

# 希冀信息类专业实验教学科研一体化平台 建设方案

[www.educg.net](http://www.educg.net)

郑州云海科技有限公司  
北京航空航天大学计算机学院教学成果转化平台

# 方案大纲

## 一、平台整体架构

## 二、在线实验环境

- 代码自动评测
- 通用自动评测
- 云桌面
- Jupyter

## 三、课程管理功能

- 在线作业
- 在线考试
- 在线实验

## 四、专业支撑方案

- 计算机专业
- 人工智能专业
- 大数据专业
- 机器人专业
- 区块链专业
- 集成电路专业
- 金融科技专业
- 信息安全专业

## 五、通用科研平台

- 科研项目管理
- 计算资源管理
- 科研环境管理

## 六、建设方案总结



---

## 平台整体架构

计算机

人工智能

大数据

机器人

区块链

集成电路

金融科技

信息安全

代码自动评测实验环境

通用自动评测实验环境

云桌面自动评测实验环境

Jupyter自动评测实验环境

开放、可扩展的信息类专业教学实验科研一体化支撑平台

云计算基础设施

(服务器、存储、网络、GPU、FPGA、嵌入式设备)

## 学生

“任何人、任何时间、任何地点” 均能开展实验学习

综合能力评价：汇集学生在整个培养阶段的学习过程数据、项目实践数据、考试成绩数据等

工程教育：动手能力、创新能力、跨界整合能力

## 教师

**开放：**只需要专注教学资源建设

- ✓ Docker开发环境 + 实验指导
- ✓ 题库

**体验：**教学全过程支持，教师视角，实用易用

**生态：**高质量的资源，能够获得持续性的经济收益

# 希冀平台整体架构



区块链与数字经济	区块链案例分析与设计	金融模型的数值方法	大数据与金融	数字集成电路设计	CMOS模拟集成电路设计	拒绝服务攻击	入侵检测
区块链原理与系统	矿机技术与开发实践	金融科技学	量化交易	VLSI数字信号处理	VLSI数字通信原理与设计	特洛伊木马	漏洞挖掘分析
共识机制与博弈论	Solidity与智能合约	软件工程	计量经济学	处理器设计	智能处理器设计	数据库安全	缓冲区溢出
密码学基础	信息安全与数字认证	数据结构与算法	人工智能原理及应用	FPGA技术	计算机组成原理	访问控制	安全编程规范
Go语言程序设计	分布式系统与P2P网络	高级语言程序设计	区块链技术与应用	电路与Spice	数字电路与Verilog	密码学基础	防火墙与入侵防御
区块链专业		金融科技专业		集成电路专业		信息安全专业	
软件工程	并行计算	智能计算系统	智能应用系统综合设计	数据采集与爬虫	大数据综合案例	机器人运动规划	机器人综合案例
数据库	计算机网络	强化学习	智能控制与机器人	Python数据科学	Hadoop大数据技术	机器人感知系统	自动驾驶
算法分析与设计	编译原理	深度学习	自然语言处理	文本挖掘	NoSQL分布式数据库	机器人运动控制	无人机智能控制
数据结构	操作系统	机器学习	计算机视觉	数据挖掘	Storm流计算技术	机器人系统仿真	机械臂智能控制
高级语言程序设计	计算机组成原理	最优化方法	模式识别	数据可视化	Spark大数据技术	机器人操作系统	移动机器人智能控制
计算机专业		人工智能专业		数据科学专业		机器人专业	

代码自动评测实验环境 + 通用自动评测实验环境 + 云桌面自动评测实验环境 + Jupyter自动评测实验环境



# 希冀平台整体架构



- 特色
- ✓ 一体化
- ✓ 全面支撑
- ✓ 开放可扩展
- 目标
- ✓ 实践能力
- ✓ 资源建设
- ✓ 教育大数据

区块链与数字经济	区块链案例分析与设计	金融模型的数值方法	大数据与金融	数字集成电路设计	CMOS模拟集成电路设计	拒绝服务攻击	入侵检测
区块链原理与系统	矿机技术与开发实践	金融科技学	量化交易	VLSI数字信号处理	VLSI数字通信原理与设计	特洛伊木马	漏洞挖掘分析
共识机制与博弈论	Solidity与智能合约	软件工程	计量经济学	处理器设计	智能处理器设计	数据库安全	缓冲区溢出
密码学基础	信息安全与数字认证	数据结构与算法	人工智能原理及应用	FPGA技术	计算机组成原理	访问控制	安全编程规范
Go语言程序设计	分布式系统与P2P网络	高级语言程序设计	区块链技术的应用	电路与Spice	数字电路与Verilog	密码学基础	防火墙与入侵防御
区块链专业		金融科技专业		集成电路专业		信息安全专业	
软件工程	并行计算	智能计算系统	智能应用系统综合设计	数据采集与爬虫	大数据综合案例	机器人运动规划	机器人综合案例
数据库	计算机网络	强化学习	智能控制与机器人	Python数据科学	Hadoop大数据技术	机器人感知系统	自动驾驶
算法分析与设计	编译原理	深度学习	自然语言处理	文本挖掘	NoSQL分布式数据库	机器人运动控制	无人机智能控制
数据结构	操作系统	机器学习	计算机视觉	数据挖掘	Storm流计算技术	机器人系统仿真	机械臂智能控制
高级语言程序设计	计算机组成原理	最优化方法	模式识别	数据可视化	Spark大数据技术	机器人操作系统	移动机器人智能控制
计算机专业		人工智能专业		数据科学专业		机器人专业	

代码自动评测实验环境 + 通用自动评测实验环境 + 云桌面自动评测实验环境 + Jupyter自动评测实验环境



## Dashboard

F11切换全屏 / 服务器时间: 2020-9-13 22:00:54 / CourseGrading 5.3.0 在线版

315

访问人数



1231

在线课程



2663610

评测次数



15927172

总代码行

34356 / 59861

今年注册人数 / 累计人数

CPU: 4.7%

内存 16.4% | 网络收发 17/134(KB/s)

### 评测时间分布

24小时

一周

### 最近一个月的活跃课程



学生人数 在线学习时间(分钟)



### 正在进行的考试

Python程序设计	18计科Python补考   106人	659分钟后开始
Python程序设计	18计科Python补考   106人	659分钟后开始
程序设计基础2	2020-2021学年 第一学期《程序设计基础2》开学补考   3人	659分钟后开始

## 计算机专业课一体化平台

### 精简风格

CG Online Judge 教师登录

**最新公告**

- 【数据结构与程序设计 (信息大类)】第三次作业提交选择库答案 2018/04/20
- 【数据结构与程序设计 (信息大类)】第一次作业提交选择库答案 2018/03/30
- 【数据结构与程序设计 (信息大类)】期中考试通知 2018/04/18
- 【数据结构与程序设计 (信息大类)】期中考试通知 2018/04/13
- 【数据结构与程序设计 (信息大类)】第二次作业提交选择库答案 2018/04/09



47  
在线用户

**学生入口**

学号

密码

登录

北京航空航天大学 计算机学院  
 重置密码, 请与当前的任课教师联系

## 在线实验中心

### 实验中心

登录 教学系统 教师入口



机器学习-第二次实验作业 机器学习-第一次实验作业 深度学习-第一次实验作业 机器学习第三次实验作业-Jupyter  
 信息安全第一次作业 操作系统第一次实验作业

虚拟仿真 Jupyter实验 公告

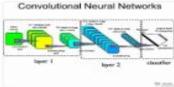
机器学习-第二次实验作业 | 机器学习-第二次实验作业



基于CNN的车牌号码识别系统



人工神经网络模型原理



卷积神经网络



神经网络游乐场: 可视化神经网络

## CourseGrading 课程中心

### 课程中心

CG Online Judge 教师登录

**最新公告**

- 【高级语言程序设计 (化工)】17级第一次作业提交选择库答案 2017/11/01
- 【算法设计与分析】2017-2018学年作业提交和附件下载 2017/10/19
- 【算法设计与分析】2017-2018学年第二次附加题问答题 2017/10/19
- 【高级语言程序设计 (化工)】欢迎2017级同学访问本系统 2016/09/20
- 【编译技术】编译技术及课程设计要求安排 2017/09/19



29  
在线用户

**学生入口**

学号

密码

登录



文件集测试



算法设计与分析 (北京学院)



检验软件工程



计算机科学研究方法与实践

## OJ风格

CG Online Judge 首页 题库 竞赛 排行榜 讨论区 F.A.Qs





icpc International Collegiate Programming Contest

icpc.foundation

本次比赛由CG-计算机专业课一体化支撑平台提供技术支持, 系统官网: [educg.net](http://educg.net)

大家在使用系统过程中, 如果需要帮助请加入QQ群: 328666683

评测环境:

- Java: JDK1.7
- C: gcc-4.4.7
- C++: g++-4.4.7
- Python: python3.5.5

最近6个月刷题榜

#	用户	解决问题	提交问题
1	201612300101	8	8
2	LSJ	6	9
3	M201742030112	5	5
4	616896236	2	2
5	QNU2017416611	2	2
6	201701001075	2	2
7	M201742060347	2	2
8	201701003017	2	3
9	M201742060238	2	4
10	17102233	1	1

[完整榜单 >](#)

刷题榜



---

**在线实验环境**

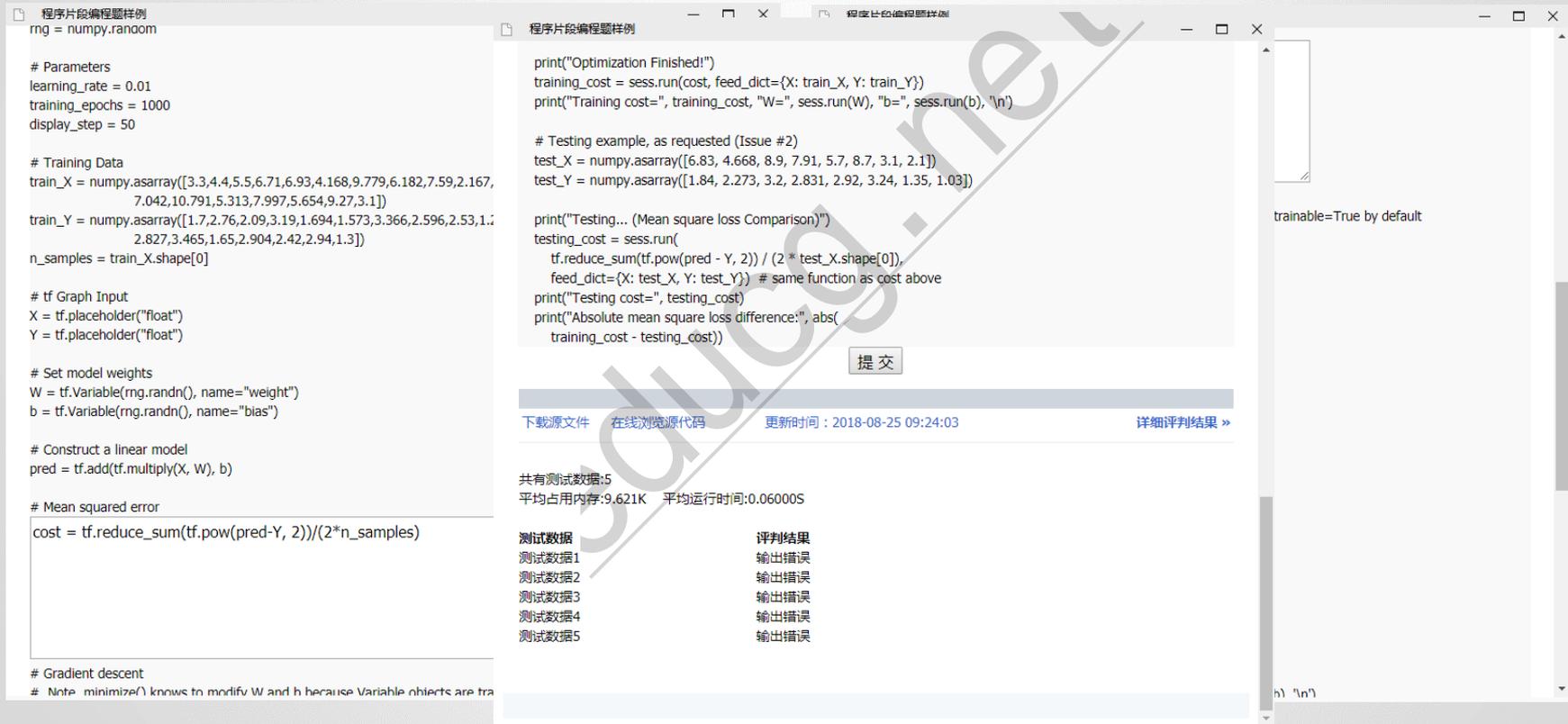


01

---

**代码自动评测**

- **自动评测：**减轻教师工作量，提高学生工程实践能力，支持Python, TensorFlow, Keras等语言和库。



The screenshot displays a web-based interface for code execution. On the left, a code editor shows Python code for a linear model using TensorFlow and NumPy. The code includes parameter settings, data loading, model construction, and a mean squared error calculation. On the right, the execution output is shown, including a '提交' (Submit) button and a table of test results. The table indicates that all five test cases failed due to output errors. Below the table, there are statistics for the execution, such as the number of test cases (5), average memory usage (9.621K), and average execution time (0.06000S). A '详细评判结果' (Detailed Evaluation Results) link is also present.

```
rng = numpy.random

# Parameters
learning_rate = 0.01
training_epochs = 1000
display_step = 50

# Training Data
train_X = numpy.asarray([3.3,4.4,5.5,6.71,6.93,4.168,9.779,6.182,7.59,2.167,
7.042,10.791,5.313,7.997,5.654,9.27,3.1])
train_Y = numpy.asarray([1.7,2.76,2.09,3.19,1.694,1.573,3.366,2.596,2.53,1.
2.827,3.465,1.65,2.904,2.42,2.94,1.3])
n_samples = train_X.shape[0]

# tf Graph Input
X = tf.placeholder("float")
Y = tf.placeholder("float")

# Set model weights
W = tf.Variable(rng.randn(), name="weight")
b = tf.Variable(rng.randn(), name="bias")

# Construct a linear model
pred = tf.add(tf.multiply(X, W), b)

# Mean squared error
cost = tf.reduce_sum(tf.pow(pred-Y, 2))/(2*n_samples)

# Gradient descent
# Note minimize() knows to modify W and b because Variable objects are tra
```

print("Optimization Finished!")
training\_cost = sess.run(cost, feed\_dict={X: train\_X, Y: train\_Y})
print("Training cost=", training\_cost, "W=", sess.run(W), "b=", sess.run(b), '\n')

# Testing example, as requested (Issue #2)
test\_X = numpy.asarray([6.83, 4.668, 8.9, 7.91, 5.7, 8.7, 3.1, 2.1])
test\_Y = numpy.asarray([1.84, 2.273, 3.2, 2.831, 2.92, 3.24, 1.35, 1.03])

print("Testing... (Mean square loss Comparison)")
testing\_cost = sess.run(
 tf.reduce\_sum(tf.pow(pred - Y, 2)) / (2 \* test\_X.shape[0]),
 feed\_dict={X: test\_X, Y: test\_Y}) # same function as cost above
print("Testing cost=", testing\_cost)
print("Absolute mean square loss difference:", abs(
 training\_cost - testing\_cost))

提交

下载源文件 在线浏览源代码 更新时间：2018-08-25 09:24:03 详细评判结果 >>

共有测试数据:5  
平均占用内存:9.621K 平均运行时间:0.06000S

测试数据	评判结果
测试数据1	输出错误
测试数据2	输出错误
测试数据3	输出错误
测试数据4	输出错误
测试数据5	输出错误

h) '\n')



02

---

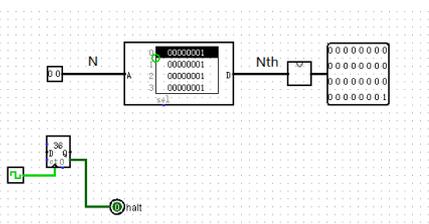
**通用自动评测**

**关键技术：**教师自定义评测kernel(Docker)，平台负责：负载均衡、异常处理、查重.....

## Web提交

使用Logisim搭建一个根据输入序号*x*计算出对应序号正弦波那款数 $b[x]$ 的电路（输入序号0对应输出数0，输入序号1对应输出数1，输入序号2对应输出数1，以此类推）并提交。

- 输入：N (32bit无符号数)
- 输出：Nth (32bit无符号数，表示第N个正弦波那款数)
- 文件内模块名：main
- 测试电路图：



- 测试要求：在64个周期内计算出结果并稳定输出，在结果未计算出之前输出端口输出0。
- HINT：矩阵乘法的快速幂

提交源文件：

注意：只能用 LOGISIM 语言编写程序。

未选择任何文件

[下载源文件](#) [在线浏览源代码](#)

共有测试数据：7

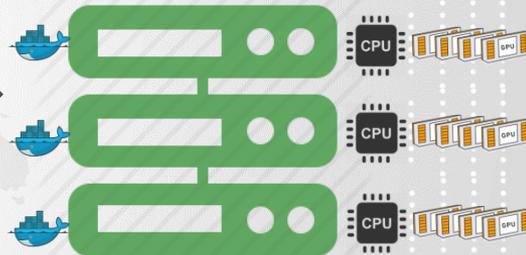
测试数据	评判结果
测试数据1	完全正确

## 评测机（集群）

调度学生提交的实验材料

负载均衡

反馈实验报告





03

---

云桌面

指导手册和实验桌面集为一体

支持云桌面分辨率自适应调整

支持多人协同、桌面分享

支持桌面全屏、桌面一键还原

支持本地与云桌面的文件互传

支持在线撰写实验报告

支持一键桌面截屏



The screenshot displays a web-based interface for a neural network experiment. On the left, a document editor titled '深度学习基础：神经网络' (Deep Learning Basics: Neural Networks) is open, showing a table of contents with sections like '1. 实验内容' (Experiment Content), '2. 实验目标' (Experiment Objectives), '3. 实验知识点' (Experiment Knowledge Points), '4. 实验环境' (Experiment Environment), and '5. 预备知识' (Prerequisite Knowledge). A context menu is visible over the document, offering options such as '重新连接桌面' (Reconnect Desktop), '剪切板' (Clipboard), '下载远程桌面内的文件' (Download files from remote desktop), '上传文件至远程桌面' (Upload files to remote desktop), '重新启动' (Restart), and '桌面还原' (Restore desktop). A green button labeled '下一个步骤' (Next Step) is at the bottom of the document.

On the right, the 'A Neural Network Playground' interface is shown. It features a control panel with 'Epoch 001,077', 'Learning rate 0.03', 'Activation Tanh', 'Regularization None', and 'Regularization rate 0'. The 'Problem type' is set to 'Classification'. The main area displays a neural network diagram with '3 HIDDEN LAYERS' (8, 8, and 4 neurons respectively). The input layer includes features like  $X_1$ ,  $X_2$ ,  $X_1^2$ ,  $X_2^2$ ,  $X_1X_2$ ,  $\sin(X_1)$ , and  $\sin(X_2)$ . The output is a heatmap of a handwritten digit '4' on a spiral dataset. The 'Test loss' is 0.074 and 'Training loss' is 0.005. A 'REGENERATE' button is at the bottom left of the playground.

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## • 一键制作云桌面实验环境



The screenshot displays a web browser window on the left side of a cloud desktop interface. The browser shows a page for 'Tianshou' (天授), a reinforcement learning platform. The desktop environment on the right features a blue background with the '希冀 Course Grading' logo and the text 'Right here waiting for you' and '在这里等着你'. Desktop icons include Home, Logisim, File System, Trash, Firefox Web Browser, Chromium Web Browser, and IDLE 3. A sidebar menu on the left contains options like '重新连接桌面', '剪贴板', '保存镜像', '共享桌面', '下载远程桌面内的文件', '上传文件至远程桌面', '评测脚本管理', '重新启动', '桌面还原', and '帮助'. A red arrow points to the '保存镜像' option.

远程桌面工作台

工作笔记 桌面全屏 隐藏桌面 更多

重新连接桌面

剪贴板

保存镜像

共享桌面

下载远程桌面内的文件

上传文件至远程桌面

评测脚本管理

重新启动

桌面还原

帮助

### Tianshou

Tianshou (天授) is a reinforcement learning platform building reinforcement learning libraries, which are mainly targeting nested classes, unfriendly API, or slow-speed. Tianshou provides a modular framework and pythonic API for building the deep reinforcement learning framework with the least number of lines of code. The supported interfaces include:

- Policy Gradient (PG)
- Deep Q-Network (DQN)
- Double DQN
- Dueling DQN
- Advantage Actor-Critic (A2C)
- Deep Deterministic Policy Gradient (DDPG)
- Proximal Policy Optimization (PPO)
- Twin Delayed DDPG (TD3)
- Soft Actor-Critic (SAC)
- Vanilla Imitation Learning
- Prioritized Experience Replay (PER)
- Generalized Advantage Estimator (GAE)

Here is Tianshou's other features:

- Elegant framework, using only ~2000 lines of code
- Support parallel environment simulation (synchronous or asynchronous) for all algorithms Usage
- Support recurrent state representation in actor network and critic network (RNN-style training for POMDP) Usage
- Support any type of environment state/action (e.g. a dict, a self-defined class, ...) Usage
- Support customized training process Usage
- Support n-step returns estimation and prioritized experience replay for all Q-learning Usage

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## 一键添加云桌面实验环境

test.educg.net/admin/index.jsp

希冀 系统管理 课程管理 CG Online Judge 科研平台 工程教育认证 GitLab 帮助 云桌面实验平台 (2020)

公告与问答 课程信息 学生 作业 考试 在线实验 在线课堂 答疑 汇总成绩 助教 设置 进入学生端

实验资源库  
实验概览  
添加实验  
回收站

实验作业管理  
实验作业概览  
布置实验作业  
回收站

计算资源管理

云桌面工作台  
远程桌面工作台  
Jupyter工作台

数据统计  
在线实验活动记录  
在线实验明细数据  
常见问题

远程桌面工作台 / 在桌面工作台可以制作实验环境或者进行教学演示。 / 镜像加入公共镜像库之后，才能被用作实验环境，并注意提醒管理员部署该镜像。

打开远程桌面工作台

#	保存的镜像	镜像描述	桌面镜像	加入公共实验镜像库	加入科研平台镜像库	加入通用评测环境	删除	编辑	最近修改时间
1		From cg/vnc/logisim:v1.0.3   所在主机 deskdocker2   Docker启动选项 --privileged --user 0 --hostname=cg   By admin   By admin	cg/vnc/logisim:v1.0.4 1.30 GB <a href="#">查看日志</a>						admin 2020-08-26 20:58:40
2		From cg/ubuntu-1804:v1.0.2   所在主机 oslabanddeskdocker   Docker启动选项 --privileged --user 0 --hostname=cg	cg/vnc/ic:v1.0.1 38.32 GB <a href="#">查看日志</a>						admin 2020-08-25 10:55:48
3		From cg/logisim:v1.0.2   所在主机 deskdocker2   Docker启动选项 --privileged --user 0 --hostname=cg   By admin   By admin	cg/vnc/logisim:v1.0.3 1.30 GB <a href="#">查看日志</a>	已加入 <a href="#">查看日志</a>					admin 2020-08-10 23:02:25
4		From cg/kvm:v1.0.4   所在主机 deskdocker2   Docker启动选项 --privileged --user 0 --hostname=cg	cg/vnc/rh19:v1.0.1 4.21 GB <a href="#">查看日志</a>	已加入 <a href="#">查看日志</a>					admin 2020-07-20 21:20:36

正在等待 test.educg.net 的响应...

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## 添加云桌面实验



The screenshot shows the '添加云桌面实验' (Add Cloud Desktop Experiment) page in the CG Online Judge system. The interface includes a top navigation bar with links for '系统管理', '课程管理', 'CG Online Judge', '科研平台', '工程教育认证', and 'GitLab'. A secondary navigation bar contains '公告与问卷', '课程信息', '学生', '作业', '考试', '在线实验' (highlighted), '在线课堂', '答疑', '汇总成绩', '助教', '设置', and '进入学...'. The left sidebar lists various management options like '实验资源库', '实验作业管理', '云桌面工作台', and '数据统计'. The main content area is titled '虚拟桌面实验环境 / 1. 填写实验介绍 / 2. 添加实验步骤'. It features a form with the following fields:

- 实验名称**: A text input field containing '实验名称'.
- 实验介绍**: A text area containing '实验介绍, 描述一下你的实验'.
- 封面图片**: A section with the instruction '使用图片是为了呈现出更好的展现效果。点击下图上传图片。建议尺寸: 280 × 157'. Below it is a preview image of the CG logo on a blue background.
- 实验环境**: A dropdown menu showing 'cg/vnc/rl:v1.0.2 / By admin | By admin | By admin |'.
- 内存上限**: A text input field with '8192' and a unit selector set to 'MB'.

At the bottom, a note states: '一个docker实例使用的内存上限, 内存上限决定了一个主机上最多同时启动的Docker数量。必须设置内存限制, 否则个别用户申请过大的内存, 会引起同一主机上其它Docker实例运行失败。'

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## 添加云桌面实验

希冀 系统管理 课程管理 CG Online Judge 科研平台 工程教育认证 GitLab 帮助 云桌面实验平台

公告与问卷 课程信息 学生 作业 考试 在线实验 在线课堂 答疑 汇总成绩 助教 设置 进入

**实验资源库**  
实验概览  
添加实验  
回收站

**实验作业管理**  
实验作业概览  
布置实验作业  
回收站

**计算资源管理**

**云桌面工作台**  
远程桌面工作台  
Jupyter工作台

**数据统计**  
在线实验活动记录  
在线实验明细数据  
常见问题

**实验环境** cg/vnc/rt:v1.0.2 / By admin | By admin | By admin |

**内存上限** 8192 MB  
一个docker实例使用的内存上限，内存上限决定了一个主机上最多同时启动的Docker数量。必须设置内存限制，否则个别用户申请过大的内存，会引起同一主机上其它Docker实例运行失败。

**实验编号** 必须为整数。可以不用填写，系统自动生成唯一编号  
自定义的编号，可用于实验排序。若不填写，系统会自动生成一个全局唯一的编号。

**设置标签** 并行计算 KNN 《机器学习实战》  
+ 添加新标签

**参考答案** ① 点击上传参考答案 ...  
可选项。可以在实验作业处，随时开启公布答案开关。已经上传的参考答案：

**实验指导编辑器** 富文本编辑器 Markdown编辑器  
注意，一旦选定编辑器，当前实验不能再切换编辑器。  
富文本编辑器：类似“word”的在线编辑器。 Markdown编辑器：使用Markdown和Latex语法撰写教程，支持图片，但不支持视频、音频等多媒体。

**实例重用**  实例重用  
一般需要勾选，如果勾选，将会重用之前作业分配的来自同一个镜像的实例，除非该实例已经被删除。

高级

保存

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## • 添加评测脚本



The screenshot displays a cloud desktop interface. On the left, a web browser window is open to the URL `test.educg.net/admin/courseAdmin/expAdmin/guides/expDesktopWorkbench.jsp?hostname=deskdocker2&token=eae77aade24e3c5e691798f574b0e43&vnc_host=test.educg.net&vnc_port=&imgaeID=cg-vn...`. The page content includes a sidebar with '工作笔记' (Work Notes) and a main area titled 'Tianshou'. The 'Tianshou' page describes it as a reinforcement learning platform and lists various algorithms such as Policy Gradient (PG), Deep Q-Network (DQN), Double DQN, Dueling DQN, Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), Twin Delayed DDPG (TD3), Soft Actor-Critic (SAC), Vanilla Imitation Learning, Prioritized Experience Replay (PER), and Generalized Advantage Estimator (GAE). A '评测脚本管理' (Evaluation Script Management) link is highlighted with a red arrow in the sidebar.

On the right, the desktop environment is visible with a blue background and the '希冀 Course Grading' logo. The desktop contains several application icons: Home, Logisim, File System, examples, Trash, Firefox Web Browser, Chromium Web Browser, and IDLE 3. The text 'Right here waiting for you' and '在这里等着你' is displayed on the desktop.

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## • 添加评测脚本

实验概览 / 编辑实验与步骤



深度学习

深度学习实战:车牌号码识别

基于CNN的车牌号码识别系统

实验编号: 25316549778875

编辑实验介绍与环境

1 实验介绍	> 自动评测	✈	🗑	⬆
2 设置工作目录	> 自动评测	✈	🗑	⬆
3 创建项目工程	> 自动评测	✈	🗑	⬆
4 实验步骤: 实验框架	> 自动评测	✈	🗑	⬆
5 实验步骤: 【从Tensorflow中导出手写体数字图片】	> 自动评测	✈	🗑	⬆
6 实验步骤: 【从Tensorflow中导出手写体数字图片】-实现导出的Python代码	> 自动评测	✈	🗑	⬆
7 实验步骤: 【从Tensorflow中导出手写体数字图片】-运行代码	> 自动评测	✈	🗑	⬆
8 实验步骤: 【从Tensorflow中导出手写体数字图片】-观察导出图片	> 自动评测	✈	🗑	⬆
9 实验步骤: 【TensorFlow训练和识别/分类自定义图片】	> 自动评测	✈	🗑	⬆
10 实验步骤: 【TensorFlow训练和识别/分类自定义图片】-自定义图片集	> 自动评测	✈	🗑	⬆
11 实验步骤: 【基于MNIST数据集实现车牌识别初步版本】	> 自动评测	✈	🗑	⬆
12 实验步骤: 【基于MNIST数据集实现车牌识别初步版本】-挑选图片	> 自动评测	✈	🗑	⬆
13 实验步骤: 【基于MNIST数据集实现车牌识别初步版本】-输入数据	> 自动评测	✈	🗑	⬆

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

- 添加评测脚本



☰ 课程信息   学生   作业   考试   在线实验   在线课堂   答疑   汇总成绩   助教   设置

### 设置评测脚本

选择评测脚本

judge.sh / 2020-05-12 15:34:44

该步骤的分数占比

100 %

该步骤得分占题目总分比例，取值范围(0, 100]。

评测超时时间

5 秒

评测超过该时间，将会强制结束。 0秒不限时。

确定

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

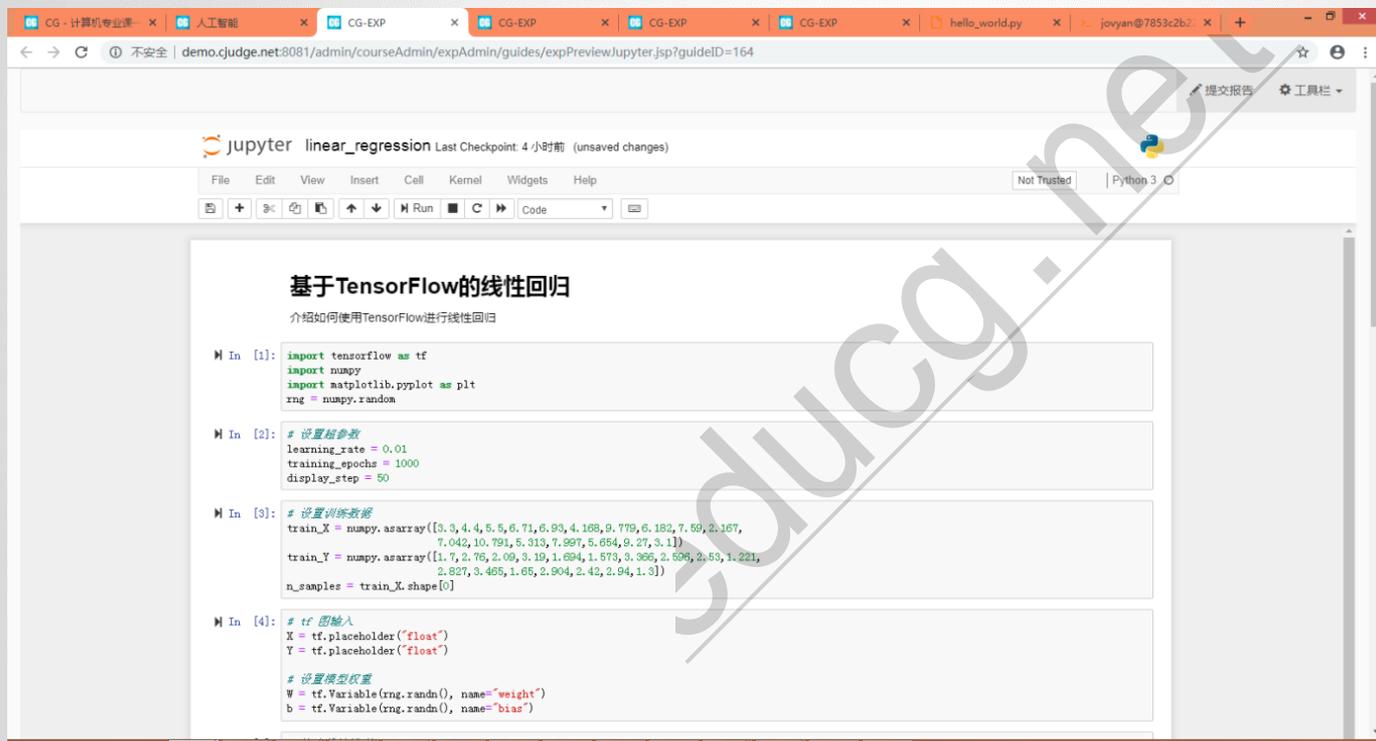


04

---

**Jupyter**

- 基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验



```
In [1]: import tensorflow as tf
import numpy
import matplotlib.pyplot as plt
rng = numpy.random

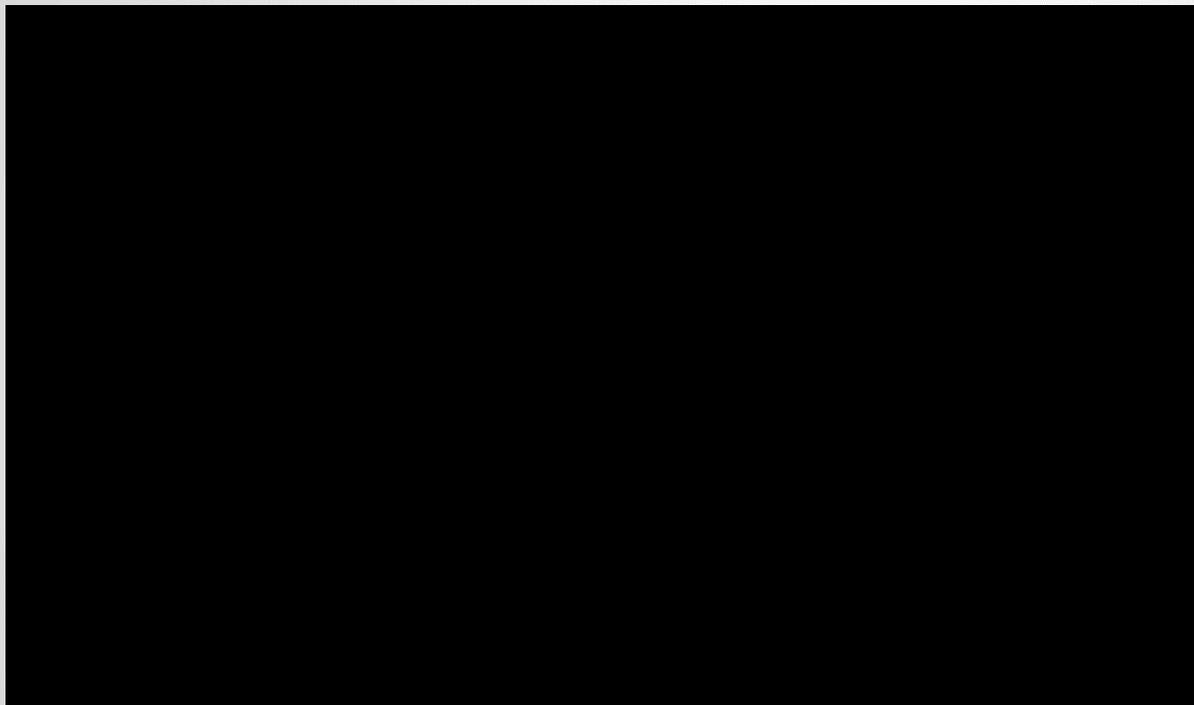
In [2]: # 设置超参数
learning_rate = 0.01
training_epochs = 1000
display_step = 50

In [3]: # 设置训练数据
train_X = numpy.asarray([0.3, 4.4, 5.5, 6.71, 6.93, 4.168, 9.779, 6.182, 7.59, 2.167,
7.042, 10.791, 5.313, 7.897, 5.654, 9.27, 3.1])
train_Y = numpy.asarray([1.7, 2.76, 2.09, 3.19, 1.694, 1.573, 3.366, 2.506, 2.53, 1.221,
2.827, 3.465, 1.65, 2.904, 2.42, 2.94, 1.3])
n_samples = train_X.shape[0]

In [4]: # tf 图输入
X = tf.placeholder("float")
Y = tf.placeholder("float")

# 设置模型权重
W = tf.Variable(rng.randn(), name="weight")
b = tf.Variable(rng.randn(), name="bias")
```

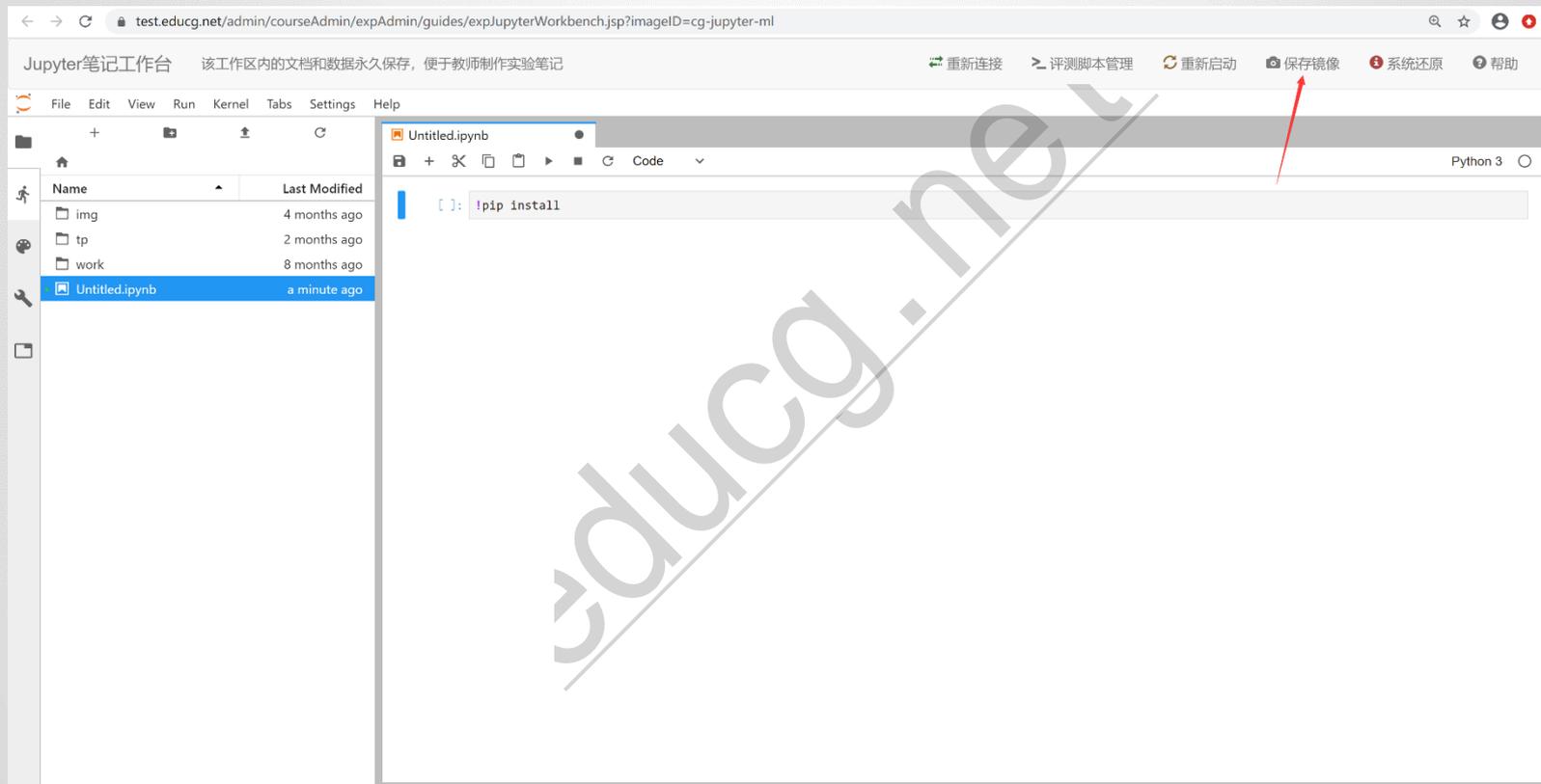
- ✓ 所见即所得
- ✓ 无缝呈现手册和代码
- ✓ 可直接调试和运行代码
- ✓ 支持图像，音频，视频
- ✓ 支持图片，图形，动画
- ✓ 支持Latex，HTML5，JS
- ✓ ....



