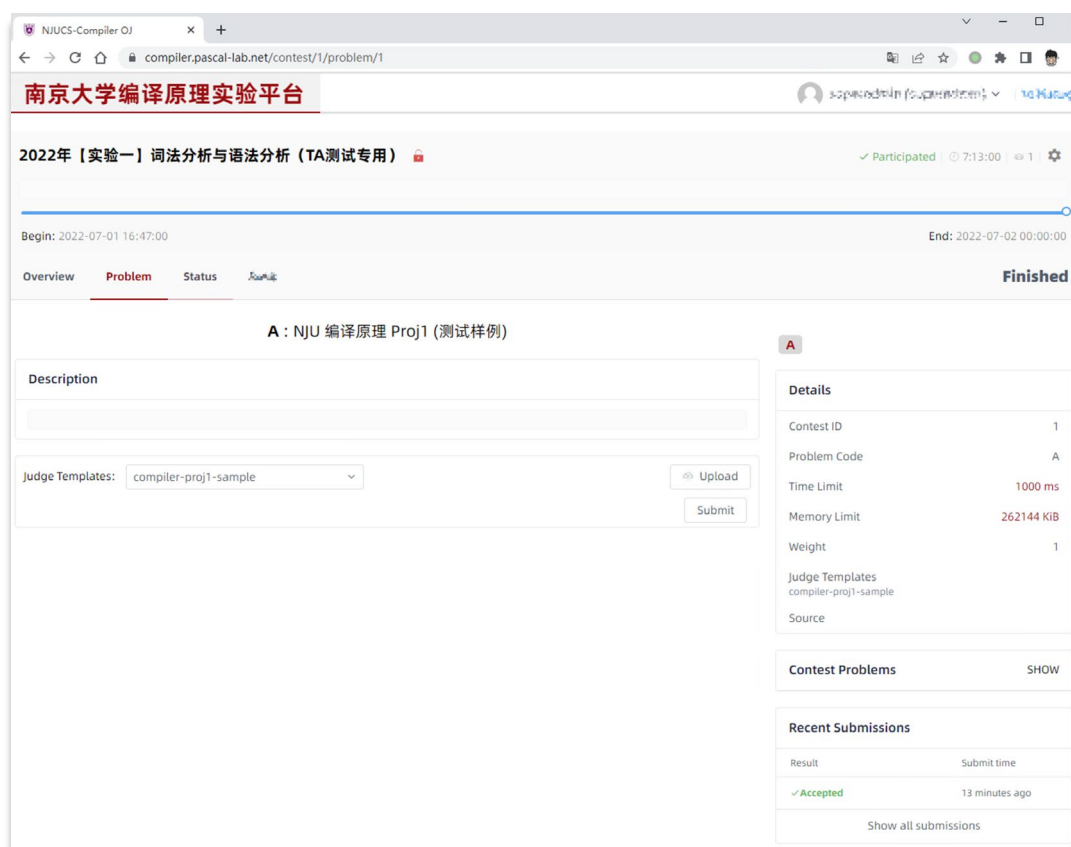
3.2 实验详情页

实验详情页内有 3 个 Tab 页，分别为“Overview”、“Problem”、“Status”，其中：

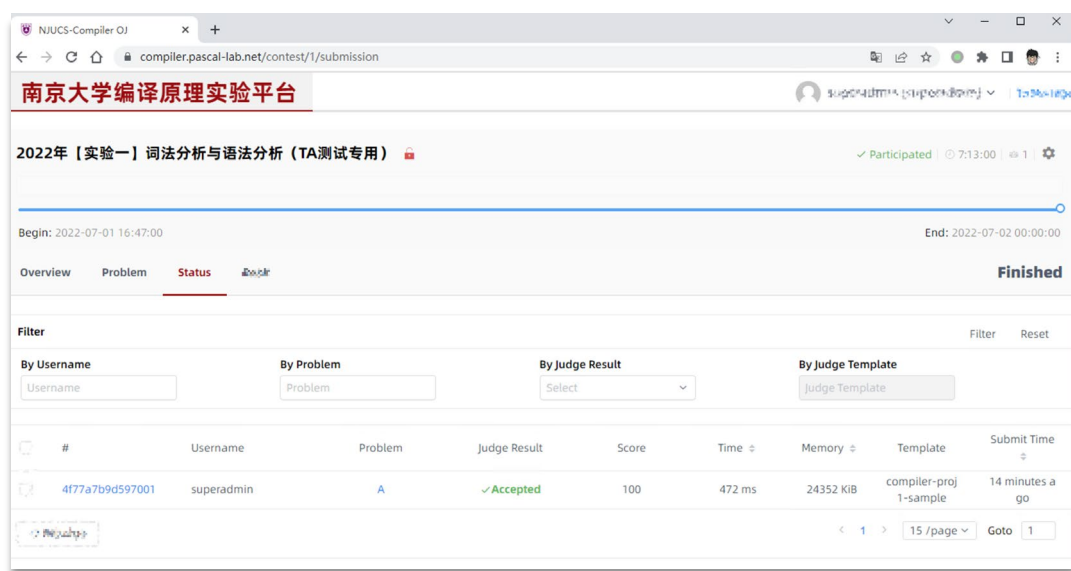
(1)“Overview”显示该实验中设置的题目数量及其基本信息(题号、名称)；点击具体的题目，会跳到该题的“Problem”页。



(2)“Problem”是题目的详情页，里面有“Description”板块，即题目的详细介绍，“Details”板块，即题目的基本配置（时空限制等），“Recent Submissions”板块，即最近提交的代码及其简要评测情况，还有题目切换、评测模板切换等功能。



（3）“Status”是提交代码的列表页，其列出了用户在该实验中所提交的所有代码及其评测情况。点击评测 ID（图中是“4f77a7b9d597001”）可以进入提交代码的详情页。



3.3 提交代码

首先按照实验要求，将实验文件打包成如下结构的 Zip 包：

张三_1234567.zip

```
├── Code
│   ├── lexical.l
│   ├── main.c
│   ├── Makefile
│   └── ...
├── README
└── report.pdf
```

注意不要嵌套，一个错误的打包例子是：

以下是错误的 Zip 文件结构

张三_1234567.zip

|

张三_1234567

```
├── Code
│   ├── lexical.l
│   ├── main.c
│   ├── Makefile
│   └── ...
├── README
└── report.pdf
```

选择一道题进入“Problem”的详情页后，点击“Upload”并选中刚刚打包好的 Zip 文件，随后点击“Submit”进行提交。