- ✓ 所见即所得
- ✓ 无缝呈现手册和代码
- ✓ 可直接调试和运行代码
- ✓ 支持图像, 音频, 视频
- ✓ 支持图片, 图形, 动画
- ✓ 支持Latex, HTML5, JS
- ✓ ....

基于B/S架构, 真正实现“任何人、任何时间、任何地点”开展实验

- 一键制作Jupyter实验环境



基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## • 一键添加Jupyter实验环境



The screenshot shows the '在线实验' (Online Experiment) section of the '希冀' (Xiyi) platform. The main content area displays a table of saved Jupyter images. A red box highlights the '加入公共实验镜像库' (Add to Public Experiment Image Library) button for the first image.

**打开Jupyter工作台**

Jupyter工作台 / 在Jupyter工作台可以制作实验环境或者进行教学演示。 / 镜像加入公共镜像库之后, 才能被用作实验环境, 并注意提醒管理员部署该镜像。

#	保存的镜像	镜像描述	Jupyter镜像	加入公共实验镜像库	加入科研平台镜像库	加入通用评测环境	删除	编辑
1		From <code>jupyter/signals-system:1.0.1</code>   所在主机 <code>jupyterhos</code> t1   Docker启动选项 <code>-e JUPYTER_ENABLE_LAB=yes --user root -e GRANT_SUDO=yes --device /dev/snd</code>   By 黄莹   By 冯国荣   includes libraries for data analysis from the Julia, Python, and R communities.	<code>cg/jupyter/sigsys:v1.0.1</code> 6.02 GB <a href="#">查看日志</a>	<a href="#">已加入</a> <a href="#">查看日志</a>		<a href="#">加入</a>		

基于B/S架构, 真正实现“任何人、任何时间、任何地点”开展实验

- 添加Jupyter实验



test.educg.net/admin/index.jsp

希冀 系统管理 课程管理 CG Online Judge 科研平台 工程教育认证 GitLab 帮助

公告与问卷 课程信息 学生 作业 考试 在线实验 在线课堂 答疑 汇总成绩 助教 设置

实验资源库  
实验概览  
添加实验  
回收站

实验作业管理  
实验作业概览  
布置实验作业  
回收站  
计算资源管理

云桌面工作台<sup>trial</sup>  
远程桌面工作台  
Jupyter工作台

数据统计  
在线实验活动记录  
在线实验明细数据  
常见问题

实验概览 / 编辑Jupyter实验

Jupyter实验环境 / 填写实验内容

实验名称 实验1: 监督式学习算法基础

实验介绍 实验1: 监督式学习算法基础

封面图片 使用图片是为了呈现出更好的展现效果。建议尺寸: 280 × 157



实验笔记 点击上传Jupyter笔记 ...

已经上传的实验笔记:

- ch02-v3.ipynb
- table\_5\_1.jpg

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## • 添加Jupyter实验

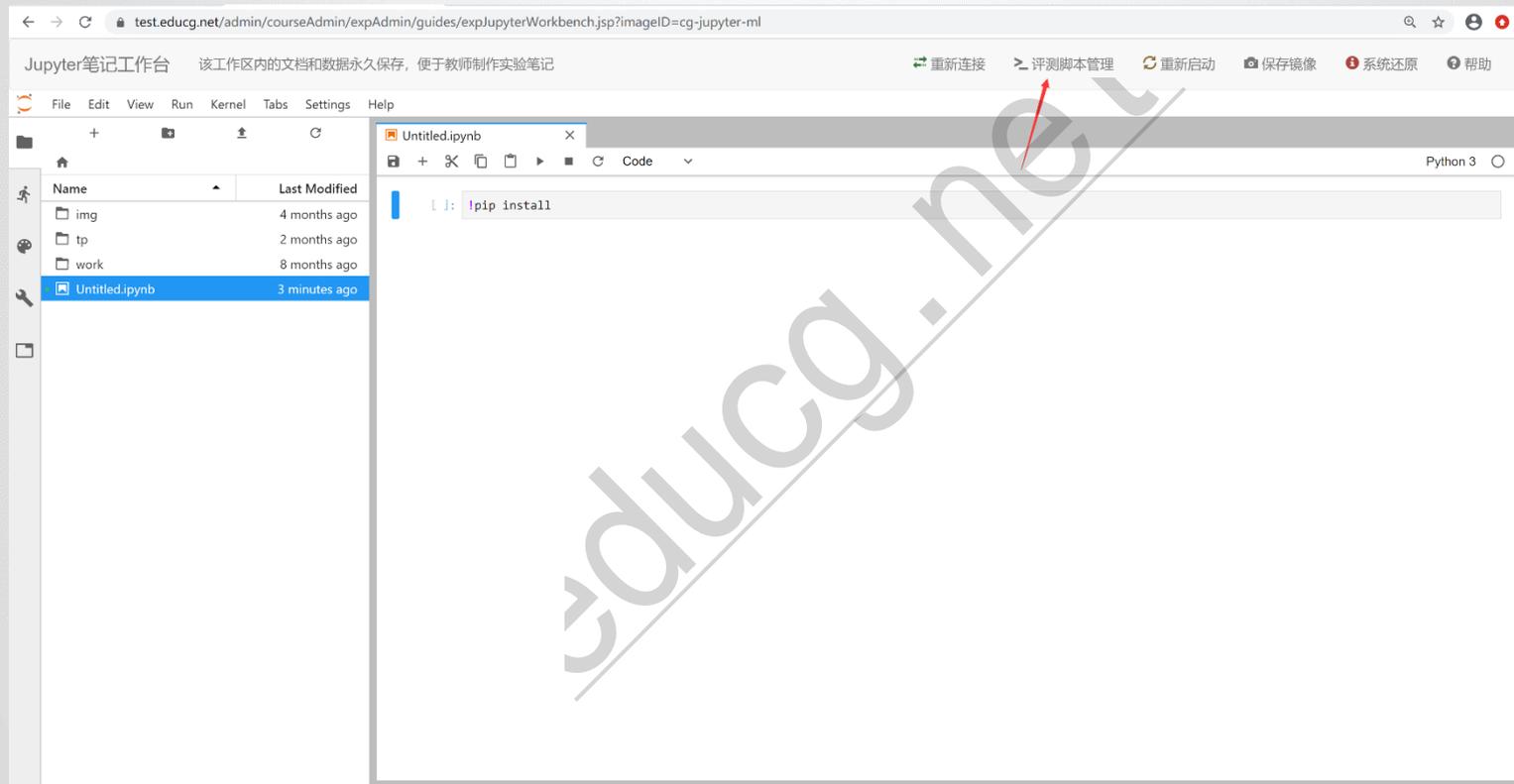


The screenshot shows the administration interface for adding a Jupyter experiment. The page is titled "test.educg.net/admin/index.jsp" and features a navigation bar with various system management options. The main content area is divided into several sections:

- 实验数据 (Experiment Data):** Includes a button to upload experiment data and a note that data is not automatically copied to the environment.
- 实验环境 (Experiment Environment):** A dropdown menu showing the current environment as "cg/jupyter/ml:v1.0.2".
- 实验界面 (Experiment Interface):** Radio buttons for selecting the interface type: "经典Jupyter笔记界面" (selected), "JupyterLab(IDE)界面", and "Shell终端界面". A link for "三类界面的区别" is provided.
- 内存上限 (Memory Limit):** A text input field set to "4096" MB, with a note explaining that this limit determines the number of Docker containers that can run on the host.
- 实验编号 (Experiment ID):** A text input field containing "25576317596155", with a note that it is a unique identifier for the experiment.
- 设置标签 (Setting Tags):** A grid of tags for categorizing the experiment, including "KNN", "决策树", "朴素贝叶斯", "逻辑回归", "SVM", "AdaBoost", "回归", "聚类", "K-Means", "关联分析", "Apriori", "FP-Growth", "PCA", "数据降维", "SVD", "推荐系统", "协同过滤", "特征工程", "机器学习实战", "机器学习理论", "机器学习验证+练习", "weka机器学习", "统计学习方法", and "机器学习算法导论".
- 参考答案 (Reference Answer):** Includes a button to upload the reference answer and a note that it can be published for students to view.

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

- 添加评测脚本



基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

## • 添加评测脚本



The screenshot shows the 'Add Evaluation Script' page in the CG Online Judge system. The page is divided into a left sidebar and a main content area. The sidebar contains navigation links for '公告与问卷', '实验资源库', '实验作业管理', '桌面资源管理', '云桌面工作台', and '数据统计'. The main content area has a top navigation bar with '系统管理', '课程管理', 'CG Online Judge', '科研平台', '工程教育认证', and 'GitLab'. Below this is a secondary navigation bar with '课程信息', '学生', '作业', '考试', '在线实验', '在线课堂', '答疑', '汇总成绩', '助教', and '设置'. The main content area is titled '添加评测脚本' and contains several form fields and sections:

- 内存上限:** A text input field with the value '4096' and a unit 'MB' dropdown. A tooltip explains that this is the memory limit for a Docker instance, and exceeding it will cause failure.
- 实验编号:** A text input field with the value '25576317596155'. A tooltip explains that this is a custom ID for experiment sorting, or the system will generate a unique one.
- 设置标签:** A grid of tags for machine learning topics, including KNN, 决策树, 朴素贝叶斯, 逻辑回归, SVM, AdaBoost, 回归, 聚类, K-Means, 关联分析, Apriori, FP-Growth, PCA, 数据降维, SVD, 推荐系统, 协同过滤, 特征工程, 专题: 机器学习实战, 类型: 练习, 类型: 验证, 线性回归, 多分类, 神经网络, 异常检测, 专题: sklearn机器学习, 专题: 机器学习理论, 类型: 验证+练习, 专题: weka机器学习, 专题: 统计学习方法, and 专题: 机器学习算法导论.
- 添加新标签:** A button to add new tags.
- 参考答案:** A section with a button '点击上传参考答案...' and a tooltip explaining that it is an optional field for publishing answers.
- 评测脚本:** A section with a dropdown menu showing 'judge.sh / 2020-05-12 15:34:44', a '评测超时' field with the value '5', and a unit '秒' dropdown. A tooltip explains that this is the evaluation script used for testing in the Jupyter workspace, and exceeding the time limit will force the process to end.
- 高级:** A button to expand advanced options.
- 保存:** A button to save the configuration.

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

- 添加评测脚本



☰ 课程信息   学生   作业   考试   在线实验   在线课堂   答疑   汇总成绩   助教   设置

### 设置评测脚本

选择评测脚本

judge.sh / 2020-05-12 15:34:44

该步骤的分数占比

100 %

该步骤得分占题目总分比例，取值范围(0, 100]。

评测超时时间

5 秒

评测超过该时间，将会强制结束。0秒不限时。

确定

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验



## 课程管理功能

- ◉作业
- ◉考试
- ◉答疑
- ◉成绩
- ◉公告
- ◉实验

## 作业

作业名称	类型	状态
4965 无向图遍历	选择题	未开始
4963 拼写检查 (综合作业—小数据集)	编程题	未开始
4961 图的最短路径	选择题	未开始
4960 最小生成树问题	选择题	未开始
4979 图的邻接矩阵表示	填空题	未开始
4976 二叉排序树的查找	填空题	未开始
4977 哈夫曼树的结点个	选择题	未开始
4976 二叉树的遍历	填空题	未开始
4975 完全二叉树的结点个	选择题	未开始

## 在线实验 (桌面 + Jupyter)

- 基于CNN的车牌识别系统** (License Plate Recognition System based on CNN)
- 卷积神经网络** (Convolutional Neural Networks)
- 神经网络模型原理** (Principles of Neural Network Models)
- Pandas数据分析** (Data Analysis with Pandas)
- Matplotlib绘图与可视化** (Matplotlib Plotting and Visualization)
- 自己动手做聊天机器人 (信息检索版)** (Build your own chatbot - Information Retrieval Edition)

## 在线考试

题目	类型	难度
3227 年龄排序	数组与指针 编程题	2
3226 无向图遍历	图 选择题	
3225 快速排序	排序算法 选择题	4
3224 最小元素位置	排序算法 选择题	
3223 后缀表达式计算	栈与队 编程题	3

## 成绩汇总

#	汇总成绩单	班级	成绩查询开关	操作
1	2018级 数据结构与程序...	钱红兵老师班	关闭	重新汇总   编辑权限   删除
2	2018级 数据结构与程序...	尚涛老师班	关闭	重新汇总   编辑权限   删除
3	2018级 数据结构与程序...	林广艳老师班	关闭	重新汇总   编辑权限   删除
4	2018级 数据结构与程序...	谭义彬老师班	关闭	重新汇总   编辑权限   删除
5	2018级 数据结构与程序...	潘清华老师班	关闭	重新汇总   编辑权限   删除
6	2018级 数据结构与程序...	刘开亮老师班	关闭	重新汇总   编辑权限   删除



01

---

**在线作业**

## 手工选题

自己掌控。

## 自动出题

系统帮忙布置作业。

## 随机作业

为学生随机分配题目。

## 小组作业

学生自由分组、互评。

### 基本信息

作业简述：  
2017年编译技术课程设计

作业要求：  
注意：作业重新打开，是因为有的同学没有按时提交，已经提交的同学请不要再次提交，否则将按“补交”算成绩。从2018年1月14日17:00开始算，在1个小时内补交扣1分，2个小时内补交扣2分，以此类推。

提交作业之前请再次确认最终成果是否按照如下要求组织的：

1. 最终成果放入三个文件夹“源代码”、“文档”、“测试程序”

### 属性信息

将作业布置给：  
实践课

作业开始时间：  
2017-11-14 00:00

作业终止时间：  
2018-01-14 17:00  设置补交截止时间 2018-01-31 12:00

### 抽题规则

随机抽题规则：  
基于 难度 规则，抽取 1 道题目(共有8道题目)，最多抽取 3 次  
每种难度最多抽取 2 次

#### 随机规则说明：

- “全局随机”：从题目列表中，随机抽选若干道题目。
- “难度”：从某种难度的题目中（假设总数为 $x$ ），随机抽选若干道题目(假设为 $y$ )，最终抽取的题目数量为： $Min(x, y)$

还可以补交

## • 丰富的题目类型

- 选择题
- 填空题
- 判断题
- **简答题**
- 文件上传题
- 编程题
- 接口编程题
- 程序片段编程题
- 算法可视化
- 并行编程题
  - MPI分布式
  - 多线程
- 项目题
- 通用评测题

5类  
通用  
题型

4类  
编程  
题型

1类  
并行编程  
题型

简答题在线评阅 / 14级第六次作业 / 什么是软件工程

学号	姓名	作业
11061116	郑广智	14级第六次作业

**简答题辅助判分**

学生答案 与参考答案的相似度为 **91.93%**，参考分数：**4.60**

您可以在下面的编辑器内直接批阅学生提交的答案，作业过期之后，评阅信息将会显示在学生端。

参考答案

软件工程是指导计算机软件开发和维护的工程学科。用分阶段的生命周期计划管理、坚持阶段评审、实行严格产品控制、采用现代设计技术、结果审查、开发小组人员少而精、并不断改进方法。

HTML 

自定义标题 段落格式 字体 字号

代码语言

元素路径:

系统将自动将学生答案与参考答案进行相似性比较，利用相似度辅助判分

## 丰富的题目类型

- 选择题
- 填空题
- 判断题
- **简答题**
- 文件上传题
- 编程题
- 接口编程题
- 程序片段编程题
- 算法可视化
- 并行编程题
  - MPI分布式
  - 多线程
- 项目题
- 通用评测题

5类  
通用  
题型

4类  
编程  
题型

1类  
并行编程  
题型



编译技术

课程信息 学生 作业 考试 设置 答疑 助教 汇总成绩 增设栏目 进入学生端

1. 最终成果放入三个文件夹“源代码”、“文档”、“测试程序”  
2. 将上述三个文件夹放入一个文件夹,并将该文件夹命名为: 学号\_姓名

属性信息

将作业布置给: 1406级

作业开始时间: 2016-11-03 00:00

作业终止时间: 2017-01-01 10:10  设置补交截止时间

抽题规则

随机抽题规则: 基于 [难度] 规则, 抽取 1 道题目(共有7道题目), 最多抽取 3 次  
每种难度最多抽取 2 次

随机规则说明:

- “全局随机”: 从题目列表中, 随机抽取若干道题目。
- “难度”: 从某种难度的题目中 (假设总数为 $x$ ), 随机抽取若干道题目(假设为 $y$ ), 最终抽取的题目数量为:  $Min(x, y)$

题目列表

文件上传题

题目描述	相似性比较	分值	移动	删除
1414. 扩充C0文法-中数组	<input checked="" type="checkbox"/> 文档相似性 文档类型: Word, PDF, Excel, PowerPoint <input checked="" type="checkbox"/> 源代码相似性 源代码语言: Java, C/C++, C#, Python	0.0	移动	删除
1413. 扩充C0文法-高-无类型	<input checked="" type="checkbox"/> 文档相似性 文档类型: Word, PDF, Excel <input checked="" type="checkbox"/> 源代码相似性 源代码语言: Java, C/C++, C#	0.0	移动	删除

文档 / 源码相似性比较

文件上传题：可以自动从压缩包内提取**文档**或者**源代码**，进行**相似性比较**。

## • 代码查重

- 赵长海等，基于编译优化和反汇编的程序相似性检测方法，《北京航空航天大学学报》，2008年。国内相似性比较研究领域，引用率最高的文章之一。



若存在任何手段能够绕开检测算法，那么查重功能形同虚设！



02

---

在线考试

## • 全方位支持在线考试：自动组卷

- 帮助教师组卷，规则包括：知识点分布、章节、难度、使用频度



The screenshot displays the '在线考试' (Online Exam) configuration page. The main content area is titled '自动组卷' (Automatic Question Pooling) and includes several configuration options:

- 题目数量** (Number of Questions): A text input field.
- 知识点分布** (Knowledge Point Distribution): A list of knowledge points including '控制流', '运算符与表达式', '结构', and '文件'. Below the list, it says '若要多选或取消选择, 按住ctrl, 点击知识点' (To select multiple or cancel selection, hold Ctrl and click the knowledge point).
- 章节分布** (Chapter Distribution): A list of chapters including '第1章', '第2章', '第3章', and '第4章'. Below the list, it says '若要多选或取消选择, 按住ctrl, 点击章节' (To select multiple or cancel selection, hold Ctrl and click the chapter).
- 难度分布** (Difficulty Distribution): Radio buttons for '均匀分布' (Uniform Distribution), '随机分布' (Random Distribution), and '手工指定' (Manual Specification). Below, it says '如果没有为题目设置难度, 注意选择“随机分布”' (If no difficulty is set for the question, please select "Random Distribution").
- 使用频度** (Usage Frequency): Radio buttons for '最少被使用' (Least Used), '经常被使用' (Used Often), and '不设定' (Not Set).

Navigation buttons at the top of the configuration area include: '← 上一步【选择出题方式】' (Previous Step), '[设置组卷规则]' (Set Question Pooling Rules), and '下一步【填写试卷属性】 →' (Next Step).

On the left sidebar, there are sections for '考试题库' (Exam Question Bank), '试卷管理' (Exam Paper Management), and '考试管理' (Exam Management).

A large red button labeled '自动组卷' (Automatic Question Pooling) is overlaid on the right side of the interface.

- 全方位支持**在线考试**：**学生端随机抽题组卷**
  - 随机抽题组卷：题型分布、知识点分布、难度分布、题目数量

## 组卷考试

- 第一步，从试卷库内选择试卷，一场考试可以使用任意多套试卷。
- 第二步，然后进入考试设置：
  - 设置考试时间、参加考试的班级、选择试卷分配算法（根据学号或者IP地址分发试卷）等。
  - 配置监考选项，包括内网监控、IP地址绑定、IP地址过滤和栏目关闭。这些选项只对参与考试的班级有效，考试结束，自动失效。

## 随机组卷考试

## 学生端随机组卷

这种考试方式不需要事先组卷。

- 第一步，设置抽题规则和题库范围，考试时，根据规则为每一位考生随机组织一套试卷。
- 第二步，然后进入考试设置：
  - 设置考试时间、参加考试的班级、选择试卷分配算法（根据学号或者IP地址分发试卷）等。
  - 配置监考选项，包括内网监控、IP地址绑定、IP地址过滤和栏目关闭。这些选项只对参与考试的班级有效，考试结束，自动失效。

编程题

题目数量

题目分布  精细设置  简洁设置

数量	类	难度
2	类	难度：随机分布
2	继承和派生	平均分布
2	算法设计与分析	随机分布
0	虚函数与多态性	
0	性能综合	1 1
0	综合	2 1
0	函数	3 0
0	数组与指针	4 0
0	结构	5 0
0	文件	手工指定
0	运算符与表达式	难度：随机分布
0	控制流	难度：随机分布

填写每个知识点的题目数量

题目分值	难度 1	难度 2	难度 3	难度 4	难度 5
	<input type="text" value="3"/>	<input type="text" value="4"/>	<input type="text" value="5"/>	<input type="text" value="8"/>	<input type="text" value="10"/>

设置各类难度系数的题目分值

[试试看](#)

- 全方位支持**在线考试**：**在线监考**
  - 能够识别**NAT网络**和**反向代理网络**下的，**客户端真实IP地址**



课程公告 | 课程信息 | 学生 | 作业 | 考试 | 实验 | 答疑 | 汇总成绩 | 在线教程 | 助教 | 设置 | 进入学生端

返回考试概览 / 修改考试属性

**基本信息**

考试简述: 2016级“数据结构与程序设计基础”期末考试（计算机学院）

**属性设置**

试卷分发策略: 根据IP地址求余分发试卷 【提示：若选择根据学号分发试卷，每个学生得到的试卷是根据学号的最后一个数字与试卷数量求余得到，请务必注意学生的座次】

此次考试由: 计算机学院1606 参与

考试时间: 2017-06-18 从 13 点 28 分开始 持续 189 分钟

**答卷设置**

编程题评判结果:  只显示翻译信息  显示所有信息【包括运行时信息，测试数据和得分】

考试辅助功能:  显示服务器时间  考试结束前 5 分钟提醒【考试临近结束，在页面显著位置倒计时提醒】  显示提前交卷按钮【提示：学生提前交卷后，无法再进入考试栏目；如遇特殊情况，“在线监考”内可以解除绑定】

**监考设置**

内网IP监控:  启用内网IP监控【提示：若内网的计算机以NAT方式访问CG服务器(点击此处自动检测)，请勾选此项。IE浏览器不支持内网IP获取，要求学生必须使用Chrome 37+，Firefox 25+，或者360浏览器(极速模式)等】

考试IP绑定:  启用考试期间IP绑定【提示：启用IP绑定后，考试期间只允许学生使用唯一的IP登录。如果NAT方式访问CG服务器，注意勾选“启用内网IP监控”。如遇特殊情况，“在线监考”内可以解除绑定】

IP地址过滤: 允许访问的IP地址，正则表达式表示  正则表达式编辑器 【提示：使用正则表达式描述允许访问本场考试的IP地址范围，如果范围之外的IP访问考试，“在线监考”能够看到报警信息】

关闭栏目: 考前 0 分钟关闭:  课程信息  在线作业  在线答疑

**试卷列表**

- 2016级“数据结构与程序设计基础”期末考试（计算机学院）（B卷）
- 2016级“数据结构与程序设计基础”期末考试（计算机学院）（A卷）

可监控内网IP  
(机房通过NAT上网)

支持IP绑定  
(一个账号对应一台机器)

IP地址访问控制  
(只允许特定的IP地址访问考试)

栏目关闭  
(关闭答疑论坛、作业等栏目)

试卷分发  
规则

考试辅助功能

多套试卷

- 全方位支持**在线考试**：**在线监考**
- 实时监测学生交换账号互相抄袭、异地替考



The screenshot shows a web-based exam monitoring system. On the left is a sidebar with navigation options like '试题库' (Question Bank) and '试卷管理' (Exam Paper Management). The main area displays '在线监考：2017级 数据结构与程序设计 期末考试（北京学院）'. It includes a '登录日志' (Login Log) table with columns for time, user name, and IP address. Below the log is an '异常信息' (Exception Information) section. On the right, there is an 'IP地址绑定列表' (IP Address Binding List) table with columns for user name and IP address. At the bottom right, there is a '提前交卷的学生列表' (Students who submitted early) section with a '解锁' (Unlock) button.

时间	用户名	IP地址
2018-06-24 13:28:11	侯环峻(16182628)	10.5.4.23
2018-06-24 13:28:10	包尔尔(16182632)	10.5.4.25
2018-06-24 13:28:09	王镇岳(16182633)	10.5.4.27
2018-06-24 13:28:11	吕天欣(16182642)	10.5.4.29
2018-06-24 13:28:22	张慧慧(16182646)	10.5.4.31
2018-06-24 13:28:12	贾拓(16182649)	10.5.4.14
2018-06-24 13:28:08	杨立军(16182662)	10.5.4.16
2018-06-24 13:28:16	元鑫(16182663)	10.5.4.18
2018-06-24 13:28:09	高立彪(16182664)	10.5.4.20
2018-06-24 13:28:07	安彪(16182682)	10.5.4.33
2018-06-24 13:28:02	孙自安(sza18)	10.5.4.80
2018-06-24 13:28:10	张子斌(17182620)	10.5.4.60

用户名	IP地址
张晨(15182605)	
白惠娜(15182644)	
齐翔宇(15182684)	
赵睿骑(16182106)	
孙祎彤(16182603)	
王聪昊(16182605)	
崔臻(16182607)	
时琪(16182609)	
宋博文(16182610)	
马俊楠(16182615)	
马帅(16182616)	

交换账号

异地登录

访问控制

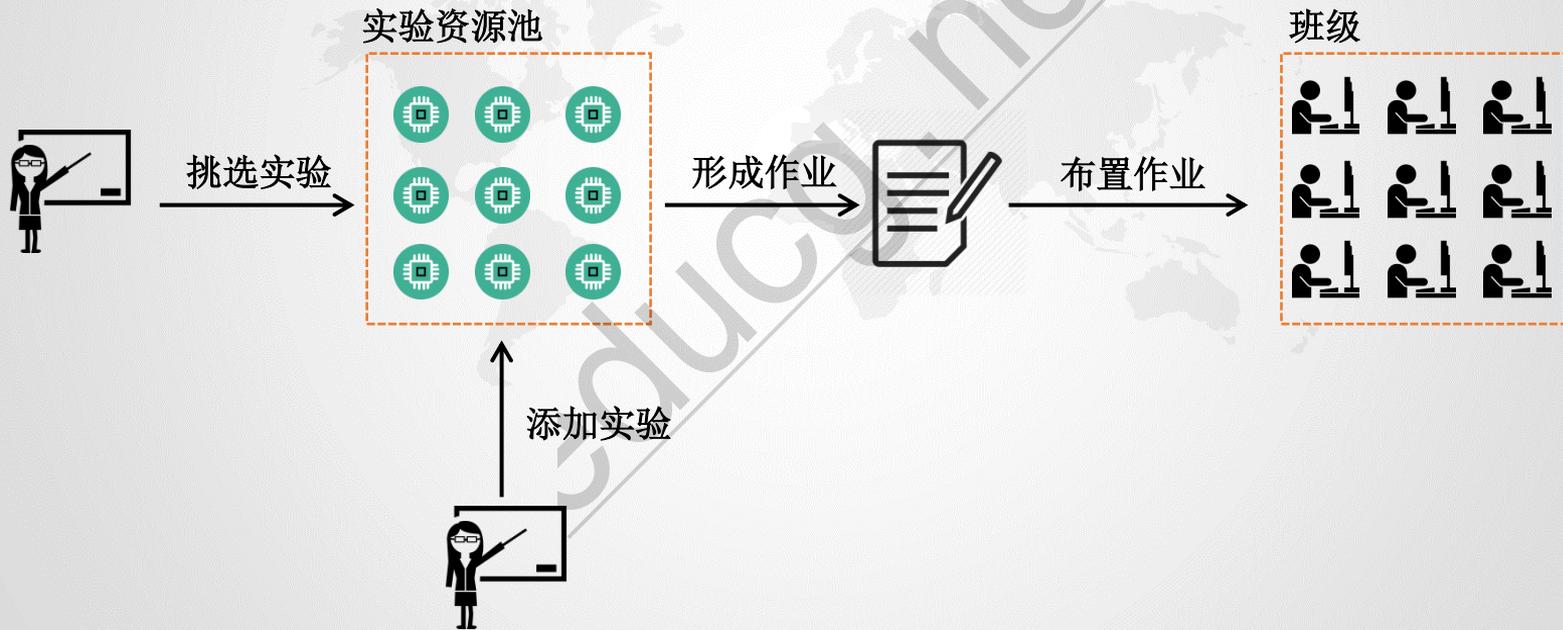
IP绑定

03

---

在线实验

## 符合教学过程：教师端视图

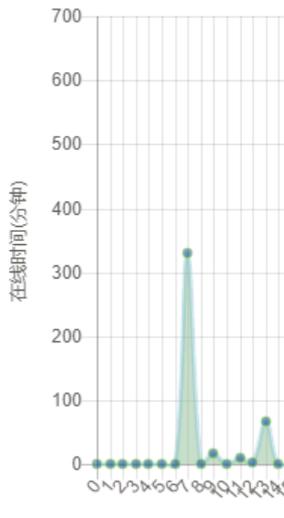
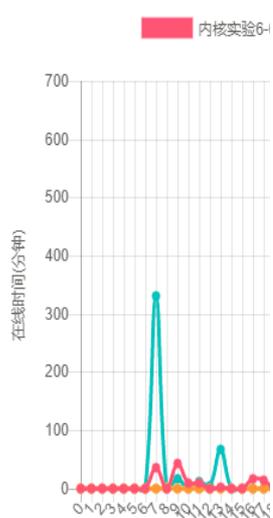


## 符合教学过程：学生端视图



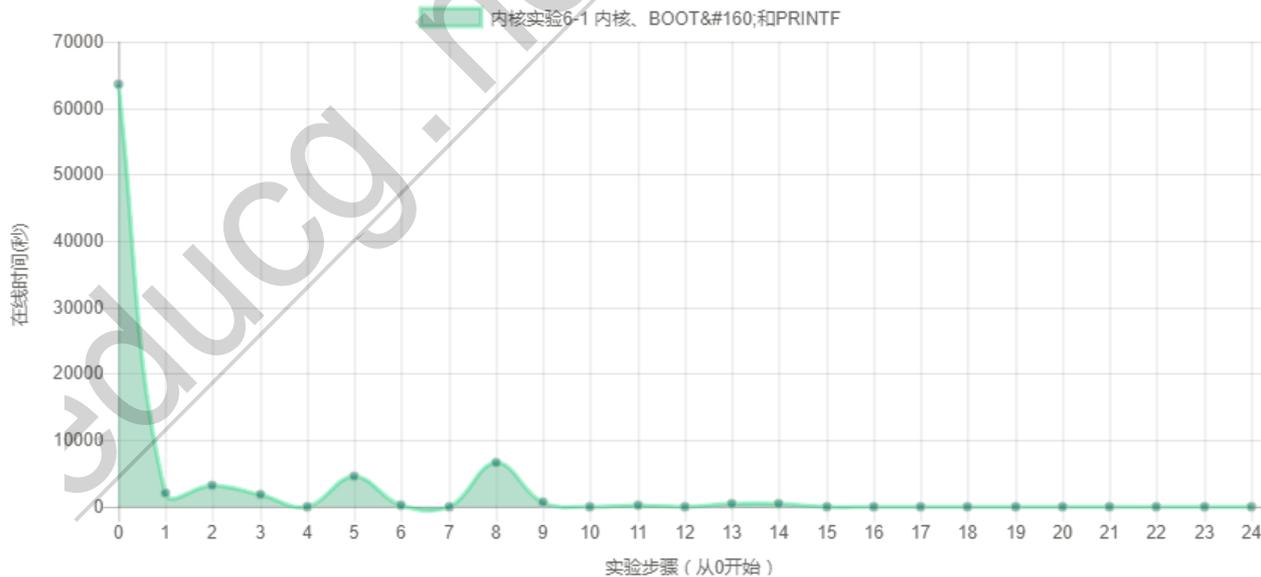
## • 实验作业评阅：追踪实验过程信息

在线时间统计



个人信息 注销

实现步骤在线时间分布





四

---

**专业支撑方案**



01

---

**计算机专业**

# 程序设计

不仅仅告知正确与否

- ◎动态测试
- ◎代码风格
- ◎静态分析
- ◎性能剖析
- ◎并发评判
- ◎代码查重
- ◎输入输出任意组合
- ◎项目级源代码评测

## • 程序评判：不仅仅对与错

### 基本测评报告

提交源文件:

注意: 只能用 C 语言编写程序。如果有多个源文件, 可以压缩成zip包提交。

未选择任何文件

编译选项: 编译选项

```
example1c.c: In function 'main':
example1c.c:7:35: warning: unused variable 'z' [-Wunused-variable]
example1c.c:7:29: warning: unused variable 'p' [-Wunused-variable]
/tmp/ccaOFvdA.o: In function 'main':
example1c.c:(.text.startup+0x59): warning: the 'gets' function is dangerous and should not be used.
```

共有测试数据: 5

平均占用内存: 441.480K 平均CPU时间: 0.614S 平均墙钟时间: 0.006S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	运行时错误(SIGFPE)
测试数据5	完全正确

[详细评判结果 >>>](#)

测试数据中包括了正常情况和边界等异常情况, 使学生体会到**工程**编码要求

- 程序评判：不仅仅对与错
  - 错误分析，协助学生调试程序，降低初学者挫折感

动态测试 代码风格检查 性能剖析 代码静态分析

动态测试

成功通过编译, 且无编译警告

测试数据	测试结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	输出错误
测试数据5	运行时错误(SIGFPE)

错误原因可能是:  
程序输出过多数据, 有可能存在死循环

动态测试 代码风格检查 性能剖析 代码静态分析

代码静态分析

共发现 4个问题.

- [find.c:18]: (style) The scope of the variable 'mid' can be reduced.
- [find.c:16]: (error) Array 'node[i].nu[12]' accessed at index 12, which is out of bounds.
- [find.c:12]: (portability) scanf without field width limits can crash with huge input data on some versions of libc.
- [find.c:15]: (warning) scanf without field width limits can crash with huge input data.

(information) Cppcheck cannot find all the include files (use --check-config for details)

动态测试 代码风格检查 性能剖析 代码静态分析

代码风格检查

共发现 104个问题.

print.c(1, 0): Please provide file info comment in front of file. It includes license/copyright information along with filename, author, date of modification, version and a brief description [RULE\_8\_1\_A\_provide\_file\_info\_comment]

print.c(3, 5): Do not start function name(main) with lowercase [RULE\_3\_3\_A\_start\_function\_name\_with\_uppercase\_windows]

print.c(5, 21): Provide spaces after operator ';' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 7): Provide spaces after operator ';' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 9): Provide spaces b/w operator '=' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 11): Provide spaces after operator ';' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 13): Provide spaces b/w operator '=' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 15): Provide spaces after operator ';' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 17): Provide spaces b/w operator '=' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 19): Provide spaces after operator ';' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 21): Provide spaces b/w operator '=' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 23): Provide spaces after operator ';' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 25): Provide spaces b/w operator '=' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 27): Provide spaces after operator ';' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(6, 30): Provide spaces b/w operator '=' [RULE\_4\_2\_A\_A\_space\_around\_operator]

print.c(7, 4): Provide spaces b/w operator '=' [RULE\_4\_2\_A\_A\_space\_around\_operator]

动态测试 代码风格检查 性能剖析 代码静态分析

执行效率

程序中语句平均执行次数: 52531

行号	执行次数	源码
1	:	#include <stdio.h>
2	:	#include <string.h>
3	:	#include <stdlib.h>
4	:	#define NHASH 3001
5	:	#define MULT 37
6	:	:
7	:	struct word
8	:	{
9	:	char s[25];
10	:	struct word *next;
11	:	};
12	:	:
13	:	char dict[3600][25], s[128];
14	:	struct word *h[3600], *p,*q;
15	:	int num=1,b=1,ind[26];
16	:	:
17	30314	unsigned int hash(char *str)
18	:	{
19	30314	unsigned int h=0;
20	:	char *p;
21	30314	for(p=str; *p!='\0'; p++) h = MULT*h + *p;
22	30314	return h % NHASH;
23	:	}

帮助学生全面认知程序的内涵，树立软件工程师思维，引导学生写出高质量的程序

## • 程序评判：支持多源文件自动编译与评判

- 复杂问题的源文件组织也是一种工程训练。特别是Java代码，一般一个类对应一个文件，若使用文本框就要破坏Java的编程规范。

### 【样例说明】

程序运行结果为按字典序输出拼写错误或字典中不存在的单词及出现位置到指定文件中，其中问题单词“lofe”在次。

### 【评分标准】

本题是一个综合性能测试题，其评分标准为在所有程序中运行最快的将得满分，其它程序的得分以最快的程序运行根据其运行时间计算得出。程序运行无结果或结果错误将不得分。

提交源文件:

选择编写程序使用的语言：。如果有多个源文件，压缩成 rar 或者 zip 包提交。

支持多源文件打包上传

未选择任何文件

得分4.4688272 最后一次提交时间:2017-05-30 10:45:53

共有测试数据:1

平均占用内存:257.703K 平均CPU时间:0.108S 平均墙钟时间:0.108S

测试数据

测试数据1

评判结果

完全正确

[性能排行榜 >>>](#) [详细评判结果 >>>](#)

接近真实工程环境的，  
源代码组织风格

“培养学生解决**复杂工程问题**的能力”  
—工程教育专业认证

- **程序评判：实时评判 (VS. 串行排队)**
  - 利用CPU多核并行评判学生程序，实时出结果

**上海大学实验：**  
300~400人同时上机，串行排队评判延迟能达到**40分钟**。

- 其它系统如何做？**
1. 提交次数多扣分
  2. 限制提交时间间隔



系统公告 | 教师账号与权限 | 课程设置 | 系统设置 | 系统维护 | 答疑论坛设置

### 程序自动评判进程池

该进程池负责编译和评测学生提交的源程序，评判进程的数目是服务器CPU核数的2倍。

进程池状态 (当前进程数: 32)

**评判进程池：CPU核数 × 2**

进程ID	占用CPU (%)	占用内存 (%)	已运行时间	运行状态
29816	0.0	0.0	2-10:17:19	Sleeping
29822	0.0	0.0	2-10:17:19	Sleeping
29829	0.0	0.0	2-10:17:19	Sleeping
29834	0.0	0.0	2-10:17:19	Sleeping
29839	0.0	0.0	2-10:17:19	Sleeping
29844	0.0	0.0	2-10:17:19	Sleeping
29849	0.0	0.0	2-10:17:19	Sleeping
29854	0.0	0.0	2-10:17:19	Sleeping
29859	0.0	0.0	2-10:17:19	Sleeping
29864	0.0	0.0	2-10:17:19	Sleeping
29869	0.0	0.0	2-10:17:19	Sleeping
29875	0.0	0.0	2-10:17:19	Sleeping
29884	0.0	0.0	2-10:17:19	Sleeping
29899	0.0	0.0	2-10:17:19	Sleeping
29906	0.0	0.0	2-10:17:19	Sleeping
29915	0.0	0.0	2-10:17:19	Sleeping
29920	0.0	0.0	2-10:17:19	Sleeping
29925	0.0	0.0	2-10:17:19	Sleeping
29930	0.0	0.0	2-10:17:19	Sleeping
29935	0.0	0.0	2-10:17:19	Sleeping
29940	0.0	0.0	2-10:17:19	Sleeping
29945	0.0	0.0	2-10:17:19	Sleeping

## • 性能

- 单台普通笔记本能够支撑**2000**人同时在线考试
- 单台双路服务器可支撑**50000**人同时在线考试
- 案例：
  - 上海大学
    - 单台服务器支撑全校3000人
  - 国防科大
    - 1GB内存台式机的抗压测试
  - 北航
    - P4 CPU 1GB内存台式机 350人考试
    - 大类招生，同时800人在线考试
  - 河北地质大学
    - Windows VMWare工作站虚拟机，2核Xeon CPU，2GB，600人同时在线考试

## • 题目度量：客观量化题目难度

- 度量指标：**平均代码行、平均完成时间、正确率**

## 北航计算机学院程序设计课程目标

对于一般问题，具备分析问题、解决问题的能力，在**40~60分钟**内编写并调试通过、能正确运行、一般在**40~50行有效代码**以内的程序；通过学习让学生能够了解程序设计思想以及初步接受到程序设计方法、技巧、风格的训练，并逐步了解工程化要求。

编辑	删除	样例	平均代码行	正确率	平均完成时间	类型	知识点	难度	章节	录入	修改
			109	0.81	87分钟	编程题	图	4		2016-03-17 16:58:24	2016-07-15 15:23:21

基于历年使用的  
历史数据统计

### 【问题描述】

编写一个程序实现北京地铁最短乘坐（站）线路查询，输入为起始站名和目的站名，输出为从起始站到目的站的最短乘坐站换乘线路。注：1. 要求采用Dijkstra算法实现；2) 本题在实际测试时对数据文件进行了调整，使得输入的两站间只有一条最短路径。

北京地铁线路图  
Beijing Subway Map



### 返回考试概览 / 试卷评阅：2016级“数据结构与程序设计基础”期中考试

学生视图

题目视图

2016级“数据结构与程序设计基础”期中考试 (B卷)

1439. 正文排版 (两端对齐)

2016级“数据结构与程序设计基础”期中考试 (A卷)

====按题目分类浏览=====

### 统计分析

最高分	最低分	平均分	标准差	正确率	平均代码行	平均完成时间(分钟)
10.0	0.0	3.86	3.71	76/133	53	54

### 功能操作

重新评判

学号

姓名

最后

基于单次考试/  
作业的统计

分数

修订得分

## • 代码查重

- 赵长海等，基于编译优化和反汇编的程序相似性检测方法，《北京航空航天大学学报》，2008年。国内相似性比较研究领域，引用率最高的文章之一。



若存在任何手段能够绕开检测算法，那么查重功能形同虚设！



## 代码相似性比较示例1:

```
VS - Google Chrome
@ judge.buaa.edu.cn/admin/courseAdmin/assignmentAdmin/compareFiles.jsp?assignID=447&problemID=2966&userID1=8465&userID2=8472

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct node{
    int data;
    struct node *link;
}LNode,*Linklist;
void JOSEPHUS(int n,int m,int q)
{
    Linklist p,r,list=NULL;
    int i;
    for(i=1;i<=n;i++){
        p=(Linklist)malloc(sizeof(LNode));
        p->data=i;
        if(list==NULL)
            list=p;
        else
            r->link=p;
        r=p;
    }
    p->link=list;
    此, 建立一个循环链表*/
    p=list;
    for(i=1;i<=q;i++){
        r=p;
        p=p->link;
    }
    此时指向第1个出发结点*/
    while(p->link!=p){
        for(i=1;i<=m;i++){
            r=p;
            p=p->link;
        }
        指向第m个结点, 指向第m-1个结点*/
        r->link=p->link;

        free(p);
        p=r->link;
    }
    printf("%d",p->data);
}
int main()
{
    int n,m,p;
    scanf("%d %d %d",&n,&m,&p);
    JOSEPHUS(n,m,p);
    return 0;
}

#include <stdio.h>
#include <stdlib.h>
typedef struct node{
    int num;
    struct node *link;
}LNode,*Linklist;
void JOSEPHUS();
int main()
{
    int i,j,k;
    scanf("%d%d%d",&i,&j,&k);
    JOSEPHUS(i,j,k);
}

void JOSEPHUS(int n,int m,int k){
    Linklist p,r,list=NULL;
    int i;
    for(i=1;i<=n;i++){
        p=(Linklist)malloc(sizeof(LNode));
        p->num=i;
        if(list==NULL)
            list=p;
        else
            r->link=p;
        r=p;
    }
    p->link=list;
    p=list;
    for(i=1;i<=k;i++){
        r=p;
        p=p->link;
    }
    while(p->link!=p){
        for(i=1;i<=m;i++){
            r=p;
            p=p->link;
        }
        r->link=p->link;
        free(p);
        p=r->link;
    }
    printf("%d\n",p->num);
}

/*申请一个新的链结点*/
/*
/*至
/*p
/*删除第m个结点*/
```

调整代码顺序 + 注释

## 代码相似性比较 示例2:

```
VS - Google Chrome
judge.buaa.edu.cn/admin/courseAdmin/assignmentAdmin/compareFiles.jsp?assignID=447&problemID=2949&userID1=8479&userID2=8408

main(int argc, char *argv[])
{
    int i = 0, j = 0;
    int k = 1;
    int n = 0;
    //char *scr = NULL;
    char *file = NULL;
    char *pt[100] = {NULL};
    FILE *pi = NULL;

    //scr = (char*)malloc(sizeof(char) * size);
    file = (char*)malloc(sizeof(char) * size);

    /*scanf("%s", scr);
    free(scr);
    scr = NULL;
    scanf("%s", file);
    if (*file++ == '-') {
        while(*file != '\0') {
            n = n*10+(*file-'0');
            file++;
        }
        scanf("%s", file);
    }
    else n = 10;*/
    if(argc == 2){
        n = 10;
        strcpy(file, argv[1]);
    }
    if(argc == 3){
        strcpy(file, argv[2]);
        while(*(argv[1]+k) != '\0') {
            n = n*10+(*(argv[1]+k)-'0');
            k++;
        }
    }

    if((p1=fopen(file, "r")) == NULL) {
        printf("cannot open the file!\n");
        exit(0);
    }
    else {
        pt[0] = (char*)malloc(sizeof(char) * size);
        while(fgets(pt[i], 100, p1) != NULL) {
            i++;
            pt[i] = (char*)malloc(sizeof(char) * size);
        }

        for(j = i-n; j<i; j++){
            puts(pt[j]);
        }
    }
}

/*主函数*/
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define x 60
main(int argc, char *argv[])
{
    int i=0, j=0, p=1, m=0;
    char *file = NULL;
    char *a[100] = {NULL};
    FILE *b = NULL;
    file = (char*)malloc(sizeof(char) * x);
    if(argc == 2){
        m = 10;
        strcpy(file, argv[1]);
    }
    if(argc == 3){
        strcpy(file, argv[2]);
        while(*(argv[1]+p) != '\0') {
            m = m*10+(*(argv[1]+p)-'0');
            p++;
        }
    }
    if((b=fopen(file, "r")) == NULL) {
        printf("cannot open the file!\n");
        exit(0);
    }
    else {
        a[0] = (char*)malloc(sizeof(char) * x);
        while(fgets(a[i], 100, b) != NULL) {
            i++;
            a[i] = (char*)malloc(sizeof(char) * x);
        }

        for(j = i-m; j<i; j++){
            puts(a[j]);
        }
    }

    return 0;
}
```

冗余代码 + 变量重命名 + 注释 + 排版

# 数据结构与算法

---

支持时间复杂度计算

◎算法可视化 ◎大数据性能评测

## 丰富的题目类型

5类  
通用  
题型

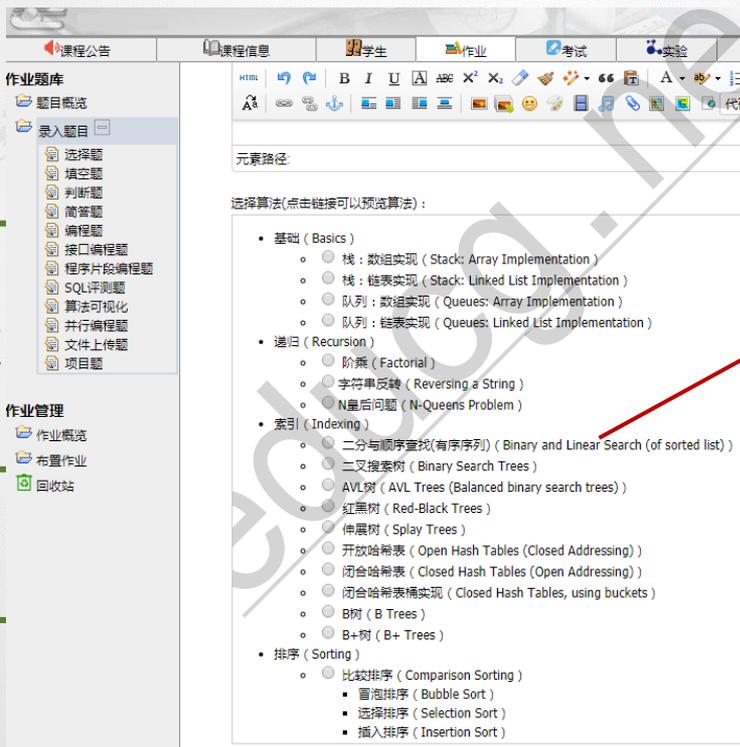
- 选择题
- 填空题
- 判断题
- 简答题
- 文件上传题

5类  
编程  
题型

- 编程题
- 接口编程题
- 程序片段编程题
- **算法可视化**
- SQL评测题

1类  
并行编程  
题型

- 并行编程题
  - MPI分布式
  - 多线程
- 项目题
- 通用编程题



课程公告 课程信息 学生 作业 考试 实验

作业题库

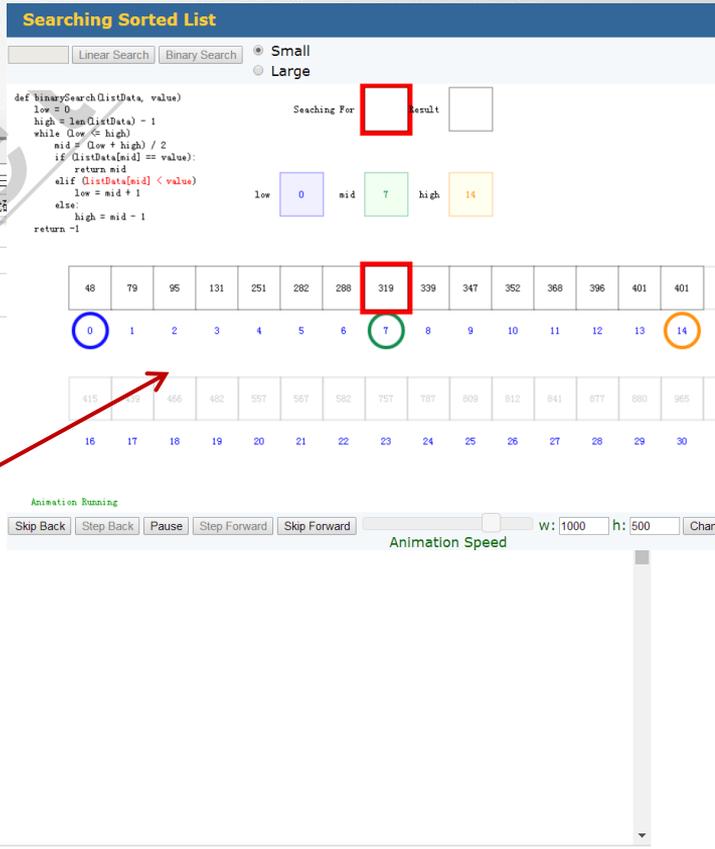
- 题目概览
- 录入题目
  - 选择题
  - 填空题
  - 判断题
  - 简答题
  - 编程题
  - 接口编程题
  - 程序片段编程题
  - SQL评测题
  - 算法可视化
  - 并行编程题
  - 文件上传题
  - 项目题

作业管理

- 作业概览
- 布置作业
- 回收站

选择算法(点击链接可以预览算法):

- 基础 (Basics)
  - 栈: 数组实现 (Stack: Array Implementation)
  - 栈: 链表实现 (Stack: Linked List Implementation)
  - 队列: 数组实现 (Queues: Array Implementation)
  - 队列: 链表实现 (Queues: Linked List Implementation)
- 递归 (Recursion)
  - 阶乘 (Factorial)
  - 字符串反转 (Reversing a String)
  - N皇后问题 (N-Queens Problem)
- 索引 (Indexing)
  - 二分与顺序查找(有序序列) (Binary and Linear Search (of sorted list))
  - 二叉搜索树 (Binary Search Trees)
  - AVL树 (AVL Trees (Balanced binary search trees))
  - 红黑树 (Red-Black Trees)
  - 伸展树 (Splay Trees)
  - 开放哈希表 (Open Hash Tables (Closed Addressing))
  - 闭哈希表 (Closed Hash Tables (Open Addressing))
  - 闭哈希表桶实现 (Closed Hash Tables, using buckets)
  - B树 (B Trees)
  - B+树 (B+ Trees)
- 排序 (Sorting)
  - 比较排序 (Comparison Sorting)
    - 冒泡排序 (Bubble Sort)
    - 选择排序 (Selection Sort)
    - 插入排序 (Insertion Sort)



Searching Sorted List

Linear Search Binary Search

Small Large

Searching For  result

```
def binarySearch(ListData, value)
low = 0
high = len(ListData) - 1
while low <= high:
mid = (low + high) / 2
if ListData[mid] == value:
return mid
elif ListData[mid] < value:
low = mid + 1
else:
high = mid - 1
return -1
```

low 0 mid 7 high 14

48	79	95	131	251	282	288	319	339	347	352	368	396	401	401
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

415 424 456 482 557 567 582 757 767 809 812 841 877 880 965

16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Animation Running

Skip Back Step Back Pause Step Forward Skip Forward

w: 1000 h: 500

Animation Speed

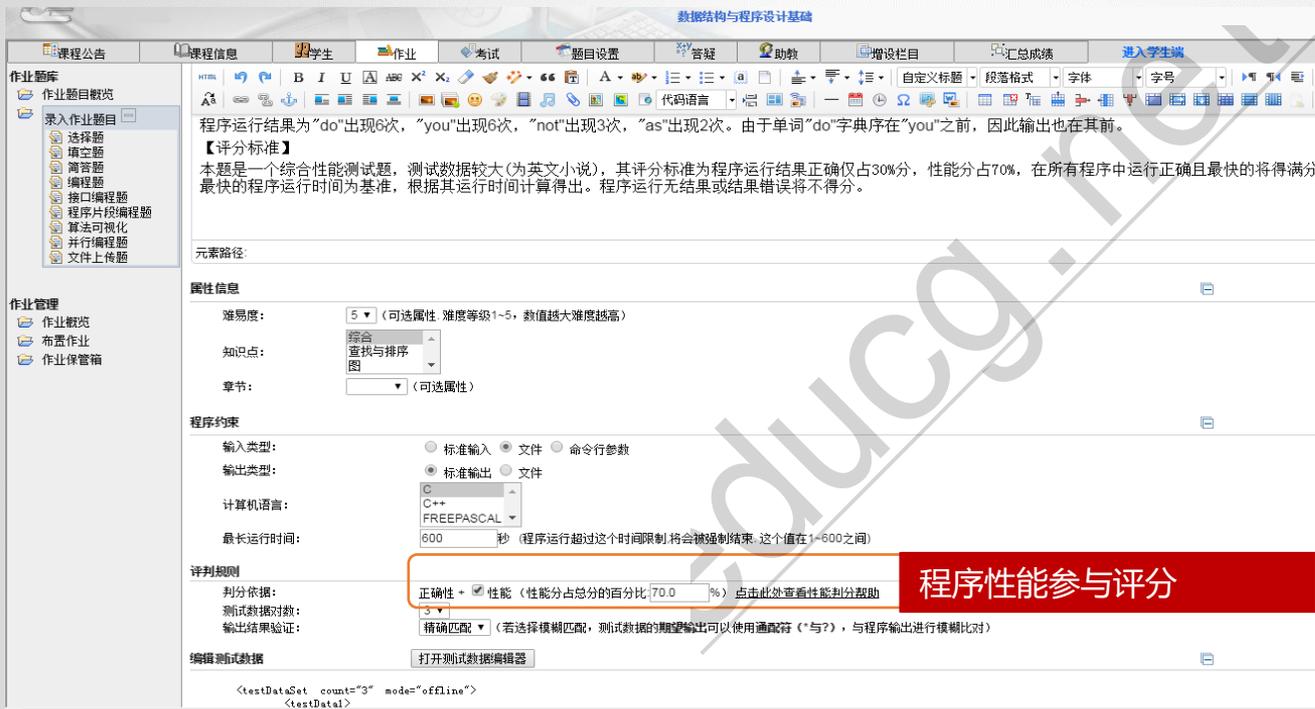
## • 程序评判：性能剖析！

度量每行代码的实际  
执行次数

```
动态测试  代码风格检查  性能剖析  代码静态分析

执行效率
程序中语句平均执行次数: 1081
行号  执行次数  源码
1      : #include <stdio.h>
2      : #include <stdlib.h>
3      : #include <string.h>
4      : struct node{
5      :     char name[21];
6      :     struct node *left,*right,*parent;
7      : };
8      586 : struct node * find(struct node *root,char *name)
9      : {
10     586 : struct node *p=NULL;
11     586 : if(strcmp(root->name,name)==0)
12     59 : return root;
13     : else
14     : {
15     527 : if(root->left!=NULL)
16     435 : p=find(root->left,name);
17     527 : if(p!=NULL)
18     190 : return p;
19     337 : else if(root->right!=NULL)
20     92 : p=find(root->right,name);
21     337 : return p;
22     : }
23     :
24     : }
25     :
26     10 : int getPath(struct node *p, struct node *path[])
27     : {
28     10 : int i=0;
29     69 : while(p!=NULL)
30     : {
31     49 : path[i++]=p;
32     40 : p=p->parent;
```

## • 大数据性能评测：性能参与评分



课程公告 | 课程信息 | 学生 | 作业 | 考试 | 题目设置 | 答疑 | 助教 | 增设栏目 | 汇总成绩 | 进入学生端

作业库

作业题目概览

录入作业题目

选择题目

填空题

简答题

编程题

接口编程题

程序片段编程题

算法可视化

并行编程题

文件上传题

作业管理

作业概览

布置作业

作业保管箱

数据结构与程序设计基础

程序运行结果为“do”出现6次，“you”出现6次，“not”出现3次，“as”出现2次。由于单词“do”字典序在“you”之前，因此输出也在其前。

【评分标准】

本题是一个综合性能测试题，测试数据较大(为英文小说)，其评分标准为程序运行结果正确仅占30%，性能分占70%，在所有程序中运行正确且最快的将得满分，最快的程序运行时间为基准，根据其运行时间计算得出。程序运行无结果或结果错误将不得分。

元素路径:

属性信息

难易度: 5 (可选属性 难度等级1~5, 数值越大难度越高)

知识点: 综合

章节: (可选属性)

程序约束

输入类型:  标准输入  文件  命令行参数

输出类型:  标准输出  文件

计算机语言: C++

最长运行时间: 600 秒 (程序运行超过这个时间限制 将会被强制结束, 这个值在1~600之间)

评判规则

评分依据: 正确性 +  性能 (性能分占总分的百分比: 70.0 %) [点击此处查看性能评判帮助](#)

测试数据个数: 3

输出结果验证: 精确匹配 (若选择模糊匹配, 测试数据的期望输出可以使用通配符 '\*' 与 '?', 与程序输出进行模糊比对)

编辑测试数据 [打开测试数据编辑器](#)

```
<testDataSet count="3" mode="offline">
  <testData1>
```

“性能” 在大数据  
下才有区分度!

适用课程:  
算法与数据结构

体验算法魅力, 理解理论与实践的相互作用 (ACM/IEEE CS2013)

计算机体系结构类课程

利用cache、指令多级流水、分支预测等优化程序, 深入理解CPU的微架构, 参考《深入理解计算机系统》

程序性能参与评分

- **大数据性能评测：性能参与评分**
  - 以执行时间最短的程序为基准判分

性能在排行榜上的名次，决定最后得分

学号-姓名	执行时间(S)	得分	优化次数
1506	0.093	5.0	709
1506	0.133	4.1050124	704
1506	0.401	2.6981711	688
1506	42.714	2.0065544	14
1506	45.167	2.0061984	118
1518	45.898	2.0060997	10
1518	54.191	2.0051663	13
1506	59.833	2.004679	21
1406	66.41	2.0042157	34
1518	70.144	2.0039911	25
1527	88.342	2.003169	24
1518	155.912	2.0017955	25
1506	0.084	5.0	746

- (项目) 题目示例

- 词频统计 (100万单词)
- 拼写检查 (90万单词)
- 文本生成器 (120万单词)

- 得分排行榜

- 总分 = 正确分 + 性能分
- 用算法的思维，综合考虑时间复杂度和空间复杂度去寻找最优的方案来解决问题

# 编程竞赛

## 高质量的OJ模块

- ◎与教学系统统一账号登录
- ◎支持ACM竞赛与训练
- ◎竞赛代码查重
- ◎3000道ACM竞赛题库

## • 编程竞赛(CG-OJ)

独立的OJ模块，与CG教学系统统一账号登录，沉淀更多的教学数据



CG Online Judge 首页 题库 竞赛 排行榜 讨论区 F.A.Qs 赵长海 教学系统 教师入口

acm International Collegiate Programming Contest

现任OJ管理员：伍俊洁 junjiwu@buaa.edu.cn

### 最近6个月刷题榜

#	用户	解决问题	提交问题
1	14061116	5	5
2	16231092	4	4
3	comajor	2	2
4	坤	2	2
5	17373157	2	2
6	17373250	2	3
7	maxdumbledore	2	5
8	刘子豪	1	1
9	余天予	1	1
10	17373502	1	1

[完整榜单 >](#)

### 刷题榜

#	用户	解决问题	提交问题
1	14061116	10	10

## 编程竞赛

- 完善的竞赛机制：公开赛和内部赛，ACM-ICPC规则，竞赛代码查重。

竞赛概览 / KDebug冬日捕虫大赛决赛

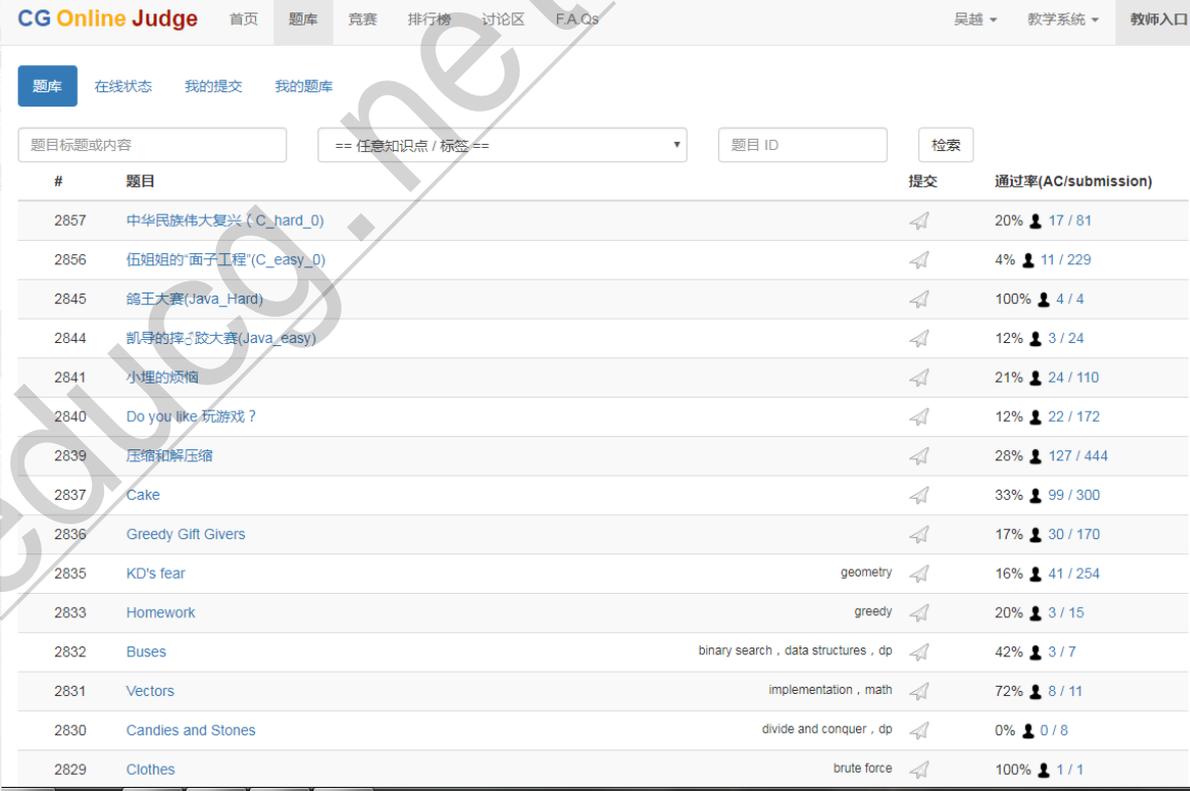
[题目](#)
[在线状态](#)
[排名](#)
导出竞赛排名
First to solve problem
Solved problem
Attempted problem

排名	Who	Solved	Penalty	A	B	C	D	E	F	G	H	I	J	K	L	M	N	删除
1	林昱同 (17373436)	12	15:20:57			01:25:52 (0)	00:23:28 (0)	00:23:36 (0)	00:27:04 (1)	00:27:10 (1)	01:35:22 (10)	01:11:13 (1)	00:52:14 (0)	00:58:17 (0)	01:01:03 (0)	01:04:32 (0)	01:11:06 (0)	
2	杨开元 (17373060)	12	16:50:35			01:44:25 (1)	00:29:36 (0)	00:29:48 (0)	00:55:03 (1)	00:55:10 (0)	01:44:10 (3)	01:34:09 (0)	01:15:59 (1)	01:18:37 (0)	01:21:17 (0)	01:28:24 (0)	01:33:57 (0)	
3	王嘉翼 (王嘉翼)	12	16:55:29			01:23:02 (0)	00:41:20 (0)	00:41:29 (0)	00:45:33 (1)	00:45:44 (0)	01:33:48 (7)	01:17:16 (2)	01:06:07 (0)	01:16:51 (2)	01:06:13 (0)	01:06:19 (0)	01:11:47 (0)	
4	叶苏鹏 (17373457)	10	19:53:02			-5	00:20:00 (0)	01:23:40 (1)	01:56:52 (1)	01:56:59 (1)	-1	01:47:35 (5)	01:48:49 (0)	01:48:55 (0)	01:47:28 (3)	01:49:00 (0)	01:33:44 (0)	
5	林鑫 (15061106)	9	13:05:45			01:35:35 (1)	01:00:57 (0)	01:00:51 (0)	00:50:14 (2)	00:50:22 (0)	01:34:59 (2)		01:47:57 (1)	01:17:54 (0)	01:06:56 (0)			
6	褚台铭 (17373215)	9	14:54:01			01:11:14 (2)	01:07:12 (0)	01:07:22 (0)	01:10:30 (1)	01:10:38 (0)	01:36:02 (1)		01:40:27 (1)	01:54:02 (1)	01:56:34 (0)			
7	马广林 (16151021)	9	15:32:30			01:15:44 (6)	00:57:40 (0)	00:57:49 (0)	01:14:00 (1)	01:14:08 (0)	01:35:12 (4)	-6		01:43:41 (0)	01:23:23 (0)	01:30:53 (0)		

## • 编程竞赛

刷题 + 竞赛选拔，与教学系统互为补充

3000道高质量的训练题库



CG Online Judge 首页 题库 竞赛 排行榜 讨论区 FA Qs 吴越 教学系统 教师入口

题库 在线状态 我的提交 我的题库

题目标题或内容 == 任意知识点 / 标签 == 题目 ID 检索

#	题目	提交	通过率(AC/submission)
2857	中华民族伟大复兴 (C_hard_0)		20% 17 / 81
2856	伍姐姐的“面子工程”(C_easy_0)		4% 11 / 229
2845	鸽王大赛(Java_Hard)		100% 4 / 4
2844	凯哥的橡皮大赛(Java_easy)		12% 3 / 24
2841	小埋的烦恼		21% 24 / 110
2840	Do you like 玩游戏?		12% 22 / 172
2839	压缩和解压缩		28% 127 / 444
2837	Cake		33% 99 / 300
2836	Greedy Gift Givers		17% 30 / 170
2835	KD's fear	geometry	16% 41 / 254
2833	Homework	greedy	20% 3 / 15
2832	Buses	binary search, data structures, dp	42% 3 / 7
2831	Vectors	implementation, math	72% 8 / 11
2830	Candies and Stones	divide and conquer, dp	0% 0 / 8
2829	Clothes	brute force	100% 1 / 1

# 数据库

---

支持SQL自动评测

◎SQL自动评测    ◎在线项目实验

## SQL自动评测：自动、准确量化实验结果

### 数据定义语言

使用SQL语句完成带有表级、列级完整性约束的基本表的创建、修改、删除。

### 数据基本查询

运用SQL语言实现数据的基本查询，包括带条件表达式的查询、自连接查询、分组统计查询、多表连接查询。

### 数据高级查询

支持SQL嵌套查询和集合查询。

### 数据更新语言

支持使用SQL语句完成基本表中数据的插入、更新、删除。



作业题库

题目描述包括： 题目描述 题目内容包括： 题目内容

导出选中题目 删除选中题目 复制选中题目至题库

选择：全选 取消

题目	类型
<input type="checkbox"/> 6250. 查找最晚入职员工的所有信息	SQL评测题
<input type="checkbox"/> 6249. 查找入职时间排名倒数第二的员工所有信息	SQL评测题
<input type="checkbox"/> 6248. 查找各个部门当前(to_date='9999-01-01')领导当前薪水详情以及其对应部门编号dept_no	SQL评测题 数据库操作语言
<input type="checkbox"/> 6247. 查找所有已经分配部门的员工的last_name和first_name	SQL评测题 数据库操作语言
<input type="checkbox"/> 6246. 从titles表获取按照title进行分组	SQL评测题 数据库操作语言
<input type="checkbox"/> 6245. 使用join查询方式找出没有分类的电影id以及名称	SQL评测题 数据库操作语言
<input type="checkbox"/> 6244. 创建表actor，并插入数据	SQL评测题 数据库定义语言

作业管理

- 作业概述
- 布置作业
- 回收站

当前题目：6250. 查找最晚入职员工的所有信息

查找最晚入职员工的所有信息

```
CREATE TABLE `employees` (  
  `emp_no` int(11) NOT NULL,  
  `birth_date` date NOT NULL,  
  `first_name` varchar(14) NOT NULL,  
  `last_name` varchar(16) NOT NULL,  
  `gender` char(1) NOT NULL,  
  `hire_date` date NOT NULL,  
  PRIMARY KEY (`emp_no`));
```

提交SQL语句：

```
1 |select * from employees where hire_date = (select max(hire_date) from employees);
```

SQL语法高亮

提交

完全正确

支持多行SQL语句

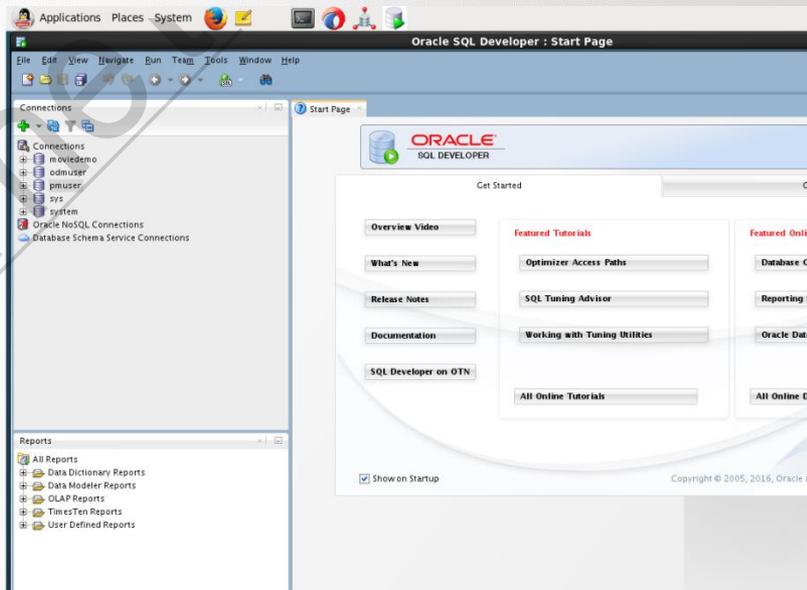
给出SQL语句调试信息

## 在线实验：体验真实的数据库操作环境，解决数据库安装部署难题

**进行复杂的实验。**数据库表设计、数据库编程、撰写一个简单的数据库系统等。

**随时随地开展实验。**解决传统机房的时空限制。

**精细化评价。**传统实验只能根据实验报告来判分，主观性非常强，在线实验系统完整记录学生实验过程中的操作记录。



# 软件工程

## 完整的软件工程教学和实训体系

- ◎小组协作
- ◎文档 / 源码查重
- ◎增量式迭代开发辅助管理
- ◎小组互评
- ◎github项目管理

## 丰富的题目类型

5类  
通用  
题型

- 选择题
- 填空题
- 判断题
- 简答题
- 编程题
- 文件上传题

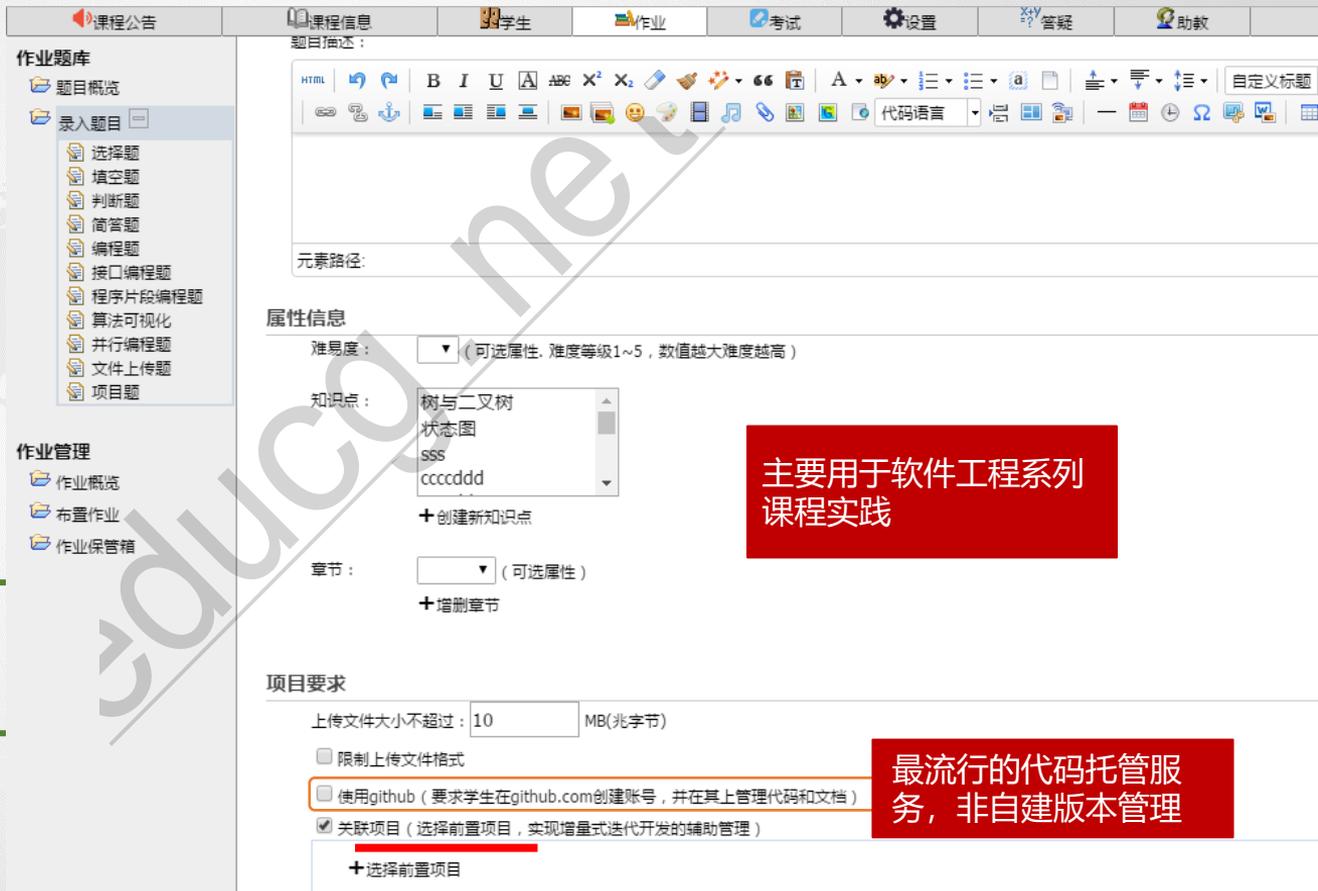
5类  
编程  
题型

- 接口编程题
- 程序片段编程题
- 算法可视化
- SQL评测题

1类  
并行编程  
题型

- 并行编程题
  - MPI分布式
  - 多线程

- **项目题**
- 通用评测题



The screenshot shows a web interface for course management. On the left, there are two panels: '作业题库' (Question Bank) and '作业管理' (Assignment Management). The '作业题库' panel has a tree view with '录入题目' (Enter Questions) expanded, showing various question types like '选择题' (Multiple Choice), '填空题' (Fill-in), '判断题' (True/False), '简答题' (Short Answer), '编程题' (Programming), '接口编程题' (Interface Programming), '程序片段编程题' (Code Snippet Programming), '算法可视化' (Algorithm Visualization), '并行编程题' (Parallel Programming), '文件上传题' (File Upload), and '项目题' (Project). The '作业管理' panel has '作业概况' (Assignment Overview), '布置作业' (Assign Homework), and '作业保管箱' (Assignment Storage). The main content area shows '课程公告' (Course Announcement), '课程信息' (Course Information), '学生' (Students), '作业' (Assignments), '考试' (Exams), '设置' (Settings), '答疑' (Q&A), and '助教' (TA). Below these are tabs for '题目描述' (Question Description), '属性信息' (Attribute Information), and '项目要求' (Project Requirements). The '属性信息' section includes '难度度' (Difficulty) with a dropdown, '知识点' (Knowledge Point) with a list (树与二叉树, 状态图, sss, cccdd) and a '+ 创建新知识点' button, and '章节' (Chapter) with a dropdown and a '+ 增删章节' button. The '项目要求' section includes '上传文件大小不超过: 10 MB(兆字节)', '限制上传文件格式' (checkbox), '使用github (要求学生在github.com创建账号, 并在其上管理代码和文档)' (checkbox), and '关联项目 (选择前置项目, 实现增量式迭代开发的辅助管理)' (checkbox). A '+ 选择前置项目' button is at the bottom.

主要用于软件工程系列  
课程实践

最流行的代码托管服  
务, 非自建版本管理

## • 软工实践教学

- 小组作业、互评、GitHub协作



**属性信息**

将作业布置给: 11级学生

作业开始时间: 2017-05-31 00:00

作业终止时间: 2017-06-10 23:59

设置补交截止时间

**分组设置**

每组最少人数: 2 每组最多人数: 5

启用小组互评

每组评审者数量: 5 小组互评得分占总比例: 30.00%

互评分计算方法:  去掉最大最小求平均  求平均

互评起止时间: 2017-06-11 00:00 至 2017-06-17 00:00

**小组(匿名)互评**

**题目列表**

题目描述	相似性比较	分值
4183. 实践1: 测试计划文档 一关联项目: 实践1: 设计文档	<input checked="" type="checkbox"/> 文档相似性 <input type="checkbox"/> 源代码相似性	30.0

**开发过程**

## • 文档 / 源代码相似性检测

- 从压缩包内提取**文档**和**源代码**，进行相似性比较

作业概览 / 2016年编译技术课程设计 / 相似性检测题目列表 / 扩充C0文法-中-数组

文档相似性比较

源代码相似性比较

Word | PDF |

文档列表
[11231033]
[12061]
[120611]
[13005]
[13005]
[1305]
[1306100]
[130610061]
[130610]
[130610]
[130610]
[13061046]

相似文档聚类结果 (点击左侧列表, 可以查看详情):

相似文档聚类结果 (点击左侧列表, 可以查看详情):

各种常见文档: Word、PDF、Excel、PPT

适用课程:

软件工程、编译技术、各种实验报告.....

作业概览 / 2016年编译技术课程设计 / 相似性检测题目列表 / 扩充C0文法-中-数组

文档相似性比较

源代码相似性比较

Language: C/C++

Subn

相似度分布情况:

90% - 100% 0  
80% - 90% 0  
70% - 80% 0  
60% - 70% 0  
50% - 60% 0  
40% - 50% 0

整体相似性比较结果 (点击链接可以进行详细比对):

相似文档-14061097-> 相似文档-14061071 (5.5%) 相似文档-14061023 (3.3%)  
相似文档-14061071-> 相似文档-14061023 (2.7%)

部分相似性比较结果 (点击链接可以进行详细比对):

相似文档-14061097-> 相似文档-14061071 (5.8%) 相似文档-14061023 (3.9%)  
相似文档-14061071-> 相似文档-14061023 (3.0%)

常用编程语言: C/C++、C#、Java、Python、Javascript、HTML等

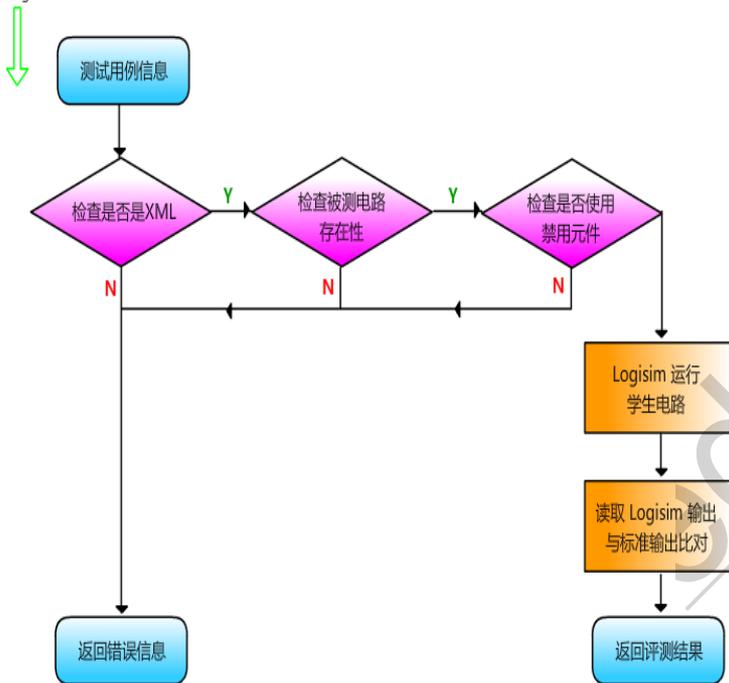
# 计算机组成原理实验

## 支持自动评测的在线实验体系

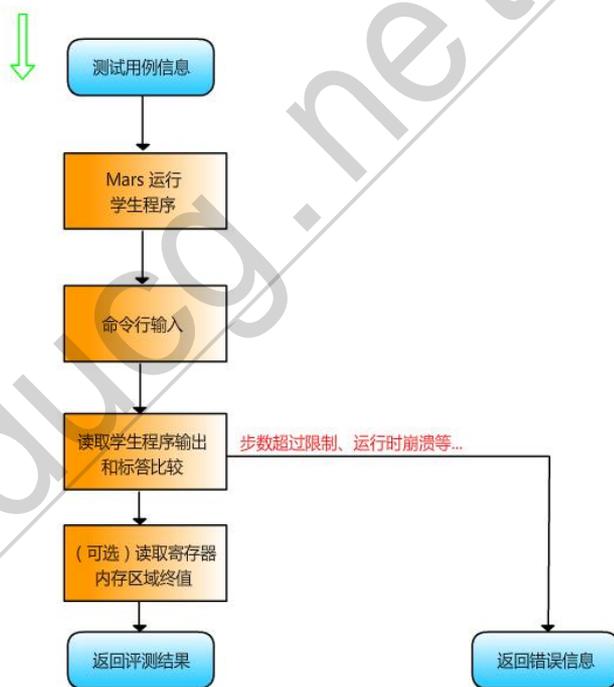
- ◎ 数字逻辑
- ◎ 汇编语言
- ◎ 数字部件
- ◎ CPU

## • 基于B/S架构的自动评测系统：自动化评价学生提交的实验代码。

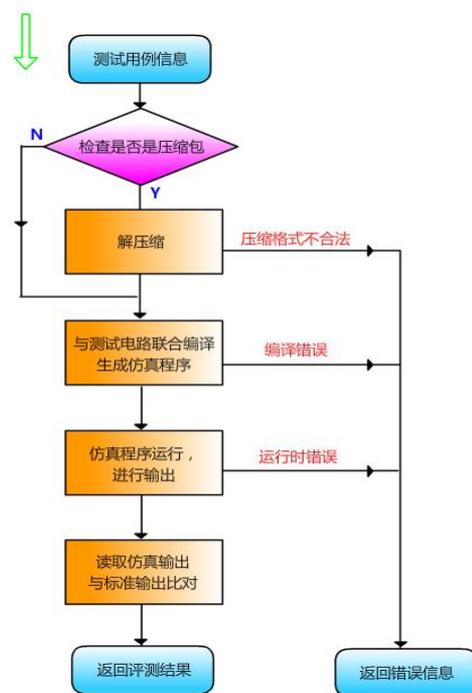
• Logisim 评测流程



• Mars 评测流程



• ISE 评测流程



## • 计算机组成在线实验：支持自动评测的在线实验体系

### 基于B/S架构的自动评测系统

自动化评价学生提交的实验代码。在线自动化测试使得严谨完善的测试成为可能，同时测试结果的统计分析有力支撑了教学设计改进。

### 计算机组成实验体系

涵盖3大类实验：**数字逻辑**、**汇编语言**、**计算机组成**。

语言/工具：MIPS汇编/Verilog、Logisim、MARS、ISE。

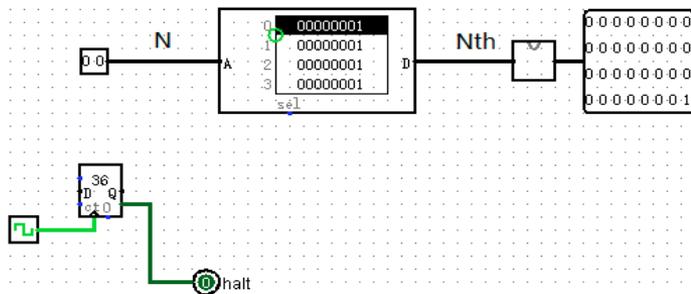
### 实验列表详见：

- <http://www.educg.net/computer.html>

#### 提交要求

使用Logisim搭建一个根据输入序号x计算对应序号斐波那契数fib[x]的电路（输入序号0对应输出数0，输入序号1对应输出数1，输入序号2对应输出数1，以此类推）并提交。

- 输入：N（32bit无符号数）
- 输出：Nth（32bit无符号数，表示第N个斐波那契数）
- 文件内模块名：main
- 测试电路图：



- 测试要求：在64个周期内计算出结果并稳定输出，在结果未计算出之前输出口输出0。
- HINT：矩阵乘法的快速幂

## • 计算机组成在线实验：支持自动评测的在线实验体系

### 基于B/S架构的自动评测系统

自动化评价学生提交的实验代码。在线自动化测试使得严谨完善的测试成为可能，同时测试结果的统计分析有力支撑了教学设计改进。

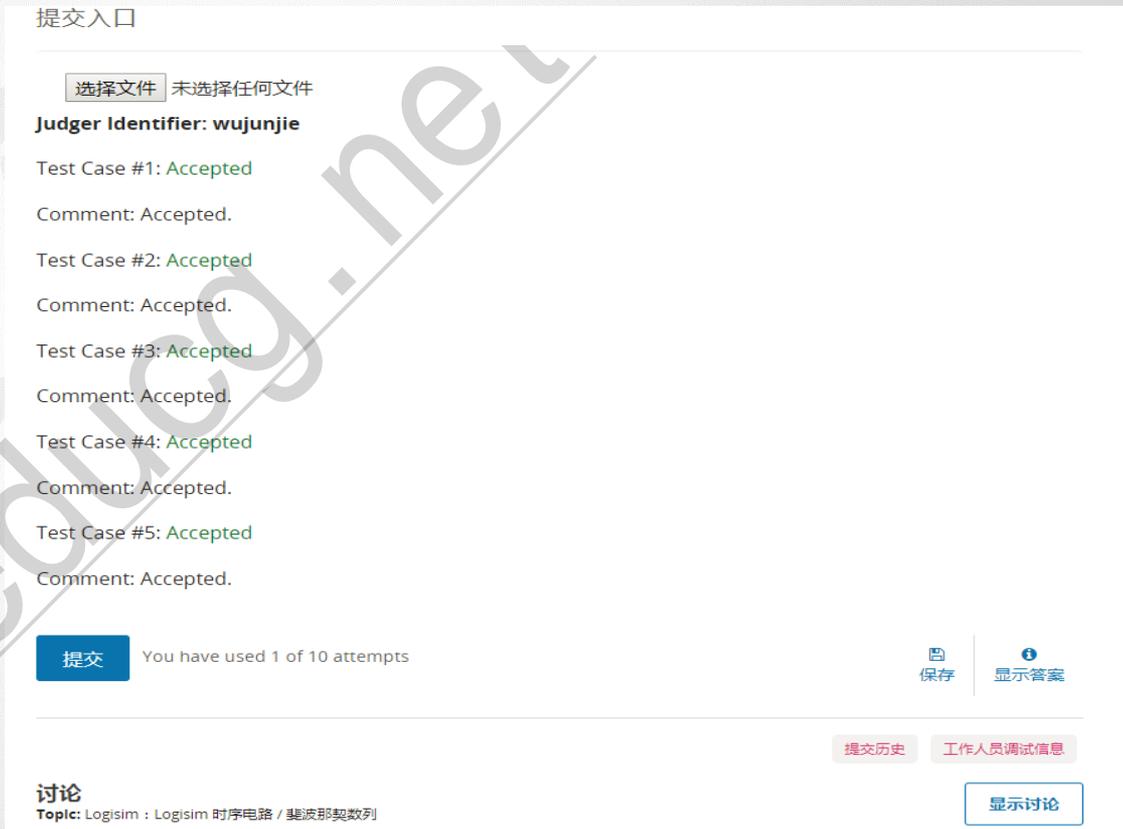
### 计算机组成实验体系

涵盖3大类实验：**数字逻辑**、**汇编语言**、**计算机组成**。

语言/工具：MIPS汇编/Verilog、Logisim、MARS、ISE。

### 实验列表详见：

- <http://www.educg.net/computer.html>



提交入口

未选择任何文件

Judge Identifier: wujunjie

Test Case #1: Accepted  
Comment: Accepted.

Test Case #2: Accepted  
Comment: Accepted.

Test Case #3: Accepted  
Comment: Accepted.

Test Case #4: Accepted  
Comment: Accepted.

Test Case #5: Accepted  
Comment: Accepted.

You have used 1 of 10 attempts

讨论  
Topic: Logisim : Logisim 时序电路 / 斐波那契数列

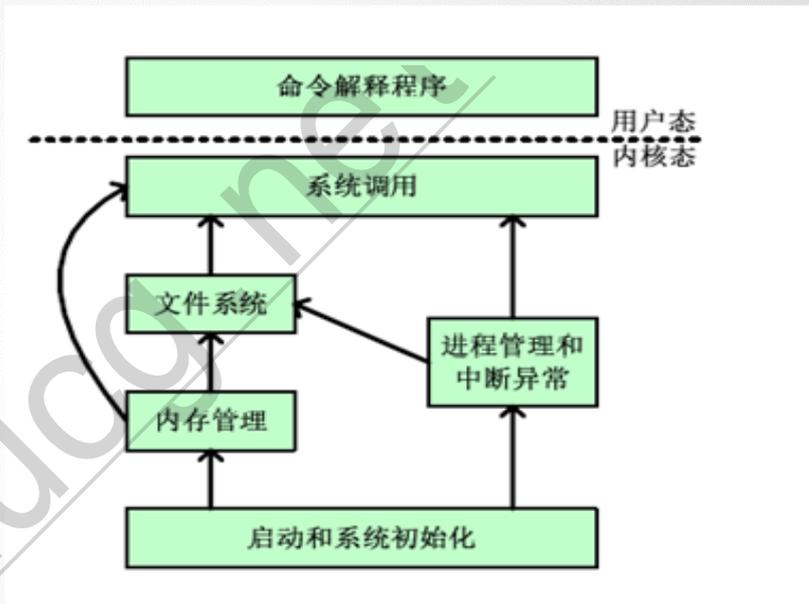
# 操作系统在线实验

## 支持自动评测和过程管理

- ◎内核实验自动评测
- ◎实验过程自动化管理
- ◎小型操作系统实验体系
- ◎在线虚拟实验环境

## • 操作系统实验：内核实验

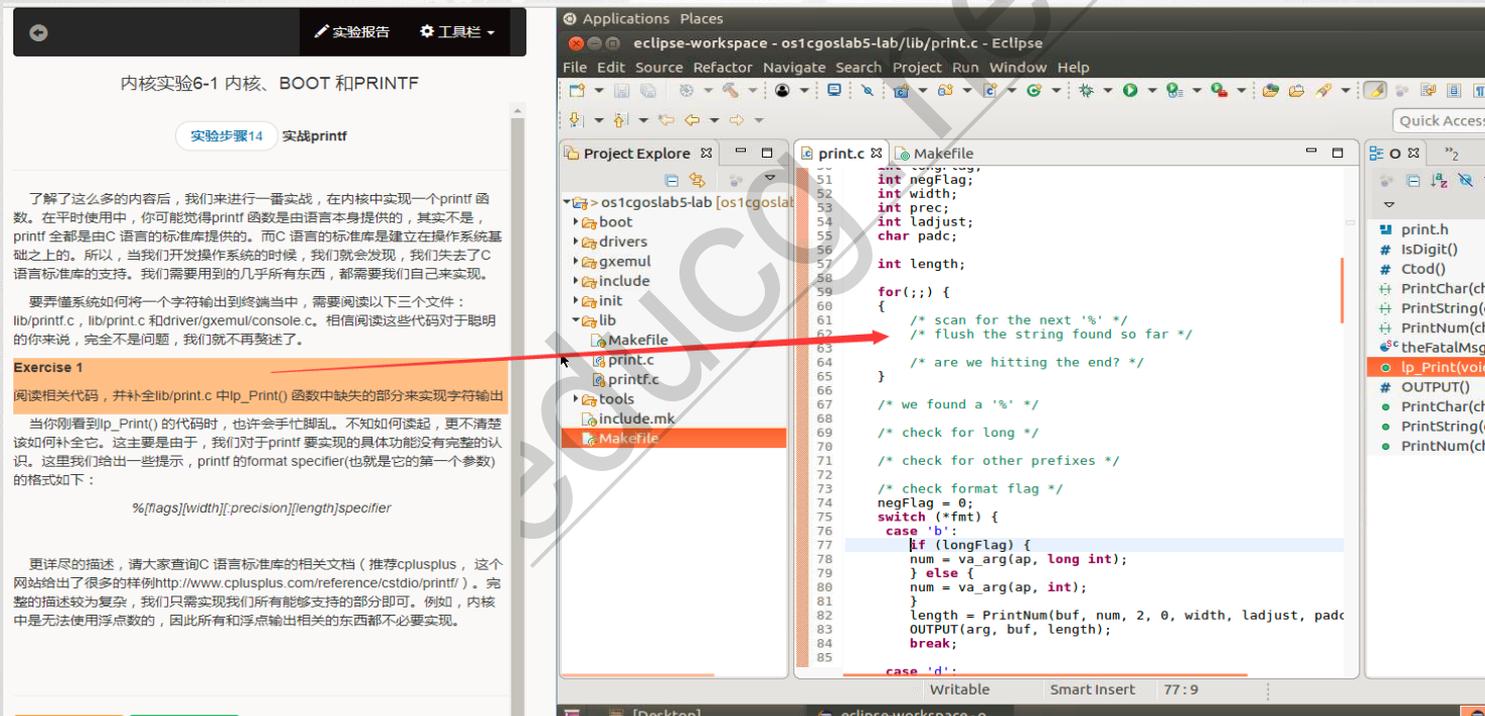
- 内核实验自动评测
- 实验过程自动化管理
- 小型操作系统实验体系：
  - 实验1：启动和系统初始化
  - 实验2：内存管理实验
  - 实验3：进程管理
  - 实验4：系统调用
  - 实验5：文件系统
  - 实验6：命令解释程序



6个相互关联的操作系统实验，学生可以从简单到复杂，最终构造出一个相对完整的操作系统。

## • 操作系统实验：内核实验

每个操作系统实验都设置多个不同难度级别的挑战性任务，并自动评测，节省教师工作量的同时，让学生获得内核开发的成就感。



The screenshot displays an Eclipse IDE environment. On the left, a web browser window shows the '内核实验6-1 内核、BOOT 和PRINTF' page. The page content includes:

了解了这么多的内容后，我们来进一步实战，在内核中实现一个printf函数。在平时使用中，你可能觉得printf函数是由语言本身提供的，其实不是，printf全都是由C语言的标准库提供的。而C语言的标准库是建立在操作系统基础之上的。所以，当我们开发操作系统的时候，我们就会发现，我们失去了C语言标准库的支持，我们需要用到的几乎所有东西，都需要我们自己来实现。

要弄懂系统如何将一个字符输出到终端当中，需要阅读以下三个文件：lib/print.c，lib/print.c和driver/gxemul/console.c。相信阅读这些代码对于聪明的你来说，完全不是问题，我们就不再赘述了。

**Exercise 1**

阅读相关代码，并补全lib/print.c中lp\_Print()函数中缺失的部分来实现字符输出

当你刚看到lp\_Print()的代码时，也许会手忙脚乱。不知如何读起，更不清楚该如何补全它。这主要是由于，我们对于printf要实现的具体功能没有完整的认识。这里我们给出一些提示，printf的format specifier(也就是它的第一个参数)的格式如下：

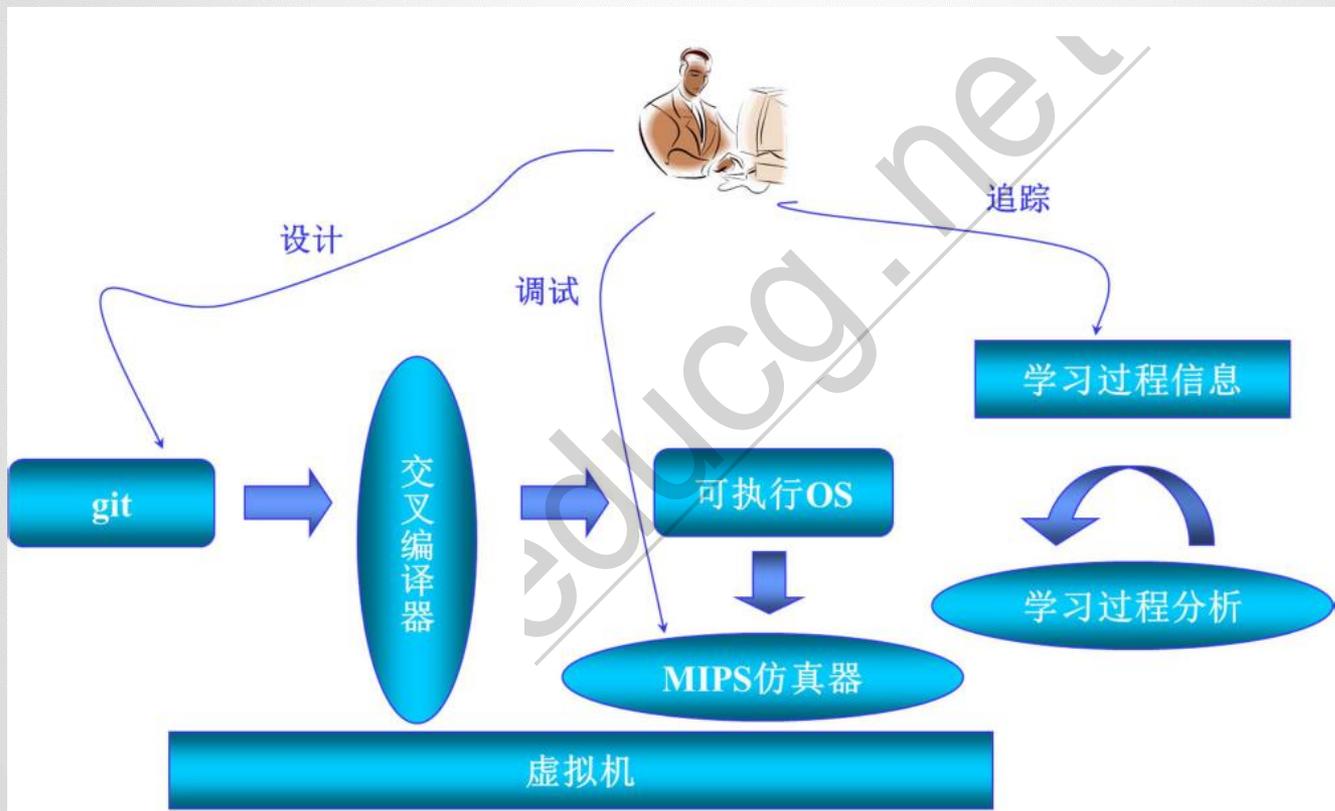
```
%[flags][width][.precision][length]specifier
```

更详尽的描述，请大家查阅C语言标准库的相关文档(推荐cplustplus，这个网站给出了很多的样例<http://www.cplustplus.com/reference/cstdio/printf/>)。完整的描述较为复杂，我们只需实现我们所有能够支持的部分即可。例如，内核中是无法使用浮点数的，因此所有和浮点输出相关的东西都不必要实现。

On the right, the Eclipse IDE shows the 'print.c' file with the following code:

```
1  #include <stdio.h>
2
3  #include "lib.h"
4
5  #include "print.h"
6
7  #include "console.h"
8
9  #include "string.h"
10
11 #include "string_helpers.h"
12
13 #include "string_helpers.h"
14
15 #include "string_helpers.h"
16
17 #include "string_helpers.h"
18
19 #include "string_helpers.h"
20
21 #include "string_helpers.h"
22
23 #include "string_helpers.h"
24
25 #include "string_helpers.h"
26
27 #include "string_helpers.h"
28
29 #include "string_helpers.h"
30
31 #include "string_helpers.h"
32
33 #include "string_helpers.h"
34
35 #include "string_helpers.h"
36
37 #include "string_helpers.h"
38
39 #include "string_helpers.h"
40
41 #include "string_helpers.h"
42
43 #include "string_helpers.h"
44
45 #include "string_helpers.h"
46
47 #include "string_helpers.h"
48
49 #include "string_helpers.h"
50
51 int negFlag;
52 int width;
53 int prec;
54 int ladjust;
55 char padc;
56
57 int length;
58
59 for(;;) {
60 {
61 /* scan for the next '%' */
62 /* flush the string found so far */
63
64 /* are we hitting the end? */
65 }
66
67 /* we found a '%' */
68
69 /* check for long */
70
71 /* check for other prefixes */
72
73 /* check format flag */
74 negFlag = 0;
75 switch (*fmt) {
76 case 'b':
77     if (longFlag) {
78         num = va_arg(ap, long int);
79     } else {
80         num = va_arg(ap, int);
81     }
82     length = PrintNum(buf, num, 2, 0, width, ladjust, padc);
83     OUTPUT(arg, buf, length);
84     break;
85 case 'd':
```

## • 操作系统实验：在线实验环境



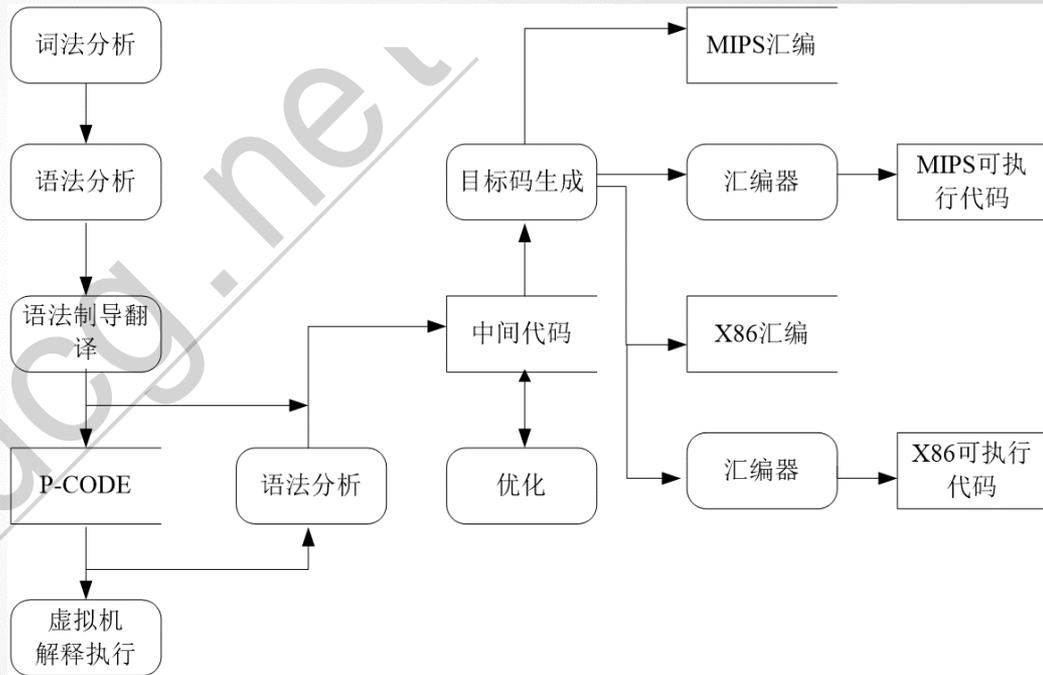
# 编译技术实验

## 支持全自动化评测的小编译器实验体系

- ◎词法分析
- ◎代码生成
- ◎代码优化
- ◎语法分析
- ◎目标码运行

## • 支持编译构建全过程分阶段自动评判的编译实验系统

- 难度层次一，仿Java虚拟机，编译解释性。完成PL/0或C0编译器的设计、编码、实现和测试（含出错处理能力）。
- 难度层次二，面向特定硬件平台的代码生成。完成C1语言编译器的设计、编码、实现和测试（含出错处理能力）C1增加了数据结构类型，如数组；增加控制结构。
- 难度层次三，编译优化，竞速排名。在提高层级基础，进行编译器优化；挑战极限，竞速排名。



## 支持编译构建全过程分阶段自动评判的编译实验系统

- 难度层次一，仿Java虚拟机，编译解释性。完成PL/0或C0编译器的设计、编码、实现和测试（含出错处理能力）。
- 难度层次二，面向特定硬件平台的代码生成。完成C1语言编译器的设计、编码、实现和测试（含出错处理能力）C1增加了数据结构类型，如数组；增加控制结构。
- 难度层次三，编译优化，竞速排名。在提高层级基础，进行编译器优化；挑战极限，竞速排名。



#	账号	Team	提交次数	最后提交时间	功能得分	性能得分	总分
1	chen10614	燃烧我的编译器 / 中国科学院大学	44	2020-08-20 15:56:46	100	93.26	94.75
2	HarryChen	编程是一件很危险的事情 / 清华大学	7	2020-08-20 11:14:30	100	69.13	75.99
3	17373451	段地址小队 / 北京航空航天大学	30	2020-08-20 16:52:44	100	60.68	69.42
4	cutepiler	想要编得更可爱一点 / 清华大学	13	2020-08-20 16:25:05	100	55.24	65.18
5	buaa_scse	栈地址小队 / 北京航空航天大学	9	2020-08-20 10:09:54	100	50.84	61.77
6	SuperSodaSea	SuperSodaSea / 上海大学	21	2020-08-20 16:38:01	100	50.47	61.48
7	y1nG1u0	DR直呼内行 / 北京科技大学	12	2020-08-20 13:51:49	100	43.83	56.31
8	dby	YAL / 中国科学技术大学	10	2020-08-20 16:15:24	100	37.87	51.67
9	follower	形式语言与复读机 / 哈尔滨工业大学 (深圳)	10	2020-08-20 16:45:57	100	36.10	50.30
10	610098971	创造1010 / 电子科技大学 (清水河)	14	2020-08-20 17:44:16	100	35.85	50.11
11	zenglj	秋水向洋 / 西北工业大学	24	2020-08-20 16:25:43	100	28.70	44.54
12	2471346734	Q2blcGlsZXI= / 东北大学	25	2020-08-20 17:15:54	100	23.34	40.38
13	ly_cs	承影圆南 / 西北工业大学	8	2020-08-20 17:16:22	100	20.96	38.52
14	Zhanglujian	华工计创队 / 华南理工大学	13	2020-08-20 02:36:41	100	20.84	38.43
15	XuYao	BitHugeCC / 北京理工大学	21	2020-08-20 18:16:52	100	20.70	38.32

- 按照编译流程，将实验划分为词法分析、语法分析、错误处理、代码生成、代码优化五个部分。
  - **实验1：词法分析。**根据给定的文法设计并实现词法分析程序。
  - **实验2：语法分析。**根据给定的文法设计并实现语法分析程序，能基于词法分析程序所识别出的单词，识别出各类语法成分。
  - **实验3：错误处理。**根据给定的文法设计并实现错误处理程序，能诊察出常见的语法和语义错误，进行错误局部化处理，并输出错误信息。
  - **实验4：代码生成。**在词法分析和语法分析的基础上，为编译器实现语义分析、代码生成功能。
  - **实验5：代码优化。**学生需要为编译器实现代码优化功能（MIPS）。

- 按照编译流程，将实验划分为词法分析、语法分析、错误处理、代码生成、代码优化五个部分。



词法分析



语法分析



错误处理



代码生成



代码优化

当前通用评测题：5338. 2019竞速排序优化编译器提交

### 【作业描述】

请在之前作业的基础上，为编译器实现现代代码优化功能，输入输出及处理要求如下：

- 将源文件（统一命名为testfile.txt）编译生成MIPS汇编并输出到文件中（统一命名为mips.txt），具体要求包括：
  - 自行设计四元式中间代码，再从中间代码生成MIPS汇编，中间代码输出的格式要求见文件“中间代码格式要求2019.docx”，请实现输出中间代码的有关函数，将优化前后的中间代码分别输出到命名为“学号\_姓名\_优化前/后中间代码.txt”的文件中；
  - 凡选择生成MIPS代码的同学，均需提交此项作业；
  - 自行测试时，可使用Mars的仿真器（使用方法见“Mars的仿真器使用说明.docx”），提交到平台的编译器只需要能按统一的要求生成MIPS汇编代码文件即可。
- 优化的要求如下：
  - 代码生成时合理利用寄存器，包括全局寄存器和临时寄存器，建议使用规范的全局寄存器分配方法（例如引用计数或者着色法）全局寄存器，使用临时寄存器池管理和分配临时寄存器。遵循MIPS的寄存器使用规范和调用规范，并能生成较高质量的目标代码；
  - 建议实现基本块内部的公共子表达式删除（DAG图）优化方法；
  - 建议通过数据流分析（活跃变量分析或者到达定义分析）实现较复杂的全局优化，例如循环不变量外提，构建变量冲突图等等；
  - 建议实现课堂上介绍过的某些典型优化，例如代码内联、常量传播、复制传播、窥孔优化等等；
  - 优化最终评分依据以竞速排名结果为主，完成优化的数量和种类、生成代码质量等因素，仅作为辅助评分依据。
- 需要中优的同学请提交中优文章到本次作业的另一道题目录下，以指导如何完成课程设计为主要内容，兼或说明在课程设计中遇到个困难点及所采用的解决方案，要求内容具体，阐述清楚，题目和内容组织自定。注意，未提交中优文章视为放弃中优，提交的中优文章将发布给下一届同学。

【输入形式】testfile.txt为符合语法要求的测试程序，另外可能存在来自于标准输入的数据。

【输出形式】按如上要求将目标代码生成结果输出至mips.txt中，并输出优化前后的中间代码，中文字符的编码格式要求是UTF-8。

【评分要求】只有运行结果完全正确才能进入排序队列，根据FinalCycle = ALU \* 1 + Jump \* 2 + Branch \* 2 + Memory \* 2 + Other \* 1的值得FinalCycle越小排名越靠前。

\*设置的分数只为直观展示得分比例，汇总成绩时将按统一的标准进行折算。

限时排行榜批量评分

2019年竞速排序作业 | 2019竞速排序优化编译器提交 [满分: 10.0]

排名范围: 排名范围限制(1-12) 分数: 0.0 确定

### 排行榜

#	学号	姓名	Branch	Memory	ALU	Other	Jump	rank	总用时(秒)	修订分
1	0	0	382	163	1	547.00	9215	10.00	修改	
2	12708	329	61743	162	18	89015.00	194663	10.00	修改	
3	20257	566	63874	162	14	105710.00	3412	10.00	修改	
4										
5	24466	549	57721	163	15	107944.00	79038	10.00	修改	
6	25516	555	57619	162	272	110467.00	15782	10.00	修改	
7	25515	550	57754	162	268	110562.00	9978	10.00	修改	
8	25516	4215	59682	163	12	119331.00	2215	10.00	修改	
9										
10	25516	580	80871	162	279	133783.00	2974	10.00	修改	
11	25516	570	81581	162	275	134485.00	16059	10.00	修改	
12	25516	700	81938	162	11	134554.00	4662	10.00	修改	
13										
14	25513	562	82316	163	12	134653.00	1356	10.00	修改	

代码行: 3456 | 正确率: 0.95 | 完成时间: 102分钟

提交源文件，只能提交以 .rar, .zip 为后缀的文件：

选择文件 | final.zip

提交

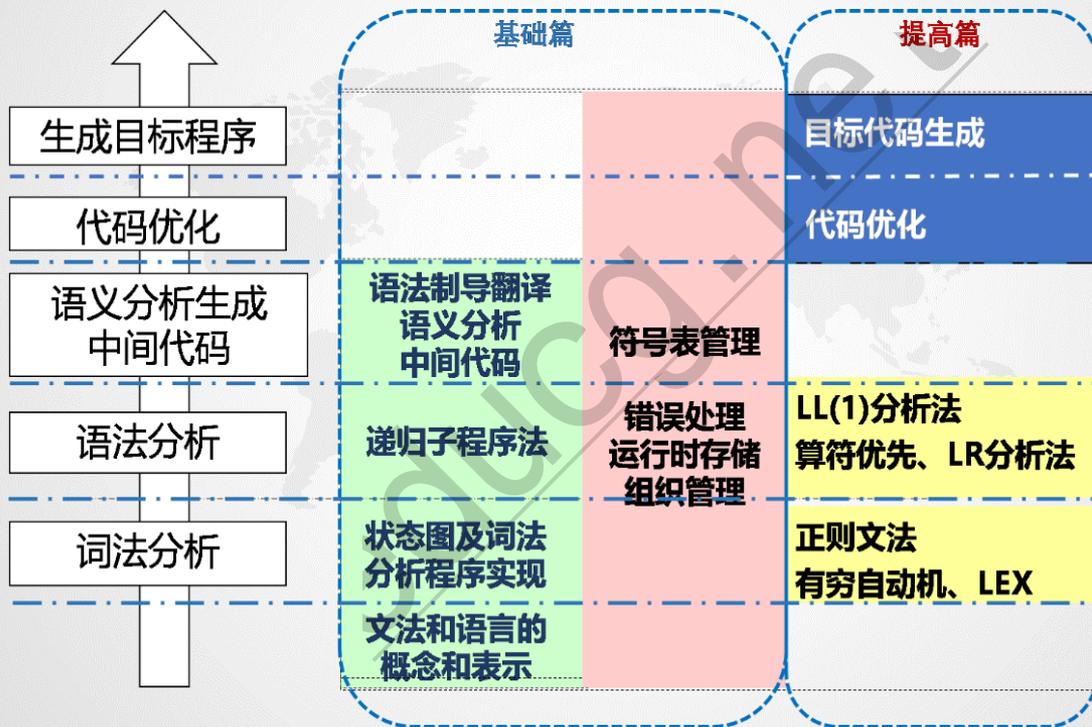
调试

下载源文件

Accepted

- 《编译技术》教材：将复杂的自动生成技术从编译过程中剥离出来，便于学生理解和掌握编译过程。

- 视频
- 讲义
- 习题



# 并行计算

## 科学全面的评判与实验体系

- ◎自动量化正确性、性能、可扩展性
- ◎评判结果可视化
- ◎支持多种并行语言和计算架构

## • 并行程序的自动评测

- 从**正确性**、**可扩展性**、**性能**三个角度全面评测并行程序
- 支持分布式MPI、多核多线程

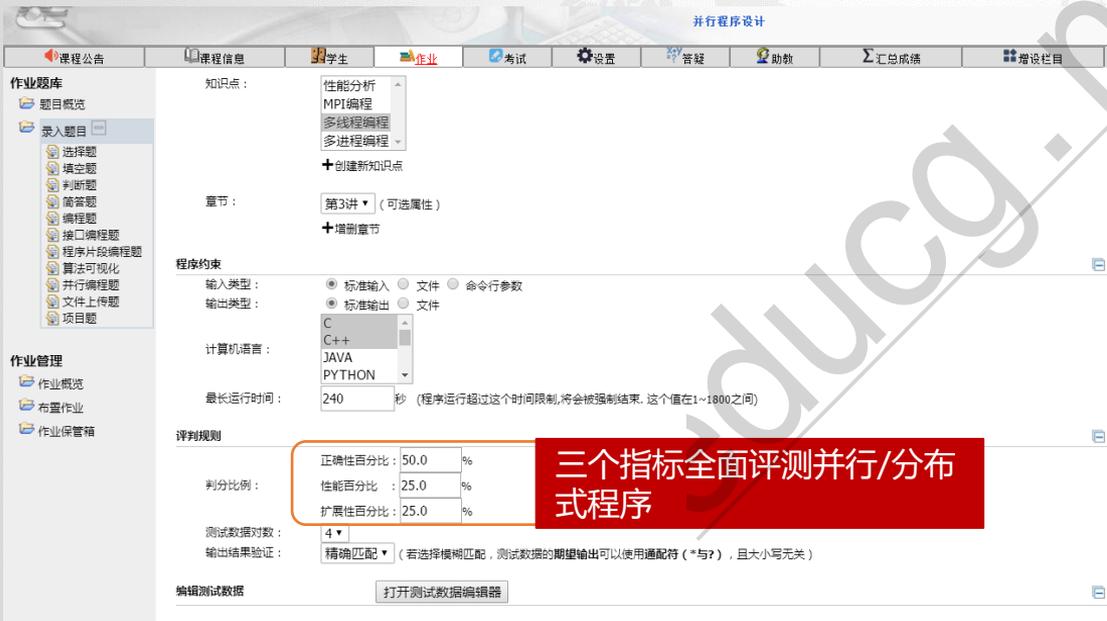
## 适用课程：

《并行程序设计/并行计算》

《操作系统实验》

多进程实验 (wait 与 fork)  
信号量  
生产者消费者

ACM/IEEE CS2013新增课程体系：  
并行和分布式计算 (PD)、信息安全、系统基础、基于平台的开发 等



评判规则	正确性百分比	性能百分比	扩展性百分比
判分比例：	50.0 %	25.0 %	25.0 %

三个指标全面评测并行/分布式程序

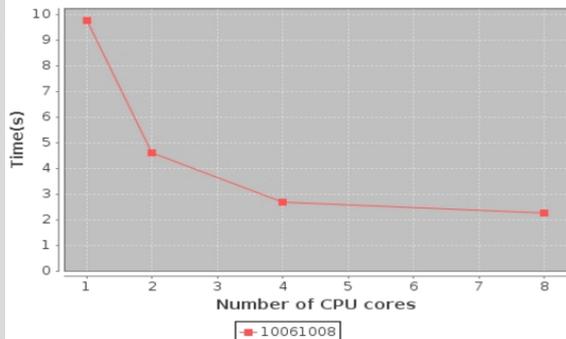
## • 并行程序的自动评测：评判报告可视化

程序通过正确性测试！

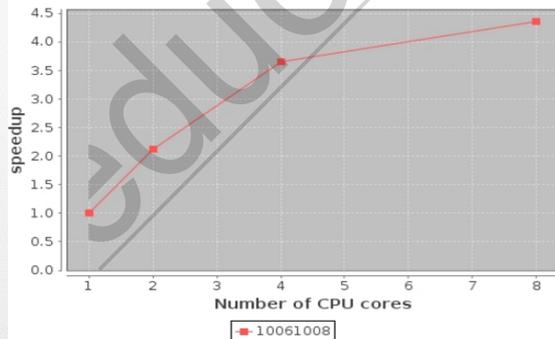
说明：最后得分综合并行程序的“正确性”，“性能”与“可扩展性”三项得到，性能和可扩展性分别由执行时间和加速比来衡量。占用内存不作为判定依据

CPU核数	占用内存(KB)	执行时间(S)	加速比
1	47.884	9.755	1
2	46.334	4.602	2.12
4	46.433	2.67	3.654
8	46.368	2.24	4.355

执行时间  
Execution time



加速比  
Speedup



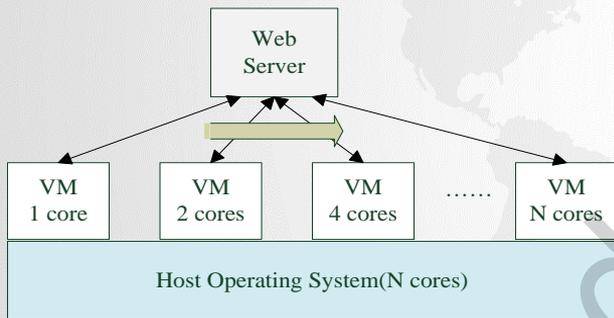
排行榜

点击表头字段，可根据该字段排序  
点击“学号-姓名”列中任意一项，生成对比图表  
性能和可扩展性在排行榜上的名次，决定最后得分

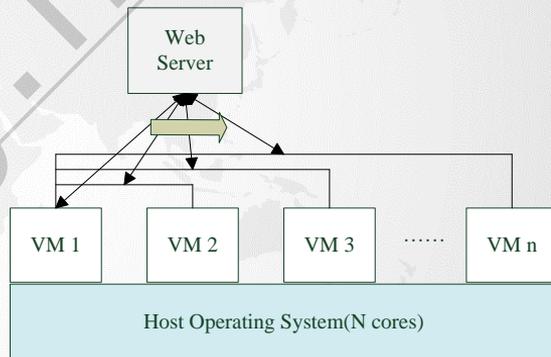
学号-姓名	性能	可扩展性	得分	优化次数
10061008	11.67	0.192	14.7236	15
10061009	2.357	0.075	16.5938	10
10061010	4.94	0.06	14.2321	10
10061011	24.128	0.015	12.0975	9
10061012	3.498	0.066	15.1309	8
10061013	2.175	0.057	16.7219	8
10061014	3.135	0.063	15.4086	8
10061015	4.054	0.033	14.3389	7
10061016	3.293	0.046	15.0553	6
10061017	4.2	0.058	14.5625	6
10061018	11.992	0.103	13.6004	6
10061019	3.349	0.048	15.0305	3
10061020	3.057	0.051	15.3464	2
10061021	9.755	0.054	13.1781	2
10061022	2.183	0.058	16.7246	2
10061023	16.934	-0	10.5801	2
10061024	11.277	0.075	13.2965	2
10061025	6.151	0.137	14.7996	2
10061026	11.411	0.056	13.0595	1
10061027	10.861	0.075	13.3324	1

- 并行程序的自动评测：虚实结合的自动评测环境

既可以使用单台服务器虚拟出多线程和分布式集群环境，也可以直接使用现有高性能计算系统。



多线程评测“集群”



MPI分布式评测“集群”



02

---

**人工智能专业**

# 任务驱动实验体系

基于四种实验环境构建了完善的任务驱动实验体系

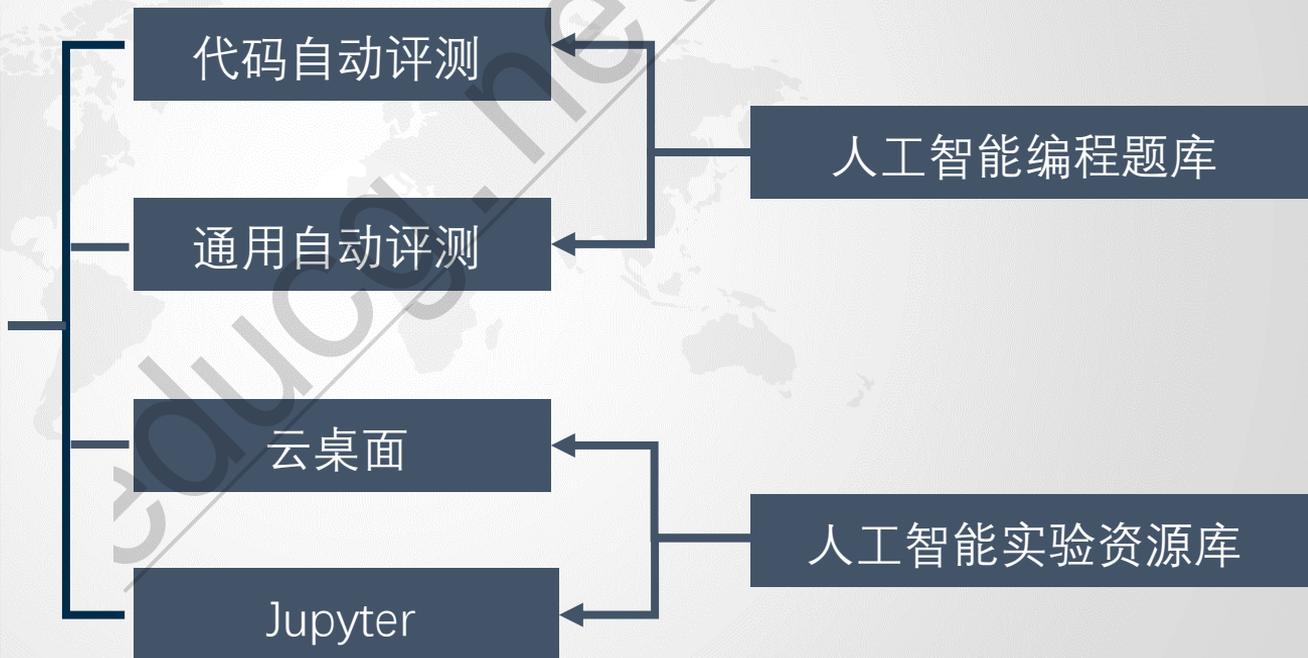
- ◎ 基于代码评测的实验任务
- ◎ 基于通用评测的实验任务
- ◎ 基于云桌面的实验任务
- ◎ 基于Jupyter的实验任务

基于希冀平台的代码评测、通用评测、云桌面、Jupyter四种不同的实验环境分别构建了不同的任务驱动实验体系。每个任务型实验均提供参考答案。

学生任务包括：

- 编写代码提交通过评测
- 完成实验中设置的实验任务

任务驱动实验体系



基于代码评测实验环境的《机器学习》算法编程实验样例。

学生需要补全代码，提交评测。

当前程序片段编程题：

【问题描述】  
给定鸢尾花数据，你  
要点：  
• 程序包含4个  
• load\_data(),  
• data\_split(),  
来划分训练集  
random的初始  
• fit(), 设计并  
分, 代码实现  
• predict(), 对

【输入形式】  
数据集类型为pd.DataFrame  
共有100行数据，每行  
在列“花的种类”中，1  
【输出形式】  
输出为两行：  
第一行包含两列，依  
第二行为测试集上的  
【样例输入】  
点击下载数据集文件：  
实现并按格式进行返回。

```
4.6,3.1,1.
5.0,3.6,1.
6.5,2.8,4.
5.7,2.8,4.
6.3,3.3,4.
4.9,2.4,3.
6.6,2.9,4.

[[ 3.4
  0.1]
 0.0
 0.83]
```

```
import numpy as np
import random
def load_data():
    inp = input()
    data = []
    while (inp):
        dx = [float(di) for di in
            data.append(dx)
        inp = input()
    data = np.array(data)
    return data

def data_split(X,y,test_size=0.4,rand
n_samples = len(X)
assert len(X)==len(y)

indices = np.arange(n_sample
random.seed(random_state)

train_indices = list(set(
test_indices = [k for k in inc
return X[train_indices,:],X[test

def fit(X,y):
    shape_x = X.shape
    dim = shape_x[1]
    w = np.zeros(shape=(dim, 1)
    b = 0
    done = False
    while not done:
        done = True
        for i in range(shape_x[0]):
            x = X[i].reshape(1,-1)
            if y[i]*(x.dot(w)+b)<=0:
                w = w+y[i]*x.T
    return w, b

def predict(X,y,w,b):
    y_hat = np.sign(X.dot(w)+b)
    return sum(yi == yi_hat for yi,yi_hat in zip(y,y_hat))/len(y)

data = load_data()
X = data[:,2]
y = data[:,1]

X_train,X_test,y_train,y_test = data_split(X,y)
w,b = fit(X_train,y_train)
acc = predict(X_test,y_test,w,b)

print(w)
print(b)
print(acc)
```

提交

共有测试数据:1  
平均占用内存:6.570K 平均运行时间:0.07240S

测试数据 评判结果  
测试数据1 输出错误

详细评判结果 >>

基于代码评测实验环境的《图像处理》算法编程实验样例。

学生需要补全代码，提交评测。

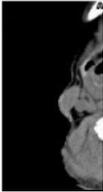
test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/programFillGapTest.jsp?problemID=24194

当前程序片段编程

【问题描述】 阈值判断Sobel

【输入形式】 图像文件

【输出形式】 图像文件

【样例输入】 

【样例输出】 

【样例说明】

【样例说明】

【评分标准】 程序通过编

```
private static I
int w=bi.
int h=bi.i
BufferedIi
//创建新程
//循环遍历
for(int y=
for(i
}
return nt
}

private static i
int s1=M.
int s2=M.

public
int y=s1+
int rgb=C
if(y>=yz'
rgb
}
return rg
}

}

privat

}

/**
 * 得到原图(x
 * @param B
 */
```

```
for(int i=(x-1);i<=(x+1);i++){
Color c=new Color(bi.getRGB(i, j));
switch (key) {
case 0:
sz[n]=(int) (0.3*c.getRed()+0.59*c.getGreen()+0.11*c.getBlue());
break;
case 1:
sz[n]=c.getRed();
break;
case 2:
sz[n]=c.getGreen();
break;
case 3:
sz[n]=c.getBlue();
break;
default:
break;
}
n++;
}
}
return sz;
}
```

提交

下载源文件 在线浏览源代码 更新时间: 2020-03-16 16:55:15

共有测试数据:1  
平均占用内存:48.457K 平均运行时间:0.43991S

测试数据 评判结果  
测试数据1 完全正确

基于通用评测实验环境的监督式机器学习编程实验样例。

学生提交代码后，平台按照代码在测试集上的准确度动态排名。

深度学习第一次作业 / 通用评测题 / 1. MNIST手写数字识别

### 【题目描述】

MNIST数据集及其相关介绍的地址为：<http://yann.lecun.com/ex>

该数据集的训练集包含55000个样本，测试集包含10000个样本。

在本题目中，对MNIST数据集进行了处理。每个样本为一张28像素为BMP，图片中每个像素的像素值为0或者255。

### 【评分标准】

在程序执行的当前目录下，包含一个 train 目录和一个 test 目录，系统会根据你提交程序输出的result.csv文件，计算你的模型在测试集上的准确率。

- train 目录为训练集，包含训练集图片，标签为数字。
- test 目录为测试集，包含测试集图片。

请你基于机器学习或者深度学习对图片进行分类。

### 【程序输出】

你的程序需要在当前目录下生成一个result.csv文件，该文件是由多行构成的，其中，预测的分类值的列表。

### 注意：

- 如果你提交了多个

- 如果你提交了多个Python文件，那么其中必须有一个文件命名为main.py作为程序的入口。
- 可以采用GPU训练模型，但需要注意训练时间，程序的最长运行时间为180秒（3分钟），超过后会被强行终止。
- 操作系统版本为：Ubuntu 18.04.1 TLS，CUDA驱动版本为：CUDA 10.0，GPU型号为：Nvidia GeForce GTX 1080Ti，GPU数量：1
- Python版本为：3.6.7，Keras版本为：2.3.0，TensorFlow版本为：1.14.0，PyTorch版本为：1.0.0，MXNet版本为：1.3.1

深度学习第一次作业 / MNIST手写数字识别

## 排行榜

#	学号	姓名	rank
1	KAGGLE_001	KAGGLE_001	0.98
2	KAGGLE_004	KAGGLE_004	0.98
3	KAGGLE_005	KAGGLE_005	0.98
4	KAGGLE_002	KAGGLE_002	0.97
5	KAGGLE_003	KAGGLE_003	0.91

基于通用评测实验环境的监督式机器学习编程实验样例。

学生提交代码后，平台按照代码在测试集上的准确度动态排名。

[深度学习第一次作业](#) / [通用评测题](#) / 1. MNIST手写数字识别

### 【题目描述】

MNIST数据集及其相关介绍的地址为：<http://yann.lecun.com/lex>

该数据集的训练集包含55000个样本，测试集包含10000个样本。

在本题目中，对MNIST数据集进行了处理。每个样本为一张28像素为BMP，图片中每个像素的像素值为0或者255。

### 【评分标准】

在程序执行的当前目录下，包含一个 train 目录和一个 test 目录，系统会根据你提交程序输出的result.csv文件，计算你的模型在测试集上的准确率。

- train 目录为训练集，包含训练集文件，如 train\_images.idx3\_ubyte, 标签为 train\_labels.idx1\_ubyte。
- test 目录为测试集，包含测试集文件，如 test\_images.idx3\_ubyte。

[深度学习第一次作业](#) / [MNIST手写数字识别](#)

请你基于机器学习或者深度学习对图片进行分类。

### 【程序输出】

你的程序需要在当前目录下生成一个result.csv文件，该文件是由多行构成的，其中，预测的分类值的列表。

### 注意：

- 如果你提交了多个

- 如果你提交了多个Python文件，那么其中必须有一个文件命名为main.py作为程序的入口。
- 可以采用GPU训练模型，但需要注意训练时间，程序的最长运行时间为180秒（3分钟），超过后会被强行终止。
- 操作系统版本为：Ubuntu 18.04.1 TLS，CUDA驱动版本为：CUDA 10.0，GPU型号为：Nvidia GeForce GTX 1080Ti，GPU数量：1
- Python版本为：3.6.7，Keras版本为：2.3.0，TensorFlow版本为：1.14.0，PyTorch版本为：1.0.0，MXNet版本为：1.3.1

## 排行榜

#	学号	姓名	rank
1	KAGGLE_001	KAGGLE_001	0.98
2	KAGGLE_004	KAGGLE_004	0.98
3	KAGGLE_005	KAGGLE_005	0.98
4	KAGGLE_002	KAGGLE_002	0.97
5	KAGGLE_003	KAGGLE_003	0.91

基于通用评测实验环境的GPU CUDA并行编程实验样例。

本实验要求学生使用CUDA C从零实现手写数字识别神经网络，要求该神经网络支持使用GPU加速训练过程。

学生提交代码后，平台按照代码在GPU上的执行效率动态排名。

test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest.jsp?problemID=33597

当前通用评测题: 3359; 紧接着为N行训练数据的标签, 标签采用One-hot的格式, 每行有10个浮点数, 仅有一个为1, 其他全为0。

【题目描述】 随后为一个正整数M, 表示测试数据数量。

test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest.jsp?problemID=33597

输入数据的第一行为一个正整数N, 表示训练数据数量。  
接下来是N行训练数据, 每行784个浮点数, 表示一副手写数字图像。  
紧接着为N行训练数据的标签, 标签采用One-hot的格式, 每行有10个浮点数, 仅有一个为1, 其他全为0。  
随后为一个正整数M, 表示测试数据数量。  
接下来M行为测试数据, 格式与训练数据相同。

【输出格式】  
请按照输入的顺序输出您对于每个测试数据的预测值, 输出数据应

提交源文件, 只能提交以 `cu` 为后缀的文件:  
 未选择任何文件

下载源文件 [在线浏览源代码](#)

Accept

name	verdict	score	time
TestCase1	Accept	9.8	34288

```
void mat_map(DTTYPE *A, int xa, int ya, DTTYPE *B, int xb, int yb, DTTYPE *C, int xc, int yc,
void (*kernel)(DTTYPE *A, int xa, int ya, DTTYPE *B, int xb, int yb, DTTYPE *C, int xc, int yc)) {
    DTTYPE *dA, *dB=NULL, *dC;
    int sA = xa * ya * sizeof(DTTYPE);
    int sB = xb * yb * sizeof(DTTYPE);
    int sC = xc * yc * sizeof(DTTYPE);
    cudaMalloc((void **)&dA, sA);
    cudaMemcpy(dA, A, sA, cudaMemcpyHostToDevice);

    if (B != NULL) {
        cudaMalloc((void **)&dB, sB);
        cudaMemcpy(dB, B, sB, cudaMemcpyHostToDevice);
    }

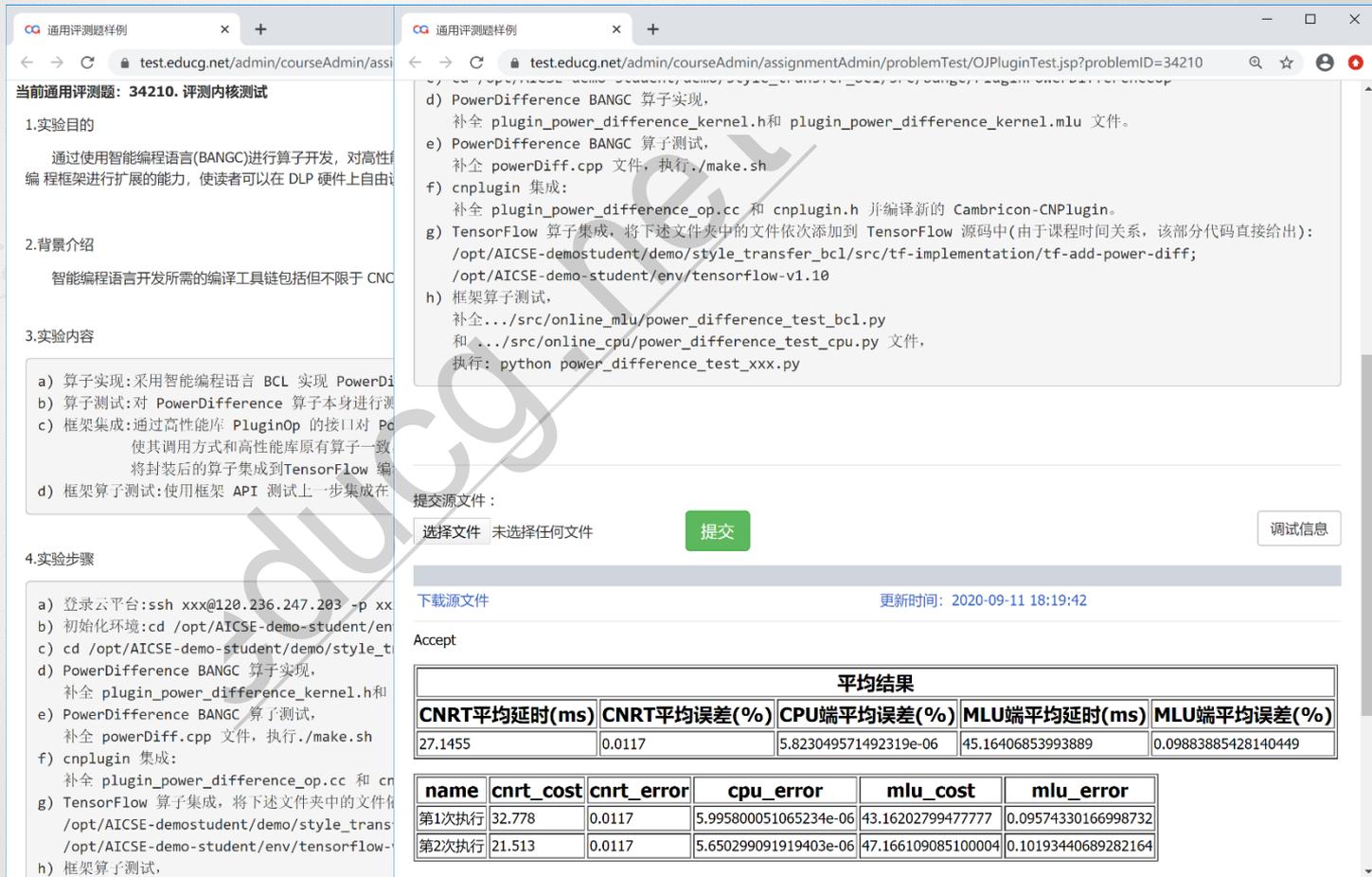
    if (A != C) {
        cudaMalloc((void **)&dC, sC);
    } else {
        dC = dA;
    }

    int threadPerBlock = THREAD_PREBLOCK;
    int blockPerGrid = (xc * yc + threadPerBlock - 1)/threadPerBlock;
    kernel <<< blockPerGrid, threadPerBlock >>> (dA, xa, ya, dB, xb, yb, dC, xc, yc);
    if (A != C) {
        cudaFree(dC);
    }
    cudaFree(dB);
    cudaFree(dA);
}
```

## 基于通用评测实验环境的“寒武纪”智能处理器(MLU)编程实验样例。

本实验要求学生使用智能编程语言(BANGC)进行算子开发,并最终将算子集成到编程框架(TensorFlow)中。算子在MLU上执行推理计算时,要求功能正确、性能高效、错误率低。

学生提交代码后,平台按照代码在CPU和MLU上的误差和延时动态排名。



通用评测题样例

test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest.jsp?problemID=34210

当前通用评测题: 34210. 评测内核测试

### 1.实验目的

通过使用智能编程语言(BANGC)进行算子开发,对高性能编程框架进行扩展的能力,使读者可以在 DLP 硬件上自由设计算子。

### 2.背景介绍

智能编程语言开发所需的编译工具链包括但不限于 CNC

### 3.实验内容

- a) 算子实现:采用智能编程语言 BCL 实现 PowerDifference
- b) 算子测试:对 PowerDifference 算子本身进行测试
- c) 框架集成:通过高性能库 PluginOp 的接口对 PowerDifference 使其调用方式和高性能库原有算子一致,将封装后的算子集成到TensorFlow 编程框架中
- d) 框架算子测试:使用框架 API 测试上一步集成在 TensorFlow 中的算子

### 4.实验步骤

- a) 登录云平台:ssh xxx@120.236.247.203 -p xxx
- b) 初始化环境:cd /opt/AICSE-demo-student/env
- c) cd /opt/AICSE-demo-student/demo/style\_transfer
- d) PowerDifference BANGC 算子实现, 补全 plugin\_power\_difference\_kernel.h 和 plugin\_power\_difference\_kernel.mlu 文件。
- e) PowerDifference BANGC 算子测试, 补全 powerDiff.cpp 文件, 执行./make.sh
- f) cnplugin 集成: 补全 plugin\_power\_difference\_op.cc 和 cnplugin.h 并编译新的 Cambricon-CNPlugin。
- g) TensorFlow 算子集成, 将下述文件夹中的文件依次添加到 TensorFlow 源码中(由于课程时间关系,该部分代码直接给出): /opt/AICSE-demo-student/demo/style\_transfer\_bcl/src/tf-implementation/tf-add-power-diff; /opt/AICSE-demo-student/env/tensorflow-v1.10
- h) 框架算子测试, 补全../src/online\_mlu/power\_difference\_test\_bcl.py 和 ../src/online\_cpu/power\_difference\_test\_cpu.py 文件, 执行: python power\_difference\_test\_XXX.py

提交源文件:

未选择任何文件

下载源文件 更新时间: 2020-09-11 18:19:42

Accept

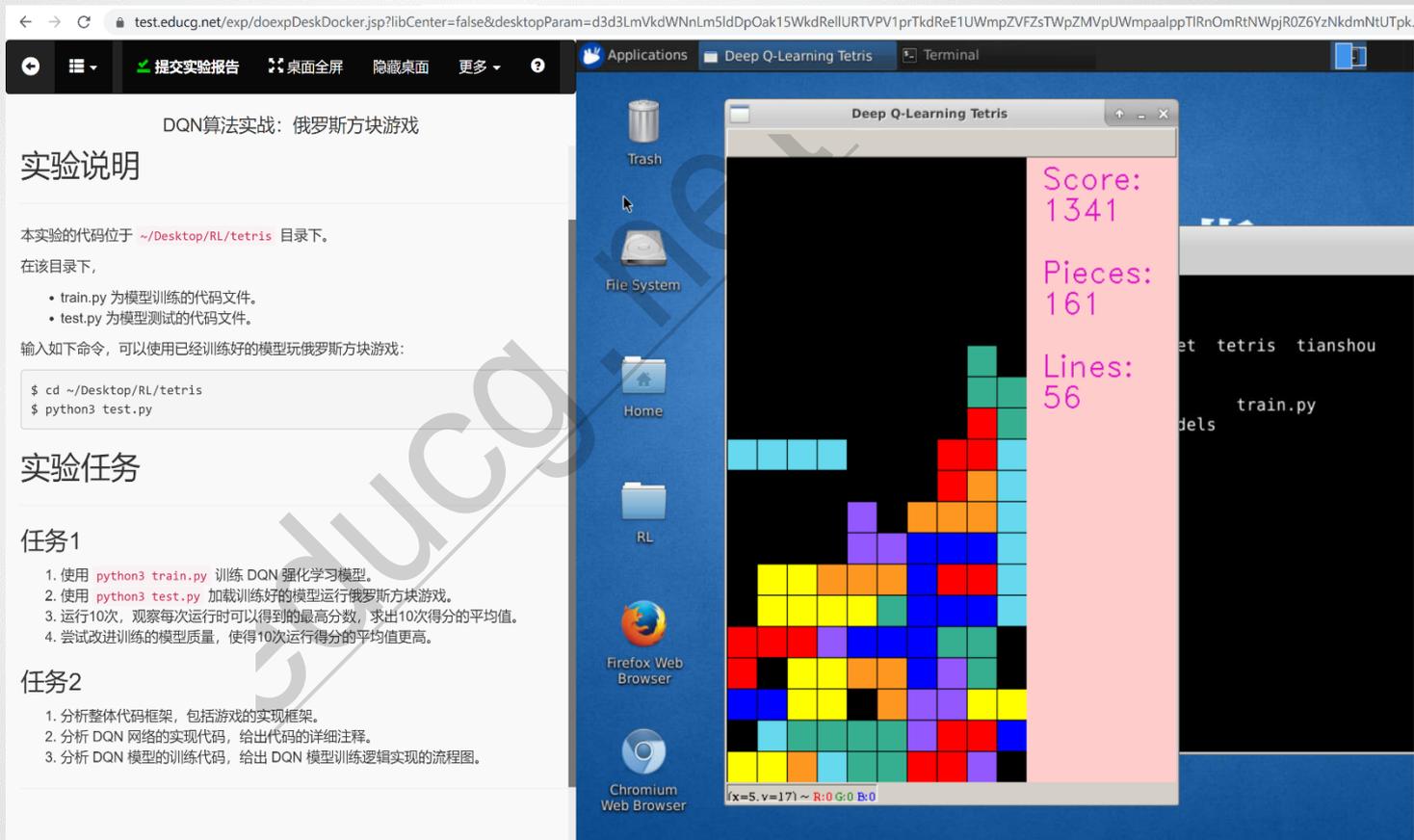
平均结果				
CNRT平均延时(ms)	CNRT平均误差(%)	CPU端平均误差(%)	MLU端平均延时(ms)	MLU端平均误差(%)
27.1455	0.0117	5.823049571492319e-06	45.16406853993889	0.09883885428140449

name	cnrt_cost	cnrt_error	cpu_error	mlu_cost	mlu_error
第1次执行	32.778	0.0117	5.995800051065234e-06	43.16202799477777	0.09574330166998732
第2次执行	21.513	0.0117	5.650299091919403e-06	47.166109085100004	0.10193440689282164

基于云桌面在线实验环境的《强化学习》课程DQN算法实验样例。

本实验要求学生基于DQN算法训练可以自动玩俄罗斯方块游戏的模型。

本实验为分析型实验，学生需要阅读已经实现的代码，分析实现原理并尝试优化改进。



The screenshot displays a cloud desktop environment. On the left, a web browser window shows the 'DQN算法实战：俄罗斯方块游戏' (DQN Algorithm Practical: Tetris Game) page. The page includes an '实验说明' (Experiment Description) section with instructions on where the code is located and how to run it. Below that is the '实验任务' (Experiment Tasks) section, which lists three tasks: training the model, observing performance, and optimizing the model. On the right, a terminal window titled 'Deep Q-Learning Tetris' shows the game in progress. The game board is a 10x10 grid with various colored blocks. A pink overlay on the right side of the terminal displays the game statistics: 'Score: 1341', 'Pieces: 161', and 'Lines: 56'. The terminal also shows the command prompt with the current directory and the command used to run the game.

test.educg.net/exp/doexpDeskDocker.jsp?libCenter=false&desktopParam=d3d3LmVkdWNnLm5ldDpOak15WkdRellURTVPV1prTkdReE1UWmpZVFZsTWpZMVpUWmpaalppTIRnOmRtNWpjR0Z6YzNkdMntUTpk.

提交实验报告 桌面全屏 隐藏桌面 更多

### DQN算法实战：俄罗斯方块游戏

#### 实验说明

本实验的代码位于 `~/Desktop/RL/tetris` 目录下。

在该目录下，

- `train.py` 为模型训练的代码文件。
- `test.py` 为模型测试的代码文件。

输入如下命令，可以使用已经训练好的模型玩俄罗斯方块游戏：

```
$ cd ~/Desktop/RL/tetris
$ python3 test.py
```

#### 实验任务

##### 任务1

1. 使用 `python3 train.py` 训练 DQN 强化学习模型。
2. 使用 `python3 test.py` 加载训练好的模型运行俄罗斯方块游戏。
3. 运行 10 次，观察每次运行时可以得到的最高分数，求出 10 次得分的平均值。
4. 尝试改进训练的模型质量，使得 10 次运行得分的平均值更高。

##### 任务2

1. 分析整体代码框架，包括游戏的实现框架。
2. 分析 DQN 网络的实现代码，给出代码的详细注释。
3. 分析 DQN 模型的训练代码，给出 DQN 模型训练逻辑实现的流程图。

Score: 1341  
Pieces: 161  
Lines: 56

```
et tetris tianshou
train.py
dels
```

ix=5, y=171 ~ R:0 G:0 B:0

基于云桌面在线实验环境的《人脸识别》课程表情识别实验样例。

本实验要求学生在云桌面实验环境中实时识别手机摄像头拍摄的视频中的人脸及人脸的表情。

学生最终实现的代码可以对云桌面中捕获的手机摄像头画面进行实时人脸识别和表情识别。



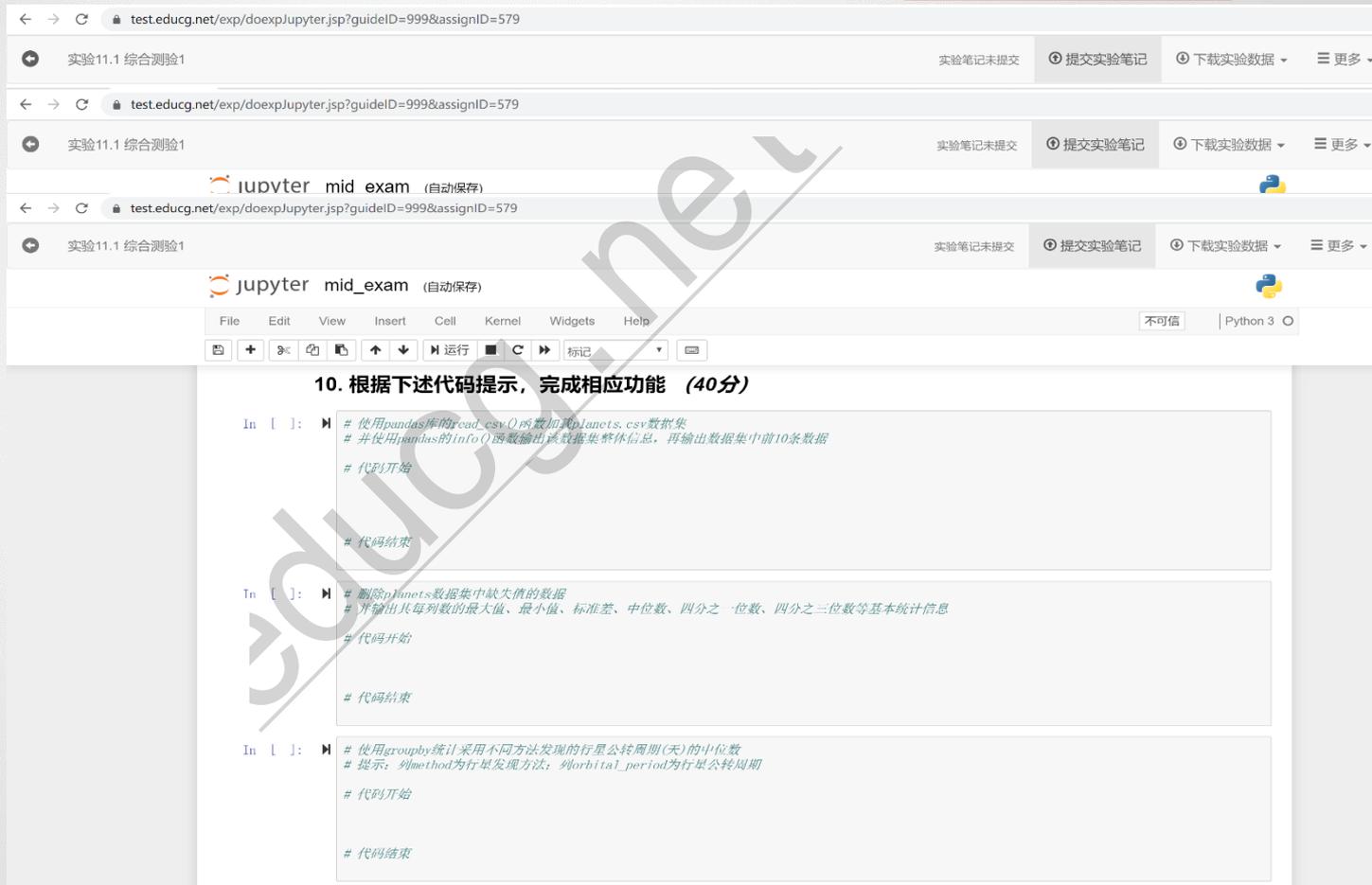
The screenshot displays a cloud desktop interface. On the left, a browser window shows a page titled "What is ray tracing?" with a diagram illustrating the ray tracing process. The diagram includes a "Camera" on the left, an "Image" plane, a "Light Source" on the right, and a "Scene Object" (a sphere) in the center. Red arrows represent "View Rays" from the camera to the object, and blue arrows represent "Shadow Rays" from the light source to the object. Below the diagram, text explains that ray tracing allows for more realistic shadows and reflections. On the right, a terminal window is open, showing the command `python3 test1.py` being executed. The desktop background features the "希冀 Course Grading" logo and the text "Right here waiting for you" and "在这里等着你". The taskbar at the bottom shows various application icons and the system clock indicating 2020/3/14.

基于B/S架构，真正实现“任何人、任何时间、任何地点”开展实验

基于Jupyter在线实验环境的《Python数据分析》课程的综合测验实验样例。

本实验要求学生根据提示编写代码完成测验。

该实验包含一系列实验练习任务。每个练习任务对应一定的得分。



The screenshot displays a JupyterLab environment with a quiz question. The browser address bar shows the URL: `test.educg.net/exp/doexp/jupyter.jsp?guidelD=999&assignID=579`. The page title is "实验11.1 综合测验1". The JupyterLab interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The main content area shows a code cell with the following text:

**10. 根据下述代码提示，完成相应功能 (40分)**

```
In [ ]: # 使用pandas的read_csv()函数加载planets.csv数据集
# 并使用pandas的info()函数输出该数据集整体信息，再输出数据集中前10条数据
# 代码开始

# 代码结束

In [ ]: # 删除planets数据集中缺失值的数据
# 并输出其每列数的最大值、最小值、标准差、中位数、四分之一位数、四分之三位数等基本统计信息
# 代码开始

# 代码结束

In [ ]: # 使用groupby统计采用不同方法发现的行星公转周期(天)的中位数
# 提示：列method为行星发现方法；列orbital_period为行星公转周期

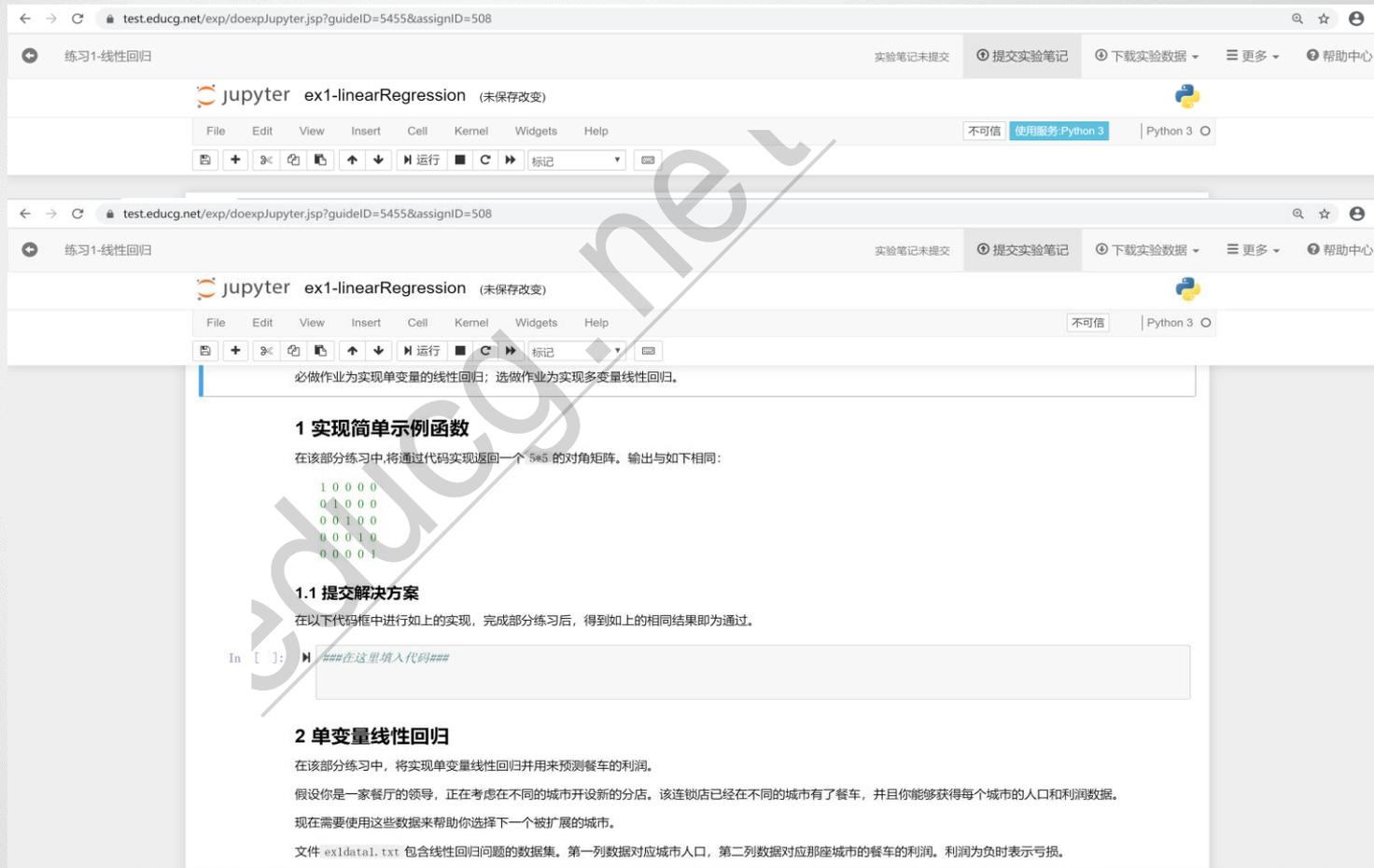
# 代码开始

# 代码结束
```

基于Jupyter在线实验环境的《机器学习》课程线性回归算法实验样例。

本实验要求学生基于提供的数据集自己动手实现线性回归算法。

该实验包含6个实验练习任务。每个练习任务对应一定的得分。



test.educg.net/exp/doexpJupyter.jsp?guidelD=5455&assignID=508

练习1-线性回归

jupyter ex1-linearRegression (未保存改变)

File Edit View Insert Cell Kernel Widgets Help 不可信 使用服务:Python 3 Python 3

test.educg.net/exp/doexpJupyter.jsp?guidelD=5455&assignID=508

练习1-线性回归

jupyter ex1-linearRegression (未保存改变)

File Edit View Insert Cell Kernel Widgets Help 不可信 Python 3

必做作业为实现单变量的线性回归；选做作业为实现多变量线性回归。

### 1 实现简单示例函数

在该部分练习中,将通过代码实现返回一个 5x5 的对角矩阵。输出与如下相同:

```
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

#### 1.1 提交解决方案

在以下代码框中进行如上的实现,完成部分练习后,得到如上的相同结果即为通过。

```
In [ ]: ###在这里填入代码###
```

### 2 单变量线性回归

在该部分练习中,将实现单变量线性回归并用来自预测餐车的利润。

假设你是一家餐厅的领导,正在考虑在不同的城市开设新的分店。该连锁店已经在不同的城市有了餐车,并且你能够获得每个城市的人口和利润数据。

现在需要使用这些数据来帮助你选择一个被扩展的城市。

文件 ex1data1.txt 包含线性回归问题的数据集。第一列数据对应城市人口,第二列数据对应那座城市的餐车的利润。利润为负时表示亏损。

基于Jupyter在线实验环境的《深度学习》课程线性回归算法实验样例。

本实验通过让学生学习如何从零开始实现CNN的批量归一化进而加深对CNN的原理解。

该实验包含6个实验练习任务。每个练习任务对应一定的得分。



test.educg.net/exp/doexpJupyter.jsp?guidelD=4800&assignID=580

批量归一化-GPU-练习

实验笔记未提交 提交实验笔记 下载实验数据 更多

test.educg.net/exp/doexpJupyter.jsp?guidelD=4800&assignID=580

批量归一化-GPU-练习

实验笔记未提交 提交实验笔记 下载实验数据 更多

test.educg.net/exp/doexpJupyter.jsp?guidelD=4800&assignID=580

批量归一化-GPU-练习

实验笔记未提交 提交实验笔记 下载实验数据 更多

jupyter batch-norm (自动保存)

File Edit View Insert Cell Kernel Widgets Help 不可信 Python 3

运行 标记

### 练习

- 我们能否将批量归一化前的全连接仿射变换或卷积计算中的偏差参数去掉？为什么？（提示：回忆批量归一化中标准化的定义。）

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 尝试调大学习率。跟前面未使用批量归一化的 LeNet 相比, 现在是不是可以使用更大的学习率？

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 尝试将批量归一化层插入到 LeNet 的其他地方, 观察并分析结果的变化。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 尝试不学习 beta 和 gamma (构造的时候加入参数 grad\_req='null' 来避免计算梯度), 观察并分析结果。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 查看 BatchNorm 类的文档来了解更多使用方法, 例如如何在训练时使用全局平均的均值和方差。

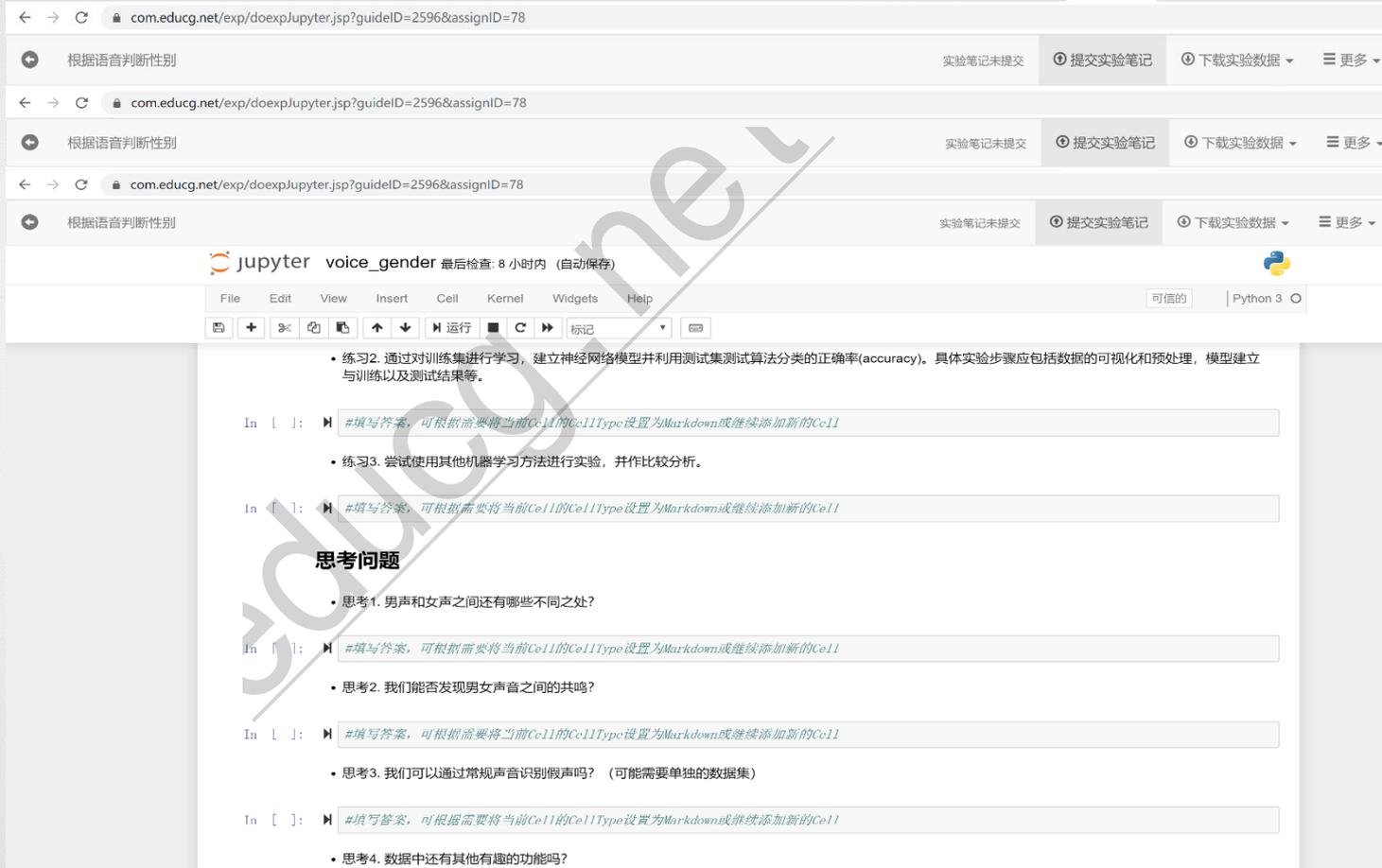
In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

### 参考文献

基于Jupyter在线实验环境的《语音识别》课程根据语音判断性别实验样例。

本实验要求学生基于提供的数据集训练一个可以根据语音区分性别的模型。

除了3个练习任务外，本实验还包含了4个思考题。



根据语音判断性别

实验笔记未提交 提交实验笔记 下载实验数据 更多

根据语音判断性别

实验笔记未提交 提交实验笔记 下载实验数据 更多

根据语音判断性别

实验笔记未提交 提交实验笔记 下载实验数据 更多

jupyter voice\_gender 最后检查: 8 小时内 (自动保存)

File Edit View Insert Cell Kernel Widgets Help 可信的 Python 3

- 练习2. 通过对训练集进行学习, 建立神经网络模型并利用测试集测试算法分类的正确率(accuracy)。具体实验步骤应包括数据的可视化和预处理, 模型建立与训练以及测试结果等。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 练习3. 尝试使用其他机器学习方法进行实验, 并作比较分析。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

### 思考问题

- 思考1. 男声和女声之间还有哪些不同之处?

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 思考2. 我们能否发现男女声音之间的共鸣?

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 思考3. 我们可以通过常规声音识别假声吗? (可能需要单独的数据集)

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 思考4. 数据中还有其他有趣的功能吗?

# 人工智能课程体系

## 理论、算法、应用、系统

- ◎ 理论基础类课程
- ◎ 应用基础类课程
- ◎ 应用能力类课程
- ◎ 系统能力类课程

理论基础	高等数学	矩阵分析	概率论与数理统计	最优化方法
	随机过程	贝叶斯分析	数学建模与仿真	复杂性科学
算法应用	Python数据分析	机器学习	深度学习	强化学习
	计算机视觉	自然语言处理	迁移学习	时间序列
	信号与系统	数字信号处理	数字图像处理	语音信号处理
	人工智能导论	知识工程	模式识别	生成式对抗网络
应用能力	语音识别	人脸识别	智能控制与机器人	智能应用系统综合设计
系统能力	神经网络算法性能优化	智能计算系统	深度学习框架设计	智能处理器设计

# 理论、算法、应用、系统

## 从理论到算法到应用再到系统的递进实验体系

- ◎ 公式推导及证明实验
- ◎ 算法应用实战型实验
- ◎ 算法从零实现实验
- ◎ 系统能力培养实验

由理论到算法到应用再到系统逐步递进的课程实验体系。

## 人工智能课程实验体系

- 理论
- 算法
- 应用
- 系统

## 理论算法应用系统

理论基础实验

算法基础实验

应用能力实验

系统能力实验

## 理论基础实验样例1

矩阵分类问题和感知机的有限性证明，来自《机器学习》课程。

学生需要使用LaTeX给出数学证明过程。

test.educg.net/admin/courseAdmin/expAdmin/guides/expPreviewJupyter.jsp?guidelD=5577

实验1: 监督式学习算法基础 此处仅预览实验, 对实验指导书和环境的改动不会保存! 定制Jupyter实验, 请进 [云桌面工作台->Jupyter工作台](#)。

提交实验笔记

jupyter ch02-v3 最后检查: 2020年9月6日 (自动保存)

File Edit View Insert Cell Kernel Widgets Help

不可信 Python 3

运行 标记

### 2.6 矩形分类问题

在矩形分类问题中, 样本空间是 $\mathbb{R}^2$ , 即全体平面上的点 $x = (x_1, x_2)$ , 标签由如下矩形函数 $f_{A,B,C,D} : \mathbb{R}^2 \rightarrow \{0, 1\}$ 给出:

$$f_{A,B,C,D}(x_1, x_2) = \begin{cases} 1, & A \leq x_1 \leq B \text{ 并且 } C \leq x_2 \leq D \\ -1, & \text{其余情况} \end{cases}$$

可以看到在矩形 $\{(x_1, x_2) : A \leq x_1 \leq B \text{ 并且 } C \leq x_2 \leq D\}$ 内部的点带有+1标签, 其余的点带有-1标签, 如图2.18所示。

任意一个矩形函数都可以用4个实参数来表示。取定0-1损失函数, 并确定模型假设 $H = \{f_{A,B,C,D} : A \leq B, C \leq D\}$ 为全体矩形函数。

(1) 考察下面算法: 输出包含所有训练数据中带+1标签的点的矩形。请证明这一算法是一个经验损失最小化算法。

(2) 试证明: 当训练数据规模 $m$ 满足:

$$m \geq \frac{4 \log 4\delta}{\epsilon}$$

时, 上面 (1) 中的算法以至少 $1 - \delta$ 的概率输出一个期望损失不超过 $\epsilon$ 的模型。

(3) 请将前两个小题的结论推广到一般的 $d$ 维空间。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

### 2.7 感知器算法的有限性

本题要求分步证明: 如果正负采样可以被一条直线分离, 则感知器算法在有限步内一定会终止。给定训练数据 $S = (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$ 。

设 $y = \langle w^*, x \rangle + b$ 是一条分离直线, 并设感知器算法的第 $t$ 步得到的直线方程为:  $y = \langle w^{(t)}, x \rangle + b^{(t)}$ 。(1) 请证明 $\langle w^*, w^{(t+1)} \rangle \geq \langle w^*, w^{(t)} \rangle + 1$

(2) 根据 (1) 中的结论证明:  $\langle w^*, w^{(t+1)} \rangle \geq t$

(3) 设 $R = \max_{1 \leq i \leq m} \|x^{(i)}\|$ , 请证明 $\|w^{(t+1)}\|^2 \leq \|w^{(t)}\|^2 + R^2$

(4) 请根据 (3) 中的结论证明 $\|w^{(t+1)}\| \leq \sqrt{t}R$

(5) 设感知器算法运行到第 $T$ 步, 请结合以上结论, 证明

$$\frac{\langle w^*, w^{(T+1)} \rangle}{\|w^*\| \|w^{(T+1)}\|} \geq \frac{\sqrt{T}}{\|w^*\| R}$$

(6) 请根据上述 (5) 中的结论, 利用柯西不等式证明 $T \leq \|w^*\|^2 R^2$ , 换句话说, 感知器算法在 $T \leq \|w^*\|^2 R^2$ 步之后必然停止。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

## 理论基础实验样例2

卷积神经网络批量归一化的理论证明及习题，来自《深度学习》课程。

学生需要使用LaTeX给出相关习题的证明过程。



test.educg.net/exp/doexpJupyter.jsp?guidelD=4800&assignID=580

批量归一化-GPU-练习

实验笔记未提交 提交实验笔记 下载实验数据 更多

test.educg.net/exp/doexpJupyter.jsp?guidelD=4800&assignID=580

批量归一化-GPU-练习

实验笔记未提交 提交实验笔记 下载实验数据 更多

test.educg.net/exp/doexpJupyter.jsp?guidelD=4800&assignID=580

批量归一化-GPU-练习

实验笔记未提交 提交实验笔记 下载实验数据 更多

jupyter batch-norm (自动保存)

File Edit View Insert Cell Kernel Widgets Help Python 3

运行 标记

### 练习

- 我们能否将批量归一化前的全连接仿射变换或卷积计算中的偏差参数去掉？为什么？（提示：回忆批量归一化中标准化的定义。）

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 尝试调大学习率。跟前面未使用批量归一化的 LeNet 相比, 现在是不是可以使用更大的学习率？

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 尝试将批量归一化层插入到 LeNet 的其他地方, 观察并分析结果的变化。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 尝试下不学习  $\beta$  和  $\gamma$  (构造的时候加入参数 `grad_req='null'` 来避免计算梯度), 观察并分析结果。

In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 查看 `BatchNorm` 类的文档来了解更多使用方法, 例如如何在训练时使用全局平均的均值和方差。

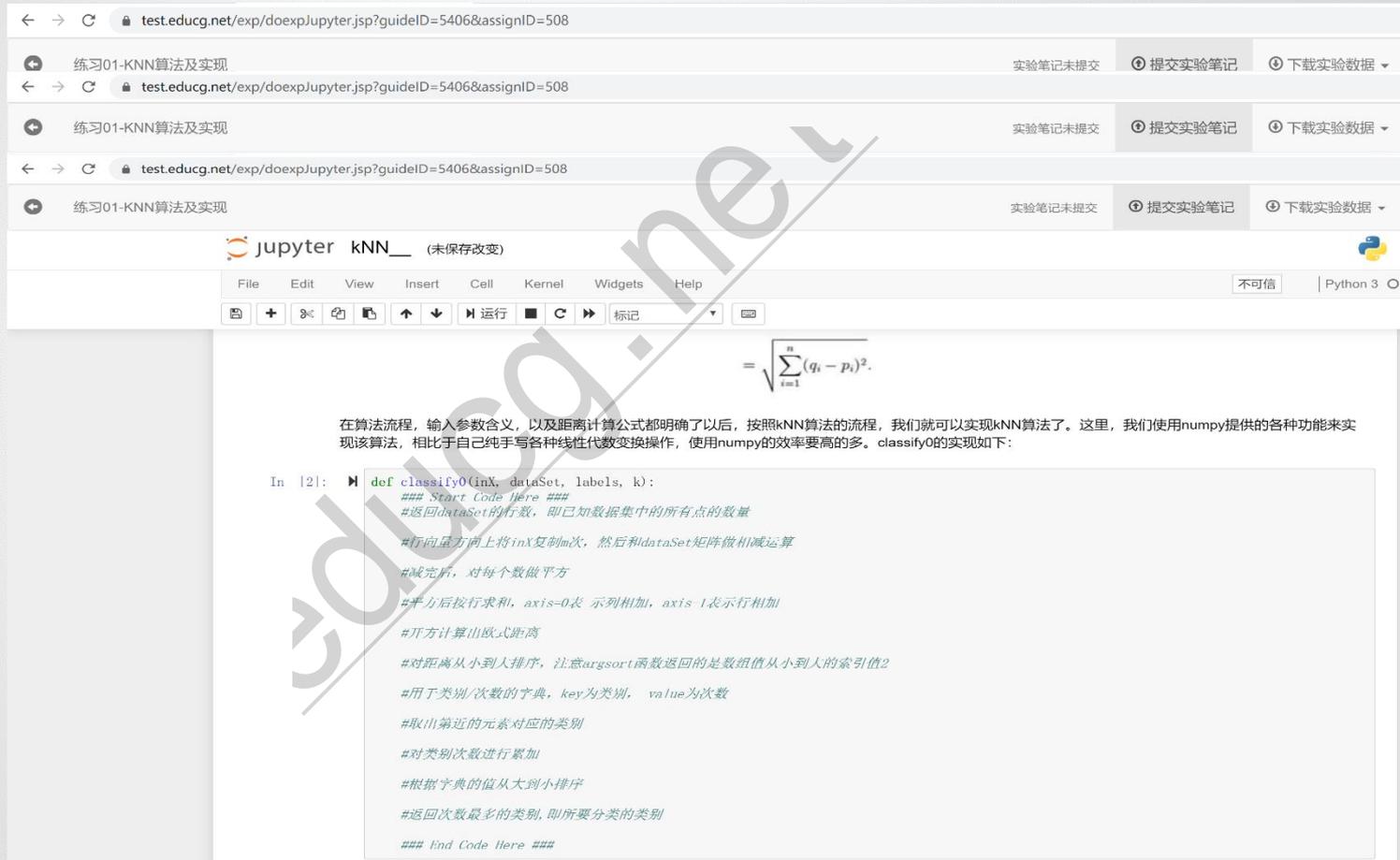
In [ ]: #填写答案, 可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

### 参考文献

## 算法基础实验样例1

KNN 算法实现实验，来自《机器学习》课程。

学生需要按照算法原理和代码注释给出实现算法。



test.educg.net/exp/doexpJupyter.jsp?guideID=5406&assignID=508

练习01-KNN算法及实现 实验笔记未提交 提交实验笔记 下载实验数据

练习01-KNN算法及实现 实验笔记未提交 提交实验笔记 下载实验数据

练习01-KNN算法及实现 实验笔记未提交 提交实验笔记 下载实验数据

Jupyter kNN\_\_ (未保存改变)

File Edit View Insert Cell Kernel Widgets Help Python 3

运行 标记

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

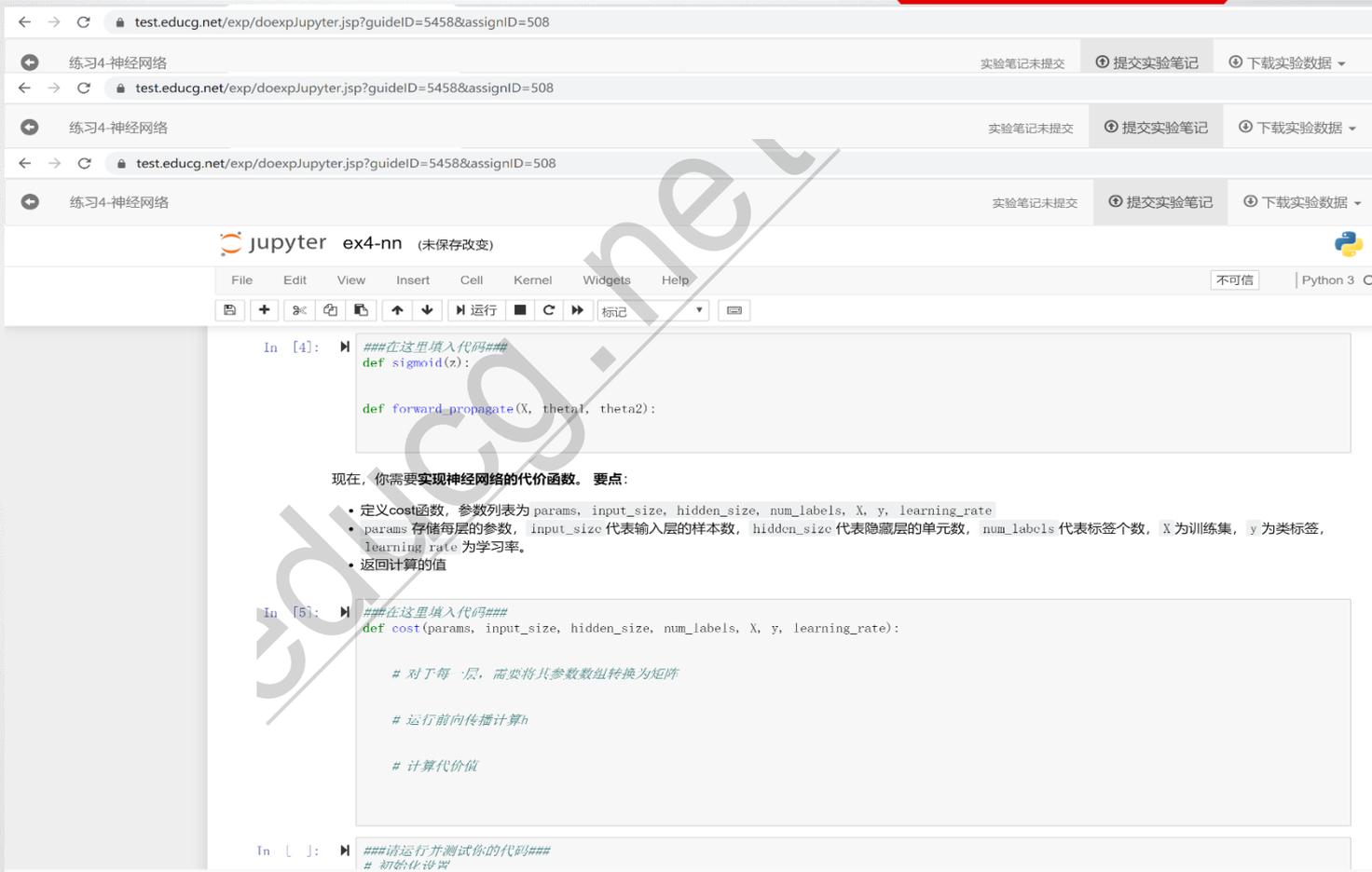
在算法流程，输入参数含义，以及距离计算公式都明确了以后，按照kNN算法的流程，我们就可以实现kNN算法了。这里，我们使用numpy提供的各种功能来实现该算法，相比于自己纯手写各种线性代数变换操作，使用numpy的效率要高的多。classify0的实现如下：

```
In [2]: def classify0(inX, dataSet, labels, k):
        """ Start Code Here """
        #返回dataSet的行数，即已知数据集中的所有点的数量
        #行向量方向上将inX复制m次，然后和dataSet矩阵做相减运算
        #减完后，对每个数做平方
        #平方后按行求和，axis=0表示列相加，axis=1表示行相加
        #开方计算出欧式距离
        #对距离从小到大排序，注意argsort函数返回的是数组值从小到大的索引值2
        #用于类别/次数的字典，key为类别，value为次数
        #取出最近的元素对应的类别
        #对类别次数进行累加
        #根据字典的值从大到小排序
        #返回次数最多的类别，即所要分类的类别
        """ End Code Here """
```

## 算法基础实验样例2

神经网络算法实现实验，来自《机器学习》课程。

学生需要按照算法原理和代码注释给出实现算法。



test.educg.net/exp/doexpJupyter.jsp?guidelD=5458&assignID=508

练习4-神经网络 实验笔记未提交 提交实验笔记 下载实验数据

test.educg.net/exp/doexpJupyter.jsp?guidelD=5458&assignID=508

练习4-神经网络 实验笔记未提交 提交实验笔记 下载实验数据

test.educg.net/exp/doexpJupyter.jsp?guidelD=5458&assignID=508

练习4-神经网络 实验笔记未提交 提交实验笔记 下载实验数据

jupyter ex4-nn (未保存改变)

File Edit View Insert Cell Kernel Widgets Help 不可信 Python 3

```
In [4]: ###在这里填入代码###
def sigmoid(z):

def forward_propagate(X, thetat, theta2):
```

现在，你需要实现神经网络的代价函数。要点：

- 定义cost函数，参数列表为 params, input\_size, hidden\_size, num\_labels, X, y, learning\_rate
- params 存储每层的参数，input\_size 代表输入层的样本数，hidden\_size 代表隐藏层的单元数，num\_labels 代表标签个数，X 为训练集，y 为类标签，learning\_rate 为学习率。
- 返回计算的值

```
In [5]: ###在这里填入代码###
def cost(params, input_size, hidden_size, num_labels, X, y, learning_rate):

# 对于每一层，需要将其参数数组转换为矩阵

# 运行前向传播计算h

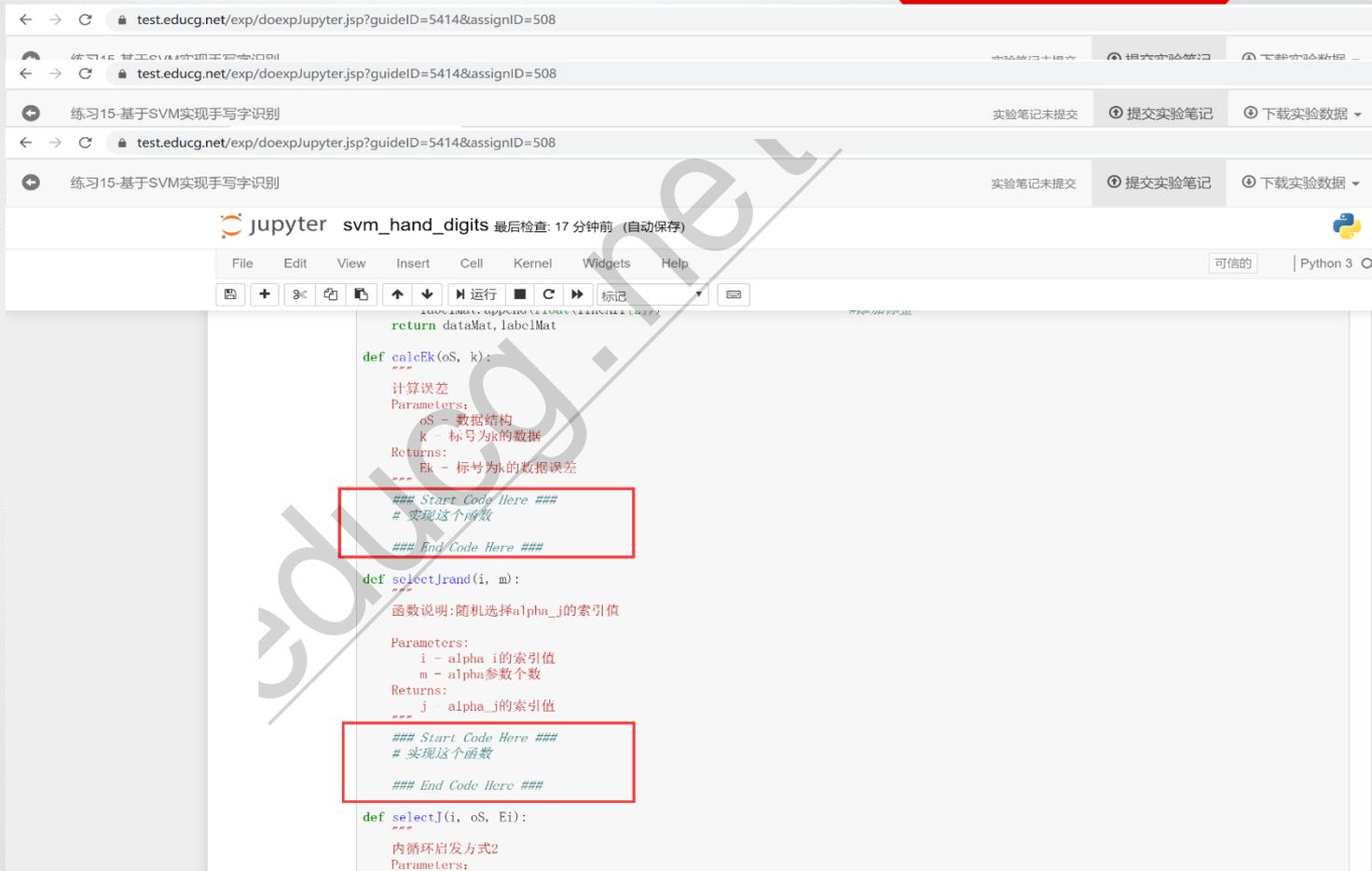
# 计算代价值
```

```
In [ ]: ###请运行并测试你的代码###
# 初始化设置
```

## 应用能力实验样例1

SVM算法应用实验，来自《机器学习》课程。

学生需要按照实验要求和代码注释编写SVM手写数字识别的实现代码。



```
return dataMat, labelMat

def calcEk(oS, k):
    """
    计算误差
    Parameters:
    oS - 数据结构
    k - 标号为k的数据
    Returns:
    Ek - 标号为k的数据误差
    """
    ### Start Code Here ###
    # 实现这个函数
    ### End Code Here ###

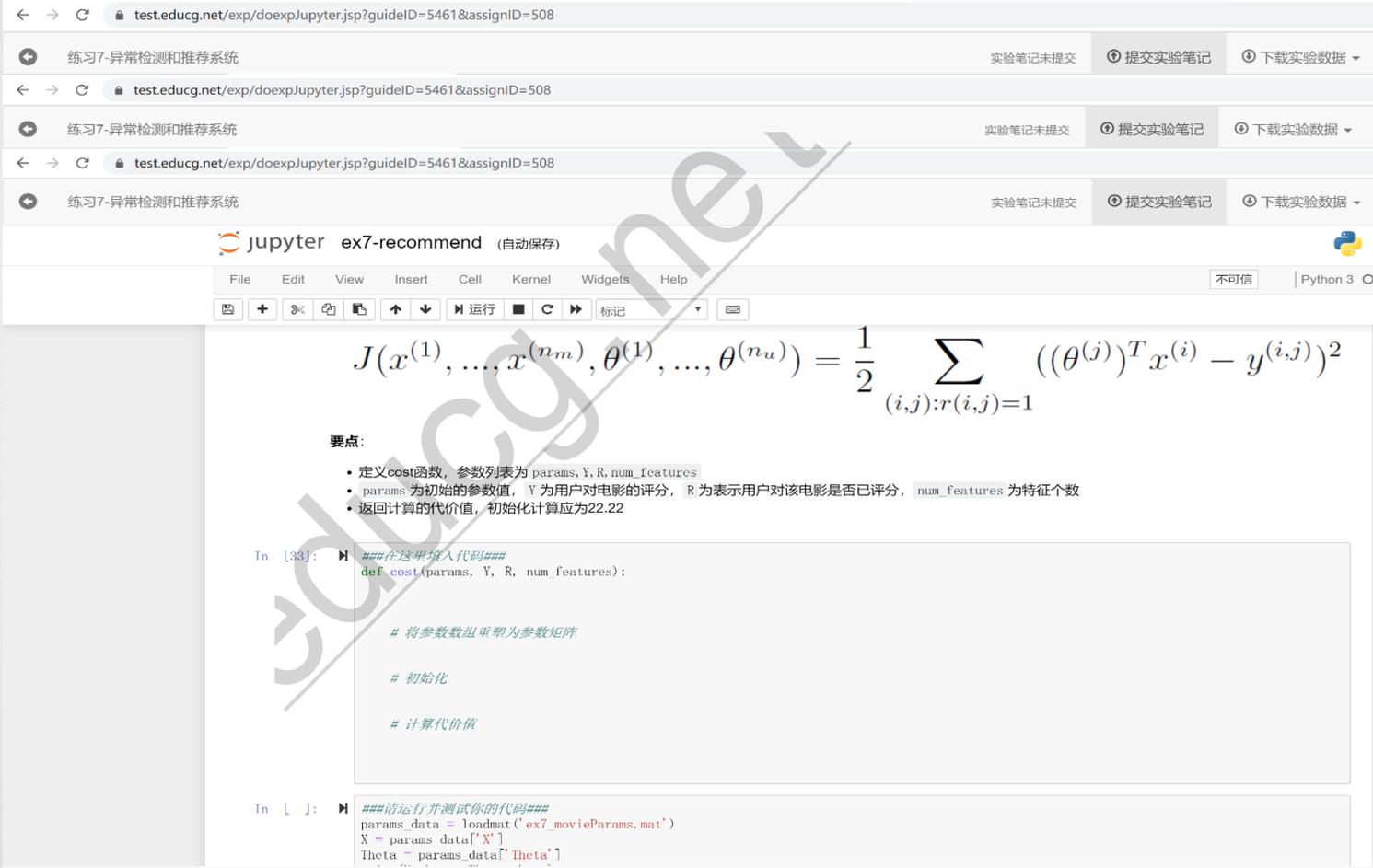
def selectJrand(i, m):
    """
    函数说明:随机选择alpha_j的索引值
    Parameters:
    i - alpha_i的索引值
    m - alpha参数个数
    Returns:
    j - alpha_j的索引值
    """
    ### Start Code Here ###
    # 实现这个函数
    ### End Code Here ###

def selectJ(i, oS, Ei):
    """
    内循环启发方式2
    Parameters:
```

## 应用能力实验样例2

异常检测和推荐系统实验，来自《机器学习》课程。

学生需要按照实验要求和代码注释编写电影推荐的实现代码。



The screenshot shows a Jupyter Notebook titled "ex7-recommend" with the following content:

URL: test.educg.net/exp/doexpJupyter.jsp?guideID=5461&assignID=508

Navigation: 练习7-异常检测和推荐系统

Buttons: 实验笔记未提交, 提交实验笔记, 下载实验数据

Equation:

$$J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2$$

要点:

- 定义cost函数，参数列表为 params, Y, R, num\_features
- params 为初始的参数值，Y 为用户对电影的评分，R 为表示用户对该电影是否已评分，num\_features 为特征个数
- 返回计算的代价值，初始化计算应为22.22

```
In [33]: ##在这里填入代码##
def cost(params, Y, R, num_features):

    # 将参数数组重塑为参数矩阵

    # 初始化

    # 计算代价值
```

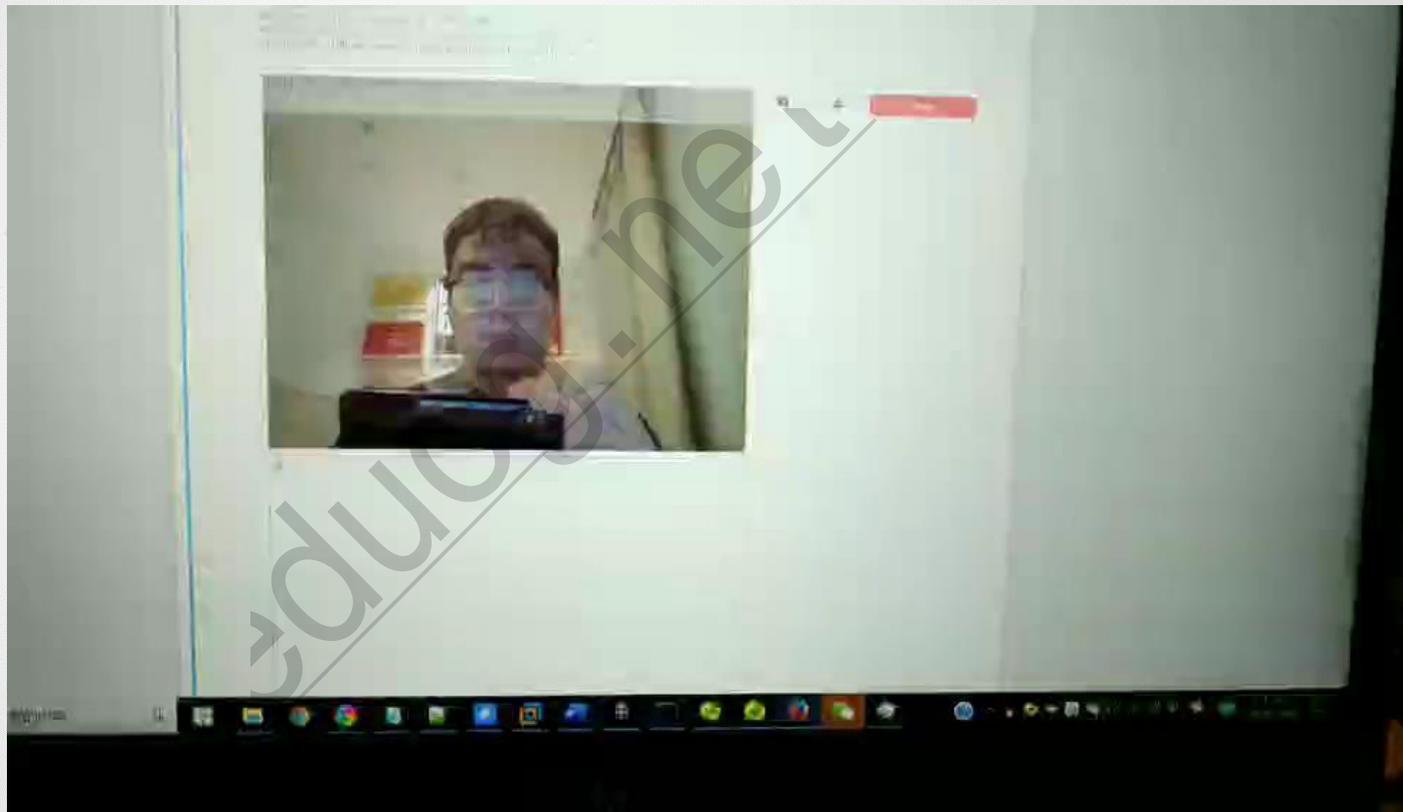
```
In [ ]: ##请运行并测试你的代码##
params_data = loadmat('ex7_movieParams.mat')
X = params_data['X']
Theta = params_data['Theta']
```

## 应用能力实验样例3

人脸识别及表情识别实验，来自《人脸识别》课程。

在本实验中，学生学习如何使用电脑摄像头进行人脸识别和表情识别。

本实验要求学生编写代码使得能够采用电脑摄像头实时识别更多的表情。

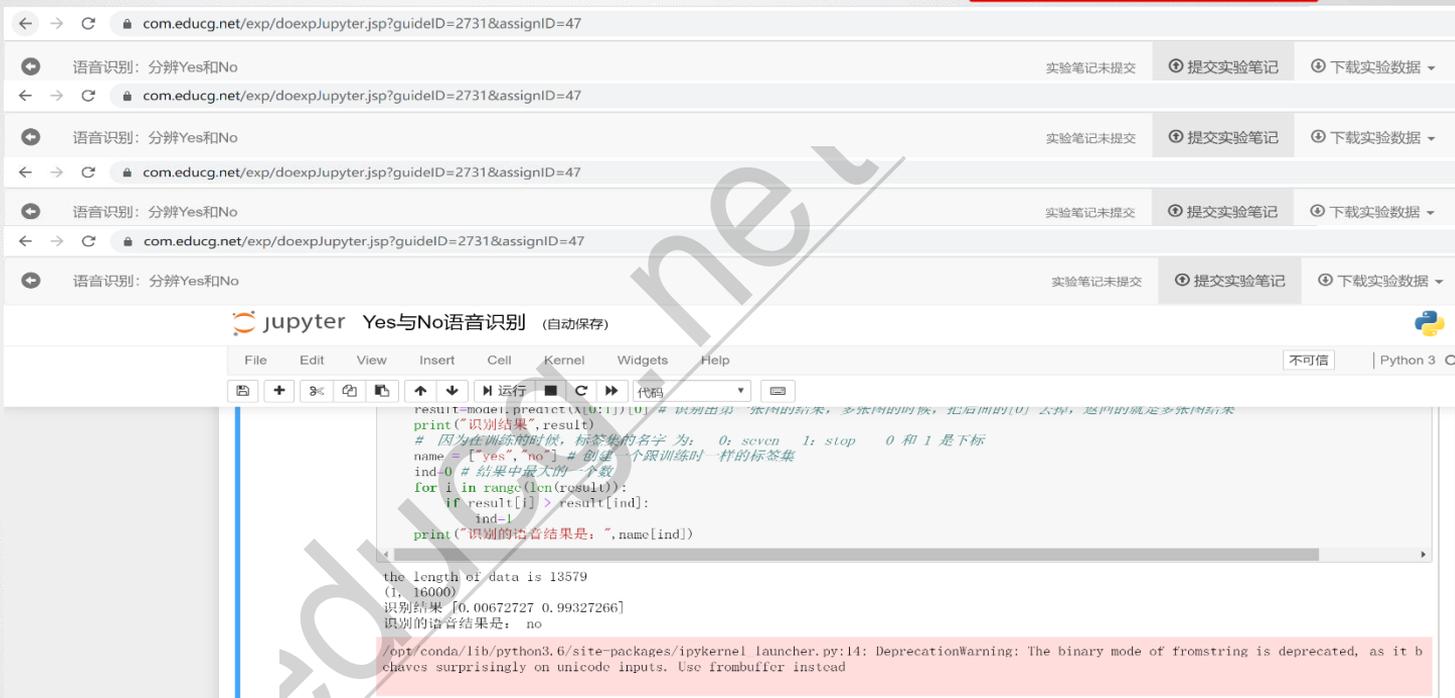


## 应用能力实验样例4

Yes/No单词语音识别实验，来自《语音识别》课程。

在本实验中，学生学习如何使用深度神经网络基于语音识别Yes/No两个单词。

本实验要求学生基于实验中给出的理论和代码，编写代码能够实现数字0~10的语音识别。



The screenshot shows a JupyterLab interface with a Python 3 kernel. The notebook title is 'jupyter Yes与No语音识别 (自动保存)'. The code in the cell is as follows:

```
result=model.predict(XU:1])U] # 识别出第一张图的结果，多张图的时候，把后面的U]去掉，选中的就是多张图的结果
print("识别结果",result)
# 因为在训练的时候，标签集的名字为： 0: seven 1: stop 0和1是下标
name=["yes","no"] # 创建一个跟训练时一样的标签集
ind=0 # 结果中最大的一个数
for i in range(len(result)):
    if result[i] > result[ind]:
        ind=i
print("识别的语音结果是：",name[ind])
```

The output shows:

```
the length of data is 13579
(1, 16000)
识别结果 [0.00672727 0.99327266]
识别的语音结果是： no
```

A red warning message is visible at the bottom of the code cell:

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:14: DeprecationWarning: The binary mode of fromstring is deprecated, as it b
chaves surprisingly on unicode inputs. Use frombuffer instead
```

### 实验任务

- 练习1. 现有模型判断yes或no的精确度为80%左右，采用其他模型提高其精度。

In [ ]: #填写答案，可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

- 练习2. 基于提供的数据集，实现对数字1~9的语音分类，并使用真实数据进行测试

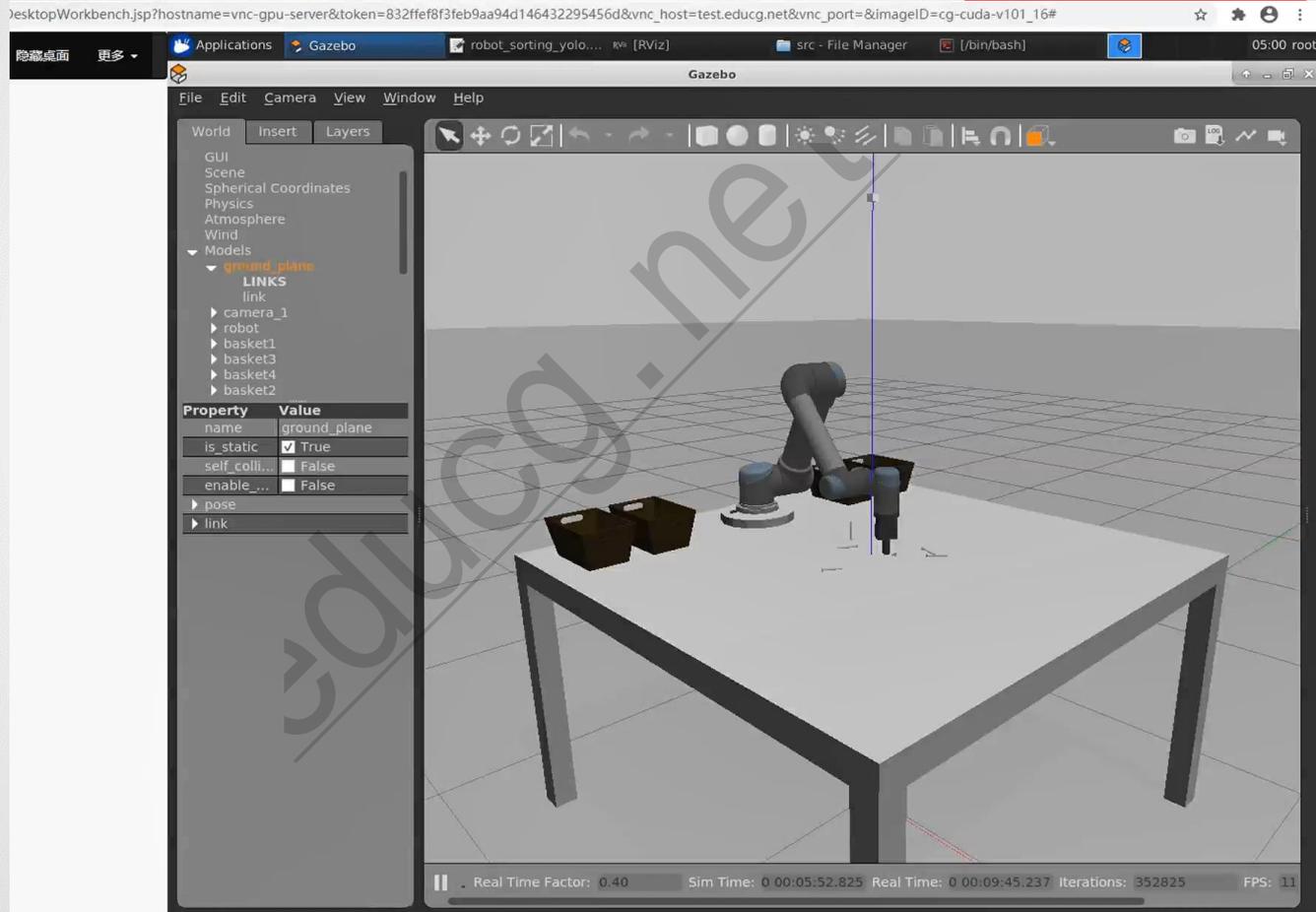
In [ ]: #填写答案，可根据需要将当前Cell的CellType设置为Markdown或继续添加新的Cell

## 应用能力实验样例5

机械臂智能抓取实验，来自《智能控制与机器人》课程。

在本实验中，学生学习如何使用YOLO v3计算机视觉模型完成桌面不同型号螺丝钉的智能分拣。

本实验要求学生桌面上按颜色随机和尺寸随机生成不同的立方体。使用机械臂将不同型号的立方体分拣到不同的盒子中。



## 应用能力实验样例6

移动机器人SLAM导航实验，来自《智能控制与机器人》课程。

在本实验中，学生学习如何基于SLAM算法使用远程移动机器人开展SLAM建图和导航实验。

本实验要求学生能够实现能够控制机器人自动巡游建图的程序。建立完毕的地图可正确地用于机器人导航。



## 应用能力实验样例7

OCR文字识别实验，来自《智能应用系统综合设计》课程。

在本实验中，学生需要按生产交付标准从零实现一个完整的OCR文字识别系统。

OCR文字识别系统要能够提供API服务，提供前端用户使用界面，支持Docker一键部署方案，支持基于K8S的弹性扩展。

实验专题	实验名称	实验任务	考核要求	学时	难度	镜像名称	实验环境
智能应用系统综合设计	数学公式拍照计算系统	自动计算机拍照图片中的数学公式的结果，支持微积分等复杂数学公式。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	纸质试卷自动评阅系统	对纸质试卷中的选择题、填空题、简答题三种题型进行自动评分。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	疲劳检测系统	基于笔记本摄像头拍摄的人脸视频判断是否疲劳。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	盲打能力测试系统	对测试人员的盲打能力进行测试，要求使用摄像头跟踪眼球。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	题库知识点自动分类系统	对题库中的大量题目自动标注知识点标签，要求达到一定的准确度。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	医疗问答专家系统	基于对话语料库实现自动医疗问答专家系统。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	银行客服机器人	基于对话语料库实现银行客服机器人。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	送餐机器人	在仿真酒店环境中，使用仿真机器人将特定餐盒送至指定的房间。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	云考试录屏视频反作弊分析	分析云考试过程中，学生本地上机的录屏视频中是否存在作弊现象。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
	PDF文档水印去除	给定一个带有水印的PDF文档，生成去除水印后的新PDF文档。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面
100年前晚清北京影像上色	对1920年拍摄的晚清北京黑白影像（10分钟）进行上色和高清化修复。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面	
水印去除	使用word生成带水印的pdf文件，去除pdf文件中的水印。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面	
普通照片三维化	将一张普通的照片三维化，可以呈现出立体化效果。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面	
OCR文字识别系统	按生产交付要求从零实现一个完整的OCR文字识别系统。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面	
聊天机器人	基于海量对话语料库实现聊天机器人系统，支持多人并发聊天，支持自学习。	掌握	24	5	cg/vnc/ai-gpu:v1.0.1	云桌面	

实验概览 / 编辑实验与步骤



OCR文字识别系统

OCR文字识别系统

实验编号：25558119277402

编辑实验介绍与环境

- 文字识别原理介绍 自动评测 分享 删除 向上
- 文字识别整体设计介绍 自动评测 分享 删除 向上
- 文字检测模型训练 自动评测 分享 删除 向上
- 文字识别模型训练 自动评测 分享 删除 向上
- API 服务** 自动评测 分享 删除 向上
- Docker 部署方案 自动评测 分享 删除 向上
- Docker 部署方案 自动评测 分享 删除 向上

+ 添加实验流程

## 系统能力实验样例1

神经网络GPU编程实验，来自《神经网络算法性能优化》课程。

学生根据要求使用CUDA C从零实现手写数字识别神经网络，要求该神经网络支持使用GPU加速训练过程。基于GPU实现的神经网络要达到特定的性能标准。



test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest.jsp?problemID=33597

当前通用评测题: 3359; 紧接着为N行训练数据的标签, 标签采用One-hot的格式, 每行有10个浮点数, 仅有一个为1, 其他全为0。

【题目描述】 随后为一个正整数M, 表示测试数据数量。

test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest.jsp?problemID=33597

输入数据的第一行为一个正整数N, 表示训练数据数量。

接下来是N行训练数据, 每行784个浮点数, 表示一副手写数字图像。

紧接着为N行训练数据的标签, 标签采用One-hot的格式, 每行有10个浮点数, 仅有一个为1, 其他全为0。

随后为一个正整数M, 表示测试数据数量。

接下来M行为测试数据, 格式与训练数据相同。

【输出格式】

请按照输入的顺序输出您每个测试数据的预测值, 输出数据应

提交源文件, 只能提交以 **cu** 为后缀的文件:

未选择任何文件

[下载源文件](#) [在线浏览源代码](#)

Accept

name	verdict	score	time
TestCase1	Accept	9.8	34288

```
void mat_map(DTYPE *A, int xa, int ya, DTYPE *B, int xb, int yb, DTYPE *C, int xc, int yc,
void (*kernel)(DTYPE *A, int xa, int ya, DTYPE *B, int xb, int yb, DTYPE *C, int xc, int yc)) {
DTYPE *dA, *dB=NULL, *dC;
int sA = xa * ya * sizeof(DTYPE);
int sB = xb * yb * sizeof(DTYPE);
int sC = xc * yc * sizeof(DTYPE);
cudaMalloc((void **)&dA, sA);
cudaMemcpy(dA, A, sA, cudaMemcpyHostToDevice);

if (B != NULL) {
cudaMalloc((void **)&dB, sB);
cudaMemcpy(dB, B, sB, cudaMemcpyHostToDevice);
}

if (A != C) {
cudaMalloc((void **)&dC, sC);
} else {
dC = dA;
}

int threadPerBlock = THREAD_PREBLOCK;
int blockPerGrid = (xc * yc + threadPerBlock - 1)/threadPerBlock;
kernel <<< blockPerGrid, threadPerBlock >>> (dA, xa, ya, dB, xb, yb, dC, xc, yc);
if (A != C) {
cudaFree(dC);
}
cudaFree(dB);
cudaFree(dA);
}
```

## 系统能力实验样例2

神经网络算子编程实验，来自《智能计算系统》课程。该课程由希冀和寒武纪联合开发。

学生根据要求实现 PowerDifference 算子，然后将该算子与 TensorFlow 集成，重新编译 TF 框架。新框架在 CPU 和 MLU 上运行时要达到特定的误差率和时延要求。

通用评测题样例

test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest.jsp?problemID=34210

当前通用评测题：34210. 评测内核测试

1.实验目的

通过使用智能编程语言(BANGC)进行算子开发，对高性能编程框架进行扩展的能力，使读者可以在 DLP 硬件上自由编程。

2.背景介绍

智能编程语言开发所需的编译工具链包括但不限于 CNC

3.实验内容

a) 算子实现:采用智能编程语言 BCL 实现 PowerDifference  
 b) 算子测试:对 PowerDifference 算子本身进行测试  
 c) 框架集成:通过高性能库 PluginOp 的接口对 Pc 使其调用方式和高性能库原有算子一致将封装后的算子集成到TensorFlow 框架  
 d) 框架算子测试:使用框架 API 测试上一步集成在

4.实验步骤

a) 登录云平台:ssh xxx@120.236.247.203 -p xxx  
 b) 初始化环境:cd /opt/AICSE-demo-student/env  
 c) cd /opt/AICSE-demo-student/demo/style\_transfer  
 d) PowerDifference BANGC 算子实现，  
 补全 plugin\_power\_difference\_kernel.h和  
 e) PowerDifference BANGC 算子测试，  
 补全 powerDiff.cpp 文件，执行./make.sh  
 f) cnplugin 集成：  
 补全 plugin\_power\_difference\_op.cc 和 cnplugin.h 并编译新的 Cambricon-CNPlugin。  
 g) TensorFlow 算子集成，将下述文件夹中的文件依次添加到 TensorFlow 源码中(由于课程时间关系，该部分代码直接给出)：  
 /opt/AICSE-demostudent/demo/style\_transfer\_bcl/src/tf-implementation/tf-add-power-diff;  
 /opt/AICSE-demo-student/env/tensorflow-v1.10  
 h) 框架算子测试，  
 补全.../src/online\_mlu/power\_difference\_test\_bcl.py  
 和 .../src/online\_cpu/power\_difference\_test\_cpu.py 文件，  
 执行：python power\_difference\_test\_xxx.py

通用评测题样例

test.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest.jsp?problemID=34210

d) PowerDifference BANGC 算子实现，  
 补全 plugin\_power\_difference\_kernel.h和 plugin\_power\_difference\_kernel.mlu 文件。  
 e) PowerDifference BANGC 算子测试，  
 补全 powerDiff.cpp 文件，执行./make.sh  
 f) cnplugin 集成：  
 补全 plugin\_power\_difference\_op.cc 和 cnplugin.h 并编译新的 Cambricon-CNPlugin。  
 g) TensorFlow 算子集成，将下述文件夹中的文件依次添加到 TensorFlow 源码中(由于课程时间关系，该部分代码直接给出)：  
 /opt/AICSE-demostudent/demo/style\_transfer\_bcl/src/tf-implementation/tf-add-power-diff;  
 /opt/AICSE-demo-student/env/tensorflow-v1.10  
 h) 框架算子测试，  
 补全.../src/online\_mlu/power\_difference\_test\_bcl.py  
 和 .../src/online\_cpu/power\_difference\_test\_cpu.py 文件，  
 执行：python power\_difference\_test\_xxx.py

提交源文件：

选择文件 未选择任何文件

下载源文件 更新时间：2020-09-11 18:19:42

Accept

平均结果				
CNRT平均延时(ms)	CNRT平均误差(%)	CPU端平均误差(%)	MLU端平均延时(ms)	MLU端平均误差(%)
27.1455	0.0117	5.823049571492319e-06	45.16406853993889	0.09883885428140449

name	cnrt_cost	cnrt_error	cpu_error	mlu_cost	mlu_error
第1次执行	32.778	0.0117	5.995800051065234e-06	43.16202799477777	0.09574330166998732
第2次执行	21.513	0.0117	5.650299091919403e-06	47.166109085100004	0.10193440689282164

## 系统能力实验样例3

自动微分框架设计实验，来自《深度学习框架设计》课程。

学生根据根据提供的一系列实验可从零开始自己动手实现一个自动微分框架。

要求该自动微分框架至少支持运行MLP模型。



实验概览 / 共 6 / 6 个实验

排序 ▾ | 远程桌面 | Jupyter实验 | 实验专题 ▾

批量导入实验 | 重置实验环境 | 更多 ▾ | 远程桌面工作台 | Jupyter工作台

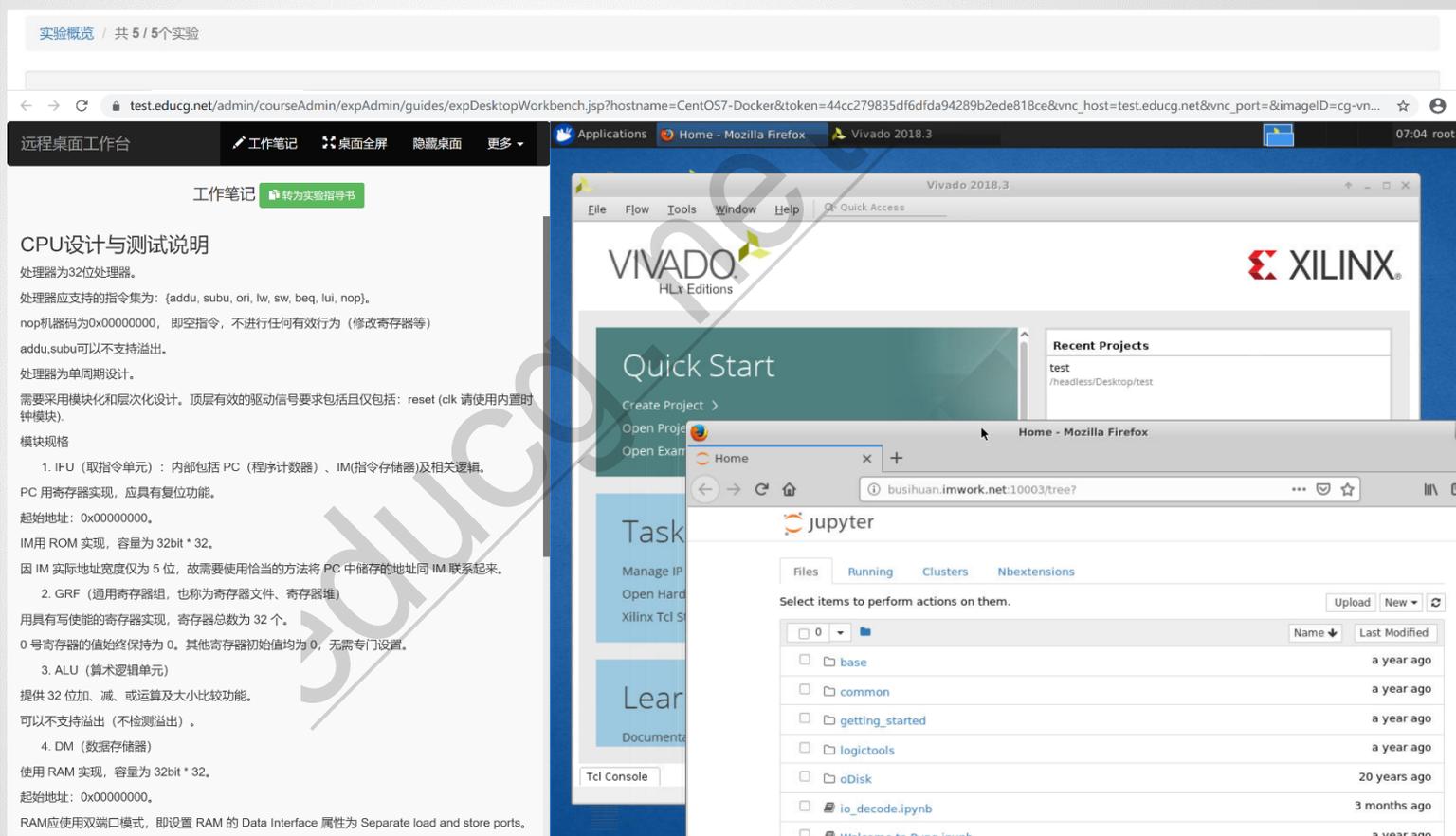
- 实验六：使用MLP验证自动微分框架
- 实验五：自动微分框架的整体实现
- 实验四：反向传播
- 实验三：前向传播
- 实验二：自动微分框架整体架构
- 实验一：神经网络基础（感知器）

## 系统能力实验样例4

智能处理器设计实验，来自《智能处理器》课程。

学生根据根据提供的一系列实验从零开始自己动手使用HLS或Verilog实现一个智能处理器MyTPU。

要求该智能处理器仿真结果正确，综合后可远程在FPGA开发板上正确运行。要求至少支持运行MLP模型。



实验概览 / 共 5 / 5 个实验

test.educg.net/admin/courseAdmin/expAdmin/guides/expDesktopWorkbench.jsp?hostname=CentOS7-Docker&token=44cc279835df6dfda94289b2ede818ce&vnc\_host=test.educg.net&vnc\_port=&imagelD=cg-vn...

远程桌面工作台 工作笔记 桌面全屏 隐藏桌面 更多

### 工作笔记 [转为实验指导书](#)

#### CPU设计与测试说明

处理器为32位处理器。

处理器应支持的指令集为：{addu, subu, ori, lw, sw, beq, lui, nop}。

nop机器码为0x00000000，即空指令，不进行任何有效行为（修改寄存器等）

addu,subu可以不支持溢出。

处理器为单周期设计。

需要采用模块化和层次化设计。顶层有效的驱动信号要求包括且仅包括：reset (clk 请使用内置时钟模块)。

#### 模块规格

- IFU (取指令单元)：内部包括 PC (程序计数器)、IM(指令存储器)及相关逻辑。  
PC 用寄存器实现，应具有复位功能。  
起始地址：0x00000000。  
IM用 ROM 实现，容量为 32bit \* 32。  
因 IM 实际地址宽度仅为 5 位，故需要使用恰当的方法将 PC 中储存的地址同 IM 联系起来。
- GRF (通用寄存器组，也称为寄存器文件、寄存器堆)  
用具有写使能的寄存器实现，寄存器总数为 32 个。  
0 号寄存器的值始终保持为 0。其他寄存器初始值均为 0，无需专门设置。
- ALU (算术逻辑单元)  
提供 32 位加、减、或运算及大小比较功能。  
可以不支持溢出 (不检测溢出)。
- DM (数据存储器)  
使用 RAM 实现，容量为 32bit \* 32。  
起始地址：0x00000000。  
RAM应使用双端口模式，即设置 RAM 的 Data Interface 属性为 Separate load and store ports,

Tcl Console

VIVADO HLx Editions XILINX

### Quick Start

Create Project >  
Open Project >  
Open Example >

### Task

Manage IP  
Open Hardware  
Xilinx Tcl Script

### Learn

Documents

Recent Projects

- test /headless/Desktop/test

Home - Mozilla Firefox

Home - Mozilla Firefox

busihuan.imwork.net:10003/tree?

### Jupyter

Files Running Clusters Nbextensions

Select items to perform actions on them.

Name	Last Modified
base	a year ago
common	a year ago
getting_started	a year ago
logictools	a year ago
oDisk	20 years ago
io_decode.ipynb	3 months ago
Welcome to Run.ipynb	a year ago

# GPU支持与调度

通用评测、云桌面、Jupyter皆支持GPU

- ◎ 基于GPU的通用评测
- ◎ GPU直通云桌面
- ◎ GPU直通Jupyter
- ◎ FPGA计算环境

基于通用评测实验环境的监督式机器学习编程实验样例。

学生提交代码后，平台按照代码在测试集上的准确度动态排名。

深度学习第一次作业 / 通用评测题 / 1. MNIST手写数字识别

### 【题目描述】

MNIST数据集及其相关介绍的地址为：<http://yann.lecun.com/ex>

该数据集的训练集包含55000个样本，测试集包含10000个样本。

在本题目中，对MNIST数据集进行了处理。每个样本为一张28像素为BMP，图片中每个像素的像素值为0或者255。

### 【评分标准】

在程序执行的当前目录下，包含一个 train 目录和一个 test 目录，系统会根据你提交程序输出的result.csv文件，计算你的模型在测试集上的准确率。

- train 目录为训练集，包含所有训练样本，每个样本为一张28x28像素的BMP文件，标签为数字0-9。
- test 目录为测试集，包含所有测试样本，每个样本为一张28x28像素的BMP文件。

深度学习第一次作业 / MNIST手写数字识别

请你基于机器学习或者深度学习对测试集中的图片进行分类。

### 【程序输出】

你的程序需要在当前目录下生成一个result.csv文件，该文件是由多行构成的，每行包含一个数字，其中，预测的分类值的列表。

### 注意：

- 如果你提交了多个

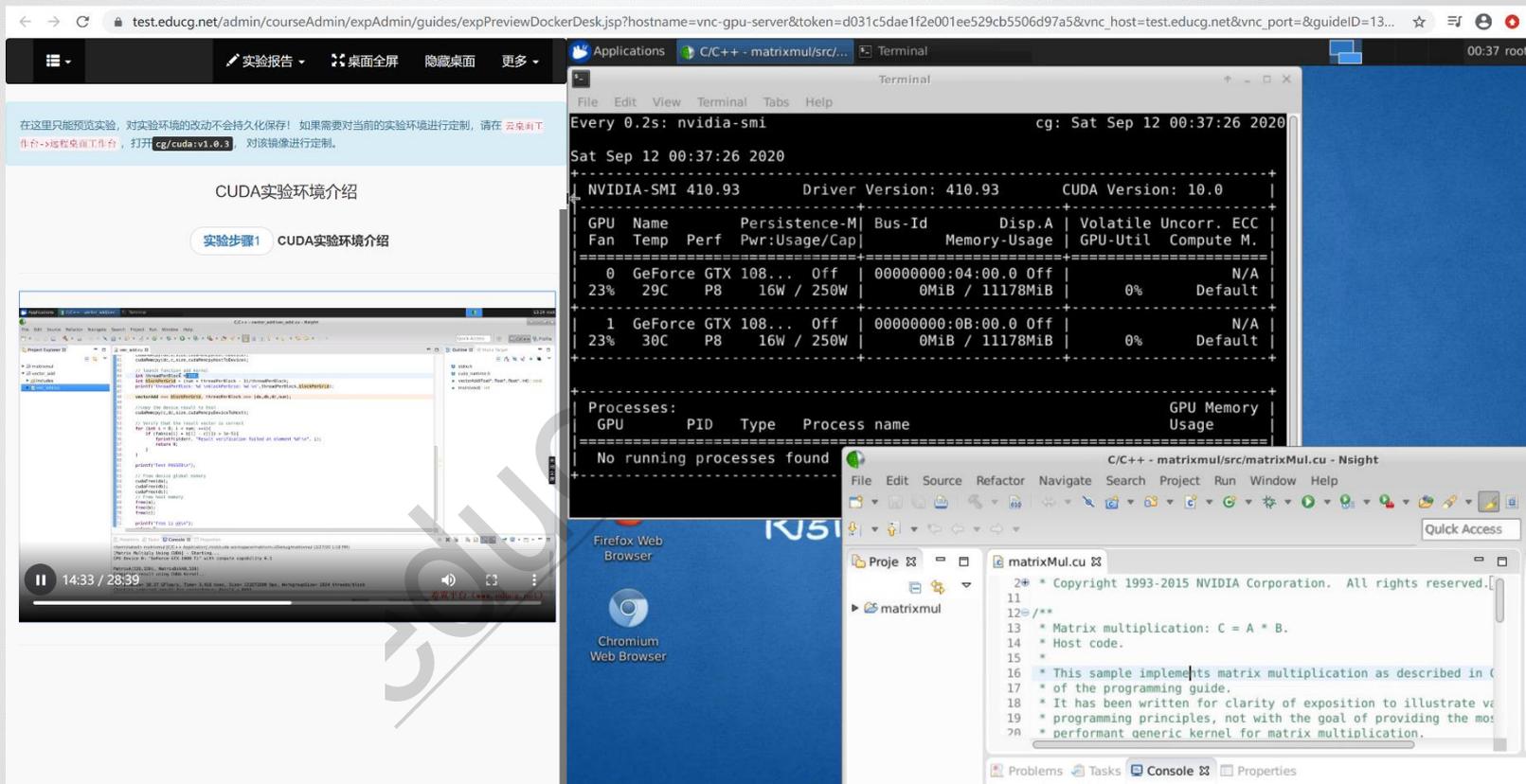
- 如果你提交了多个Python文件，那么其中必须有一个文件命名为main.py作为程序的入口。
- 可以采用GPU训练模型，但需要注意训练时间，程序的最长运行时间为180秒（3分钟），超过后会被强行终止。
- 操作系统版本为：Ubuntu 18.04.1 TLS，CUDA驱动版本为：CUDA 10.0，GPU型号为：Nvidia GeForce GTX 1080Ti，GPU数量：1
- Python版本为：3.6.7，Keras版本为：2.3.0，TensorFlow版本为：1.14.0，PyTorch版本为：1.0.0，MXNet版本为：1.3.1

## 排行榜

#	学号	姓名	rank
1	KAGGLE_001	KAGGLE_001	0.98
2	KAGGLE_004	KAGGLE_004	0.98
3	KAGGLE_005	KAGGLE_005	0.98
4	KAGGLE_002	KAGGLE_002	0.97
5	KAGGLE_003	KAGGLE_003	0.91

基于云桌面的GPU计算环境。

学生可在云桌面实验环境中直接调用GPU开展深度学习、强化学习、并行计算等课程实验。



The screenshot displays a cloud desktop interface. At the top, a browser window shows a page titled "CUDA实验环境介绍" (CUDA Experimental Environment Introduction) with a "实验步骤1" (Step 1) button. Below it, a terminal window runs the command `nvidia-smi`, displaying GPU information for two GeForce GTX 1080 Ti cards. The output table is as follows:

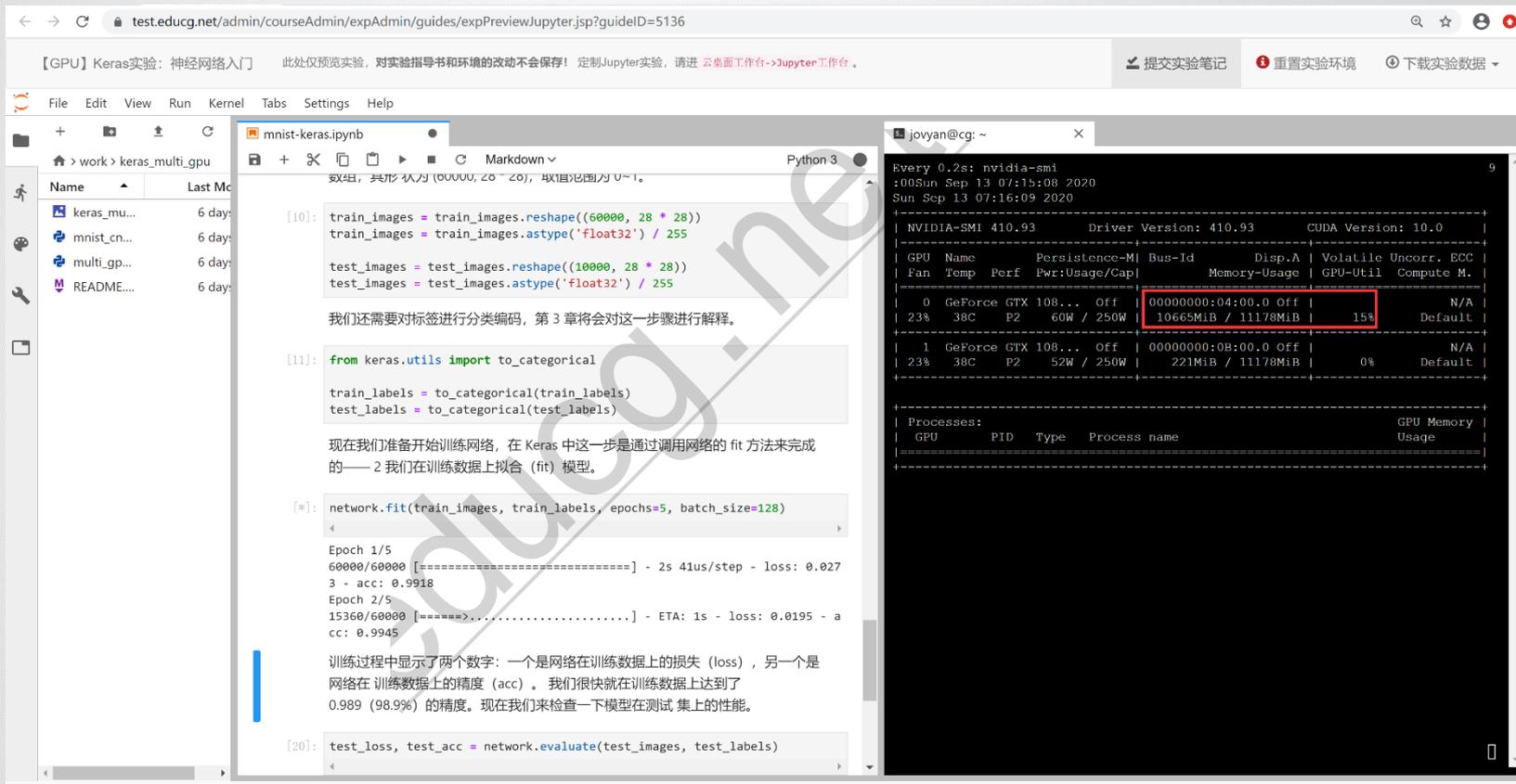
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
0	GeForce GTX 108...	Off	00000000:04:00.0	Off	N/A
23%	29C	P8	16W / 250W	0MiB / 11178MiB	0% Default
1	GeForce GTX 108...	Off	00000000:0B:00.0	Off	N/A
23%	30C	P8	16W / 250W	0MiB / 11178MiB	0% Default

Below the terminal, a code editor shows C++ code for matrix multiplication. The code includes comments in Chinese and C++ code for matrix multiplication. The code is as follows:

```
2 * Copyright 1993-2015 NVIDIA Corporation. All rights reserved.
11
12 /**
13  * Matrix multiplication: C = A * B.
14  * Host code.
15  */
16
17 * This sample implements matrix multiplication as described in
18 * of the programming guide.
19 * It has been written for clarity of exposition to illustrate va
20 * programming principles, not with the goal of providing the mo
21 * performant generic kernel for matrix multiplication.
```

基于 Jupyter 的 GPU 计算环境。

学生可在 Jupyter 实验环境中直接调用 GPU 开展深度学习、强化学习、并行计算等课程实验。



The screenshot displays a JupyterLab environment. The main notebook, titled 'mnist-keras.ipynb', contains Python code for training a Keras model on MNIST data. The code includes image reshaping, categorical label conversion, and the training process using `network.fit`. The training progress shows two epochs, with the second epoch reaching a loss of 0.0195 and an accuracy of 0.9945. A comment explains that the training process displays loss and accuracy, and that the model's performance on the test set is being checked.

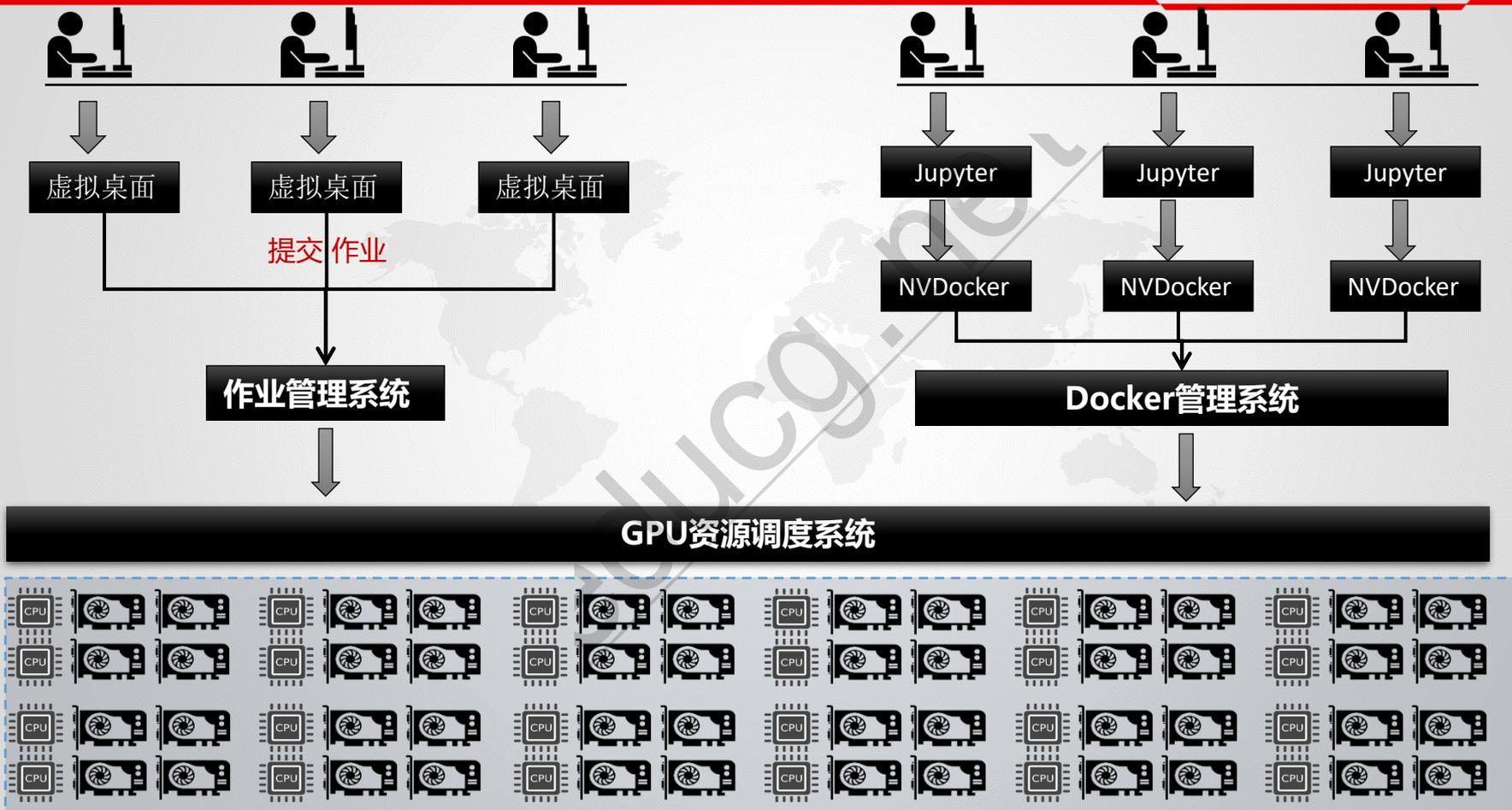
On the right side, a terminal window shows the output of the `nvidia-smi` command. The output includes system information and a table of GPU usage:

```
Every 0.2s: nvidia-smi
:00Sun Sep 13 07:15:08 2020
Sun Sep 13 07:16:09 2020
```

NVIDIA-SMI 410.93		Driver Version: 410.93		CUDA Version: 10.0	
GPU Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC	
Fan Temp Perf Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.		
0 GeForce GTX 108...	Off	00000000:04:00:0	Off	N/A	
23% 38C P2 60W / 250W	10665MiB / 11178MiB	15%		Default	
1 GeForce GTX 108...	Off	00000000:08:00:0	Off	N/A	
23% 38C P2 52W / 250W	221MiB / 11178MiB	0%		Default	

Below the table, the terminal shows the `Processes:` section with columns for GPU, PID, Type, Process name, and GPU Memory Usage.

# 人工智能专业：GPU支持与调度





03

---

**大数据专业**

## · 大数据专业一站式解决方案

基于CG一体化平台，沉淀教育数据，开展教育大数据科研

### 大数据基础能力培养



### 大数据专业能力培养



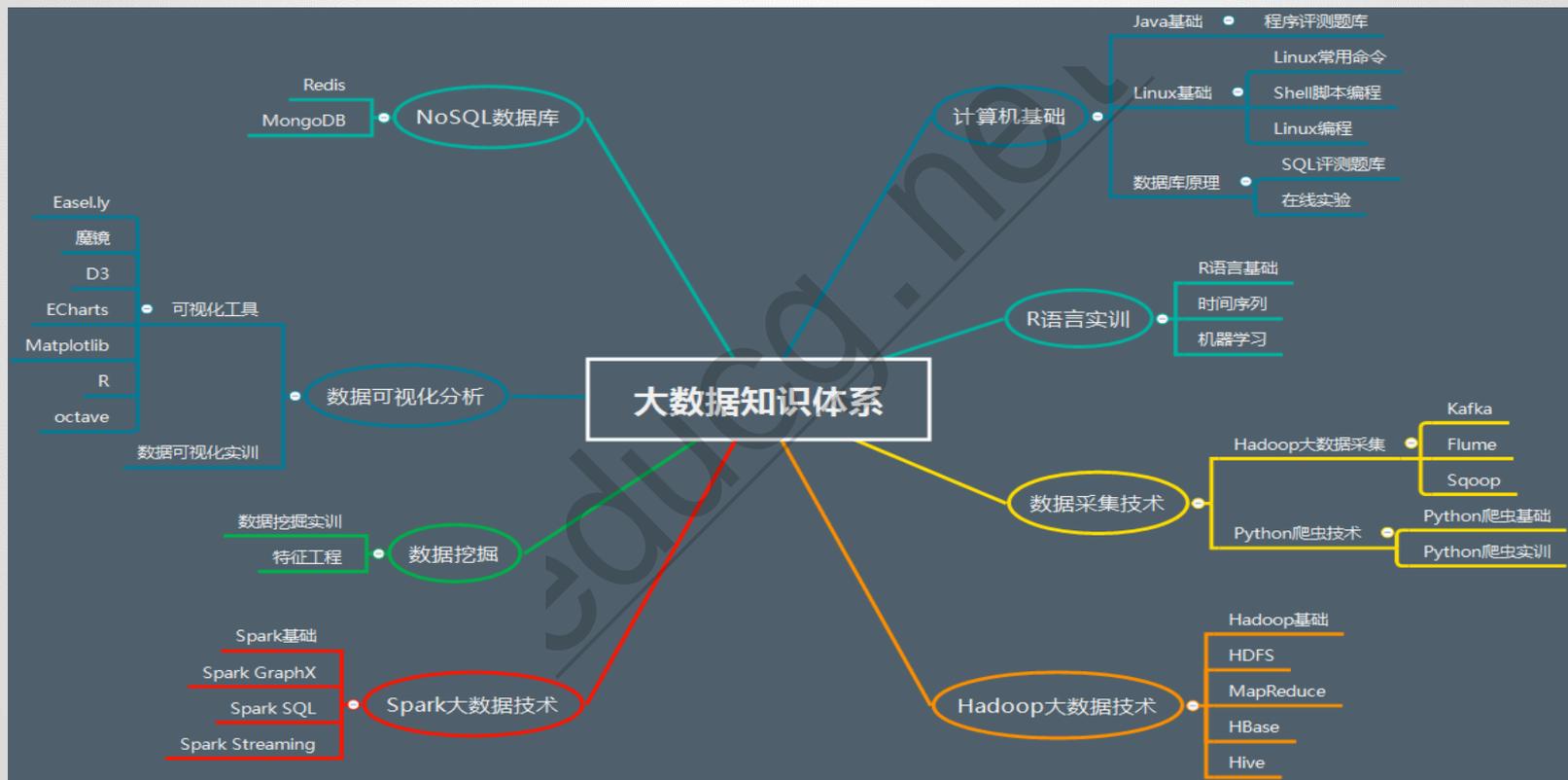
### 课程管理平台



### 系统管理与自动化运维

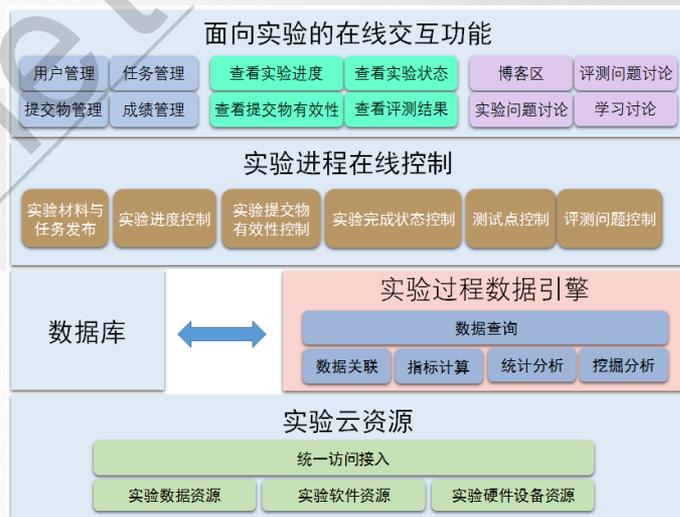
教育大数据

## • 大数据类课程实验体系

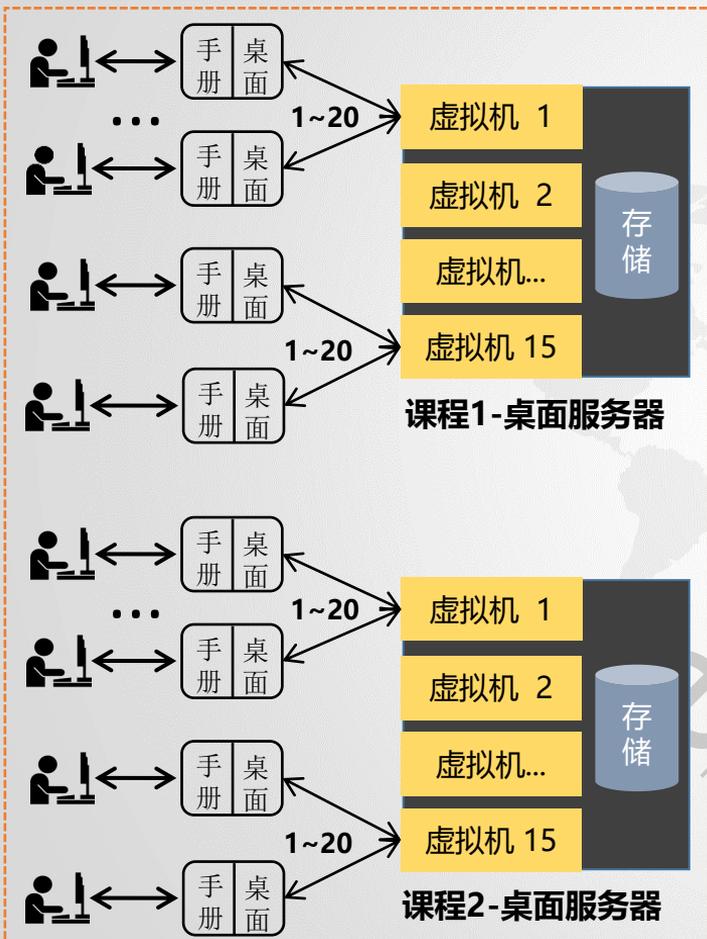


## • 大数据解决方案特点

	CG大数据	其它
软件平台与硬件松耦合	✓ 独立建设、独立维护升级换代	✗ 一体机模式，与服务器和虚拟化软件紧耦合
对专业支撑的全面性	✓ 支撑所有大数据专业课程的教学与实验	✗ 昂贵的软硬件，只能做有限的大数据实验，无法支持相关课程的实验， <b>例如数据库、编程语言、操作系统等。</b>
资源的可扩展性	✓ 轻松自建教学与实验资源	✗ 教学与实验资源固化
使用体验	✓ B/S架构图形桌面，客户端分辨率自适应	✗ C/S架构或者命令行界面

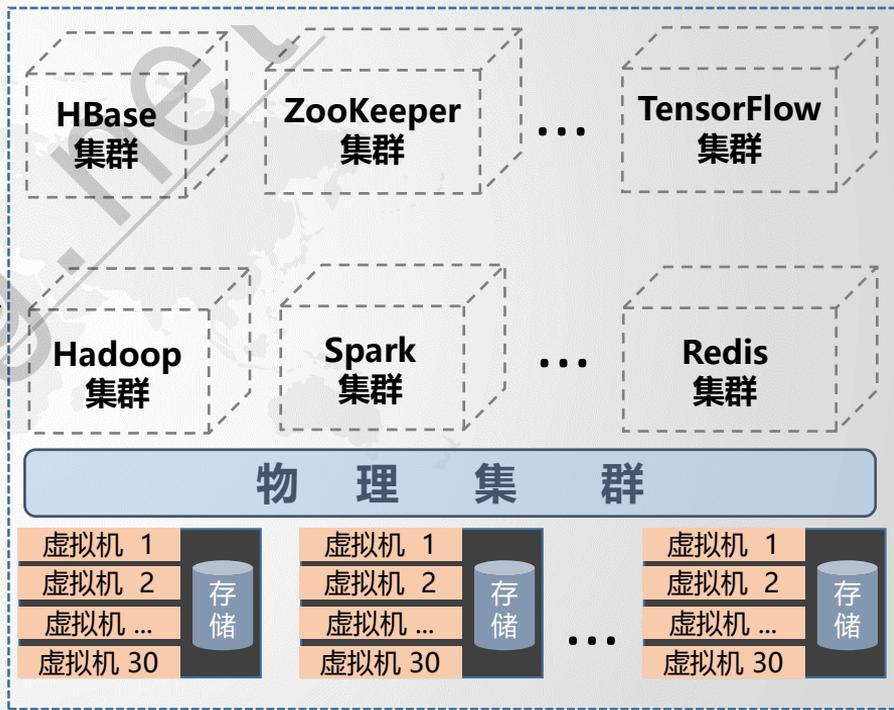


基于云计算技术，实验所需数据、软件共建和硬件实验设备均抽象为实验资源，提高了软硬件资源利用率，并利于系统维护和升级换代。



发送作业  
体验真实生产环境

- 大幅减少服务器数量。
- 真实生产环境。
- 发挥大数据工具真正的魅力。
- 支撑学生开展大型实验。

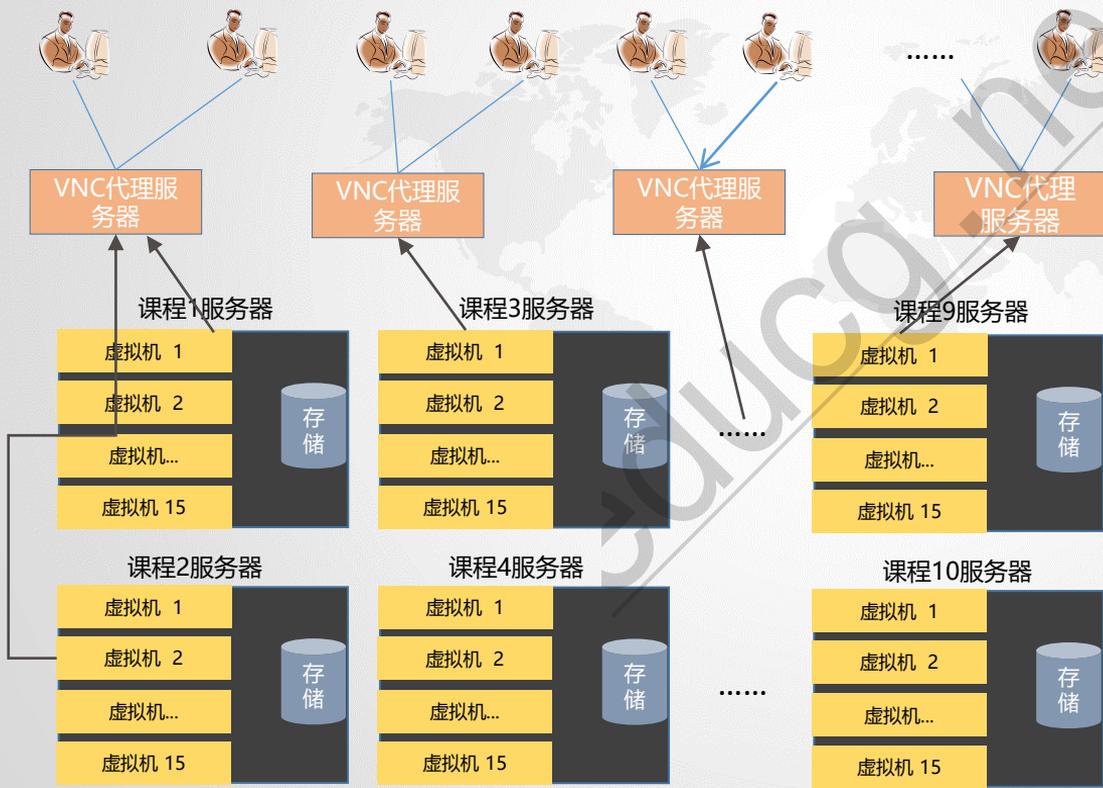


真正体会到大数据工具在编程模型、弹性调度、水平扩展、运行时容错、高可靠设计等方面的魅力。

## 若有3000人同时在线实验，如何部署？

保证桌面的流畅性，每用户1Mb的网络带宽。总带宽需求为3Gb，假设服务器为千兆网卡。

10台课程服务器，提供虚拟机环境。4台VNC代理服务器，总的理论带宽为4Gb。



## 一门课只需要一台服务器

### 说明

1. 一门课一般只有一个镜像，每位学生使用一个实例完成所有的实验。
2. 一门课对应一个学期，最大并发人数为学院每届招生人数。
3. CG实验环境支持一台虚拟机多个用户，相比独占虚拟机，对资源的需求可以呈几十倍的减少。

### 课程服务器推荐配置

1. CPU: 2 \* xeon E7, 15核
2. 内存: 256GB。
3. 硬盘: 4 \* 4TB
4. 网卡: 1Gb

### VNC服务器推荐配置

1. CPU: 1 \* xeon E5, 8核
2. 内存: 32GB
3. 硬盘: 300GB
4. 网卡: 1Gb

## 若有3000人同时在线实验，如何部署？

保证桌面的流畅性，每用户1Mb的网络带宽。总带宽需求为3Gb，假设服务器为千兆网卡。

10台课程服务器，提供虚拟机环境。4台VNC代理服务器，总的理论带宽为4Gb。



The screenshot shows the 'VNC代理' (VNC Proxy) configuration page in the CG Online Judge admin panel. A red arrow points to the 'VNC代理' menu item in the left sidebar. The main content area contains a notice: '一般情况下不用配置VNC代理！CG Web服务器已经自动部署了VNC服务。' (Generally, you don't need to configure VNC proxy! CG Web servers have automatically deployed VNC services.) Below this, it lists two scenarios where VNC proxy is needed: 1. High concurrent access (>1000 users) requiring dedicated VNC proxy servers to share bandwidth. 2. Remote access from a public IP network, where a proxy server with a public IP is needed. A table below shows the current VNC proxy server configuration:

主机ID	IP地址或域名	root密码	操作系统	备注	状态	部署	删除
CGVNCProxyServer	演示版隐藏IP	演示版隐藏密码	Ubuntu14		运行中, 查看日志	打开部署窗口	

At the bottom, there is a '+ 添加VNC代理服务器' (Add VNC proxy server) button.

## 一门课只需要一台服务器

### 说明

1. 一门课一般只有一个镜像，每位学生使用一个实例完成所有的实验。
2. 一门课对应一个学期，最大并发人数为学院每届招生人数。
3. CG实验环境支持一台虚拟机多个用户，相比独占虚拟机，对资源的需求可以呈几十倍的减少。

### 课程服务器推荐配置

1. CPU: 2 \* xeon E7, 15核
2. 内存: 256GB。
3. 硬盘: 4 \* 4TB
4. 网卡: 1Gb

### VNC服务器推荐配置

1. CPU: 1 \* xeon E5, 8核
2. 内存: 32GB
3. 硬盘: 300GB
4. 网卡: 1Gb



04

---

机器人专业

## 科研级仿真环境

### 科研领域广泛认可的仿真环境

- ◎ ROS
- ◎ MoveIt!
- ◎ OpenAI Gym
- ◎ 无人机
- ◎ 无人驾驶
- ◎ Gazebo
- ◎ RViz
- ◎ 机械臂
- ◎ 双臂机器人
- ◎ GPU

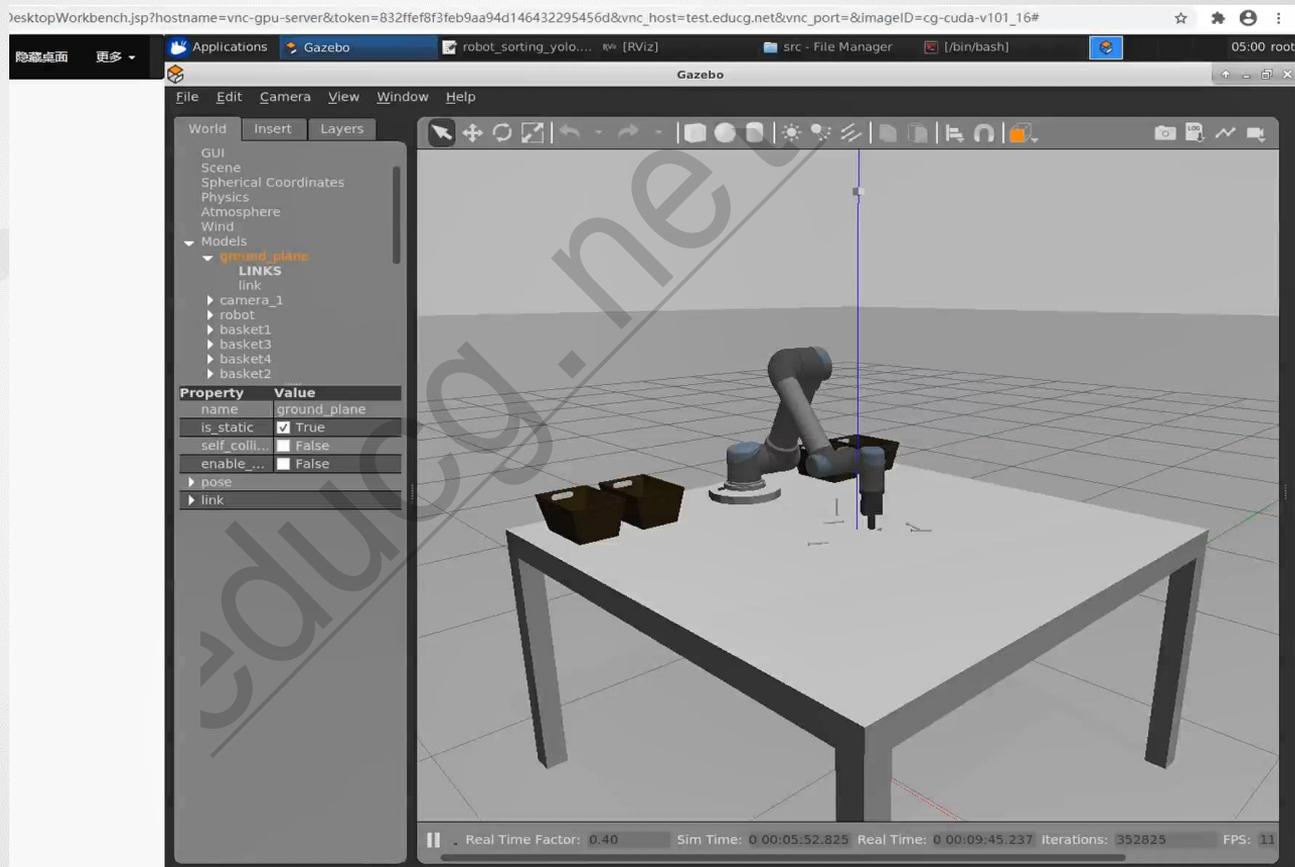
## • 提供科研级仿真实验环境

Gazebo机器人仿真环境	RViz三维可视化平台	MoveIt!机器人运动控制平台
Webots机器人仿真平台	V-REP机器人仿真平台	ADAMS移动机器人仿真平台
OpenAI Gym强化学习环境	Mujoco机器人仿真环境	PyBullets机器人仿真环境
Matlab机器人仿真环境	OpenAI、ROS、Gazebo联合机器人仿真环境	Matlab、ROS、Gazebo联合机器人仿真环境
Rotors Simulator、大疆DJI-Mavic等固定翼和多旋翼无人机智能飞行控制仿真环境	LGSVL、Autoware、Appolo自动驾驶仿真环境	Unity无人驾驶小车仿真环境
TurtleBot2、TurtleBot3等轮式机器人模型	UR3、UR5、PR2等机械臂模型	Husky与UR5构成的移动机械臂协作机器人模型

机器人仿真环境可快速、低成本、高安全性地验证机器人在结构设计、运动控制、轨迹规划、智能控制、复杂逻辑、高层AI算法等方面的工作在原理层面的有效性。

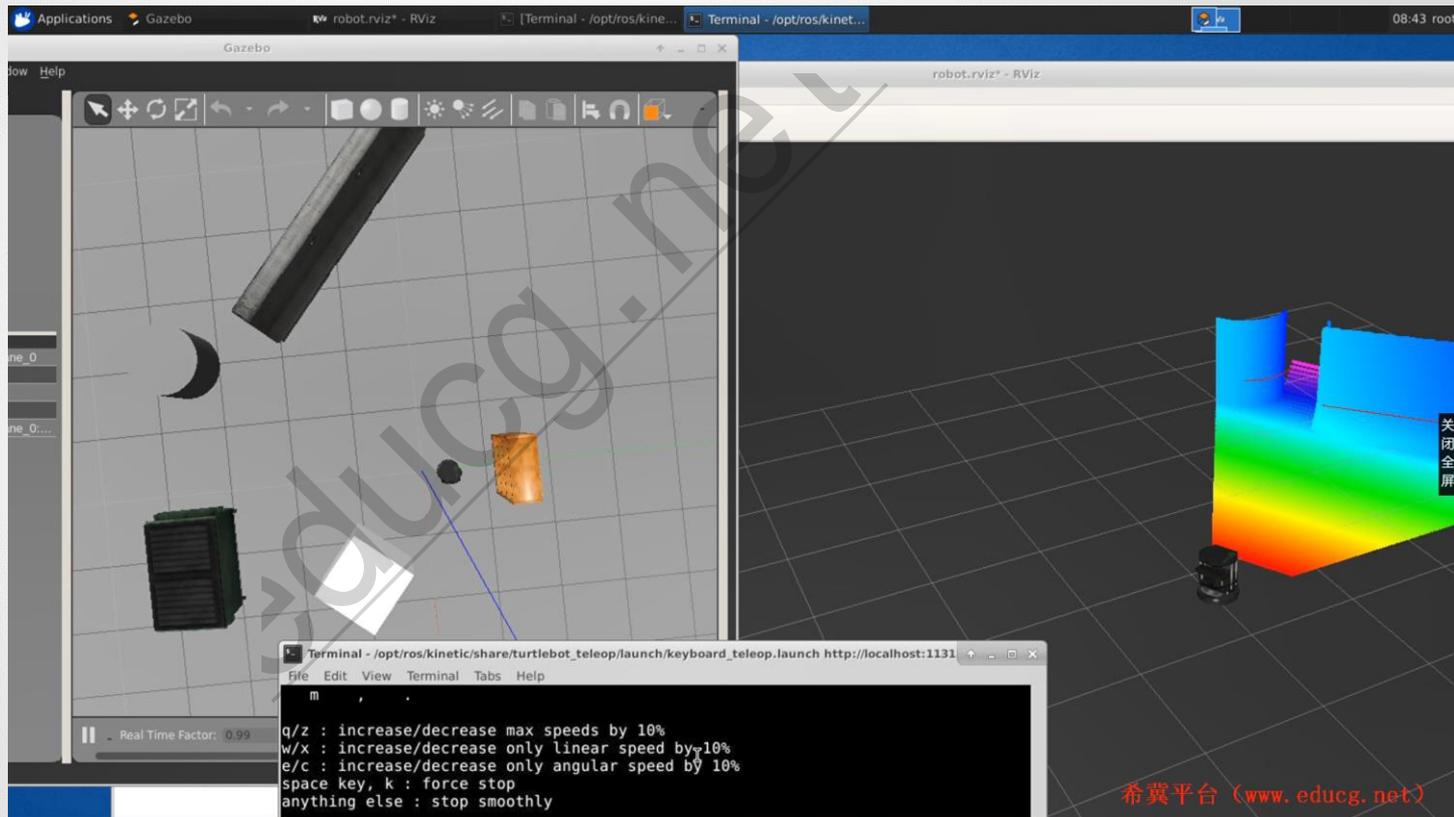
## • 随时随地在线实验

在希冀云桌面实验环境内，安装了实验所需的各类仿真软件，集成了各实验所需的实验支撑材料。



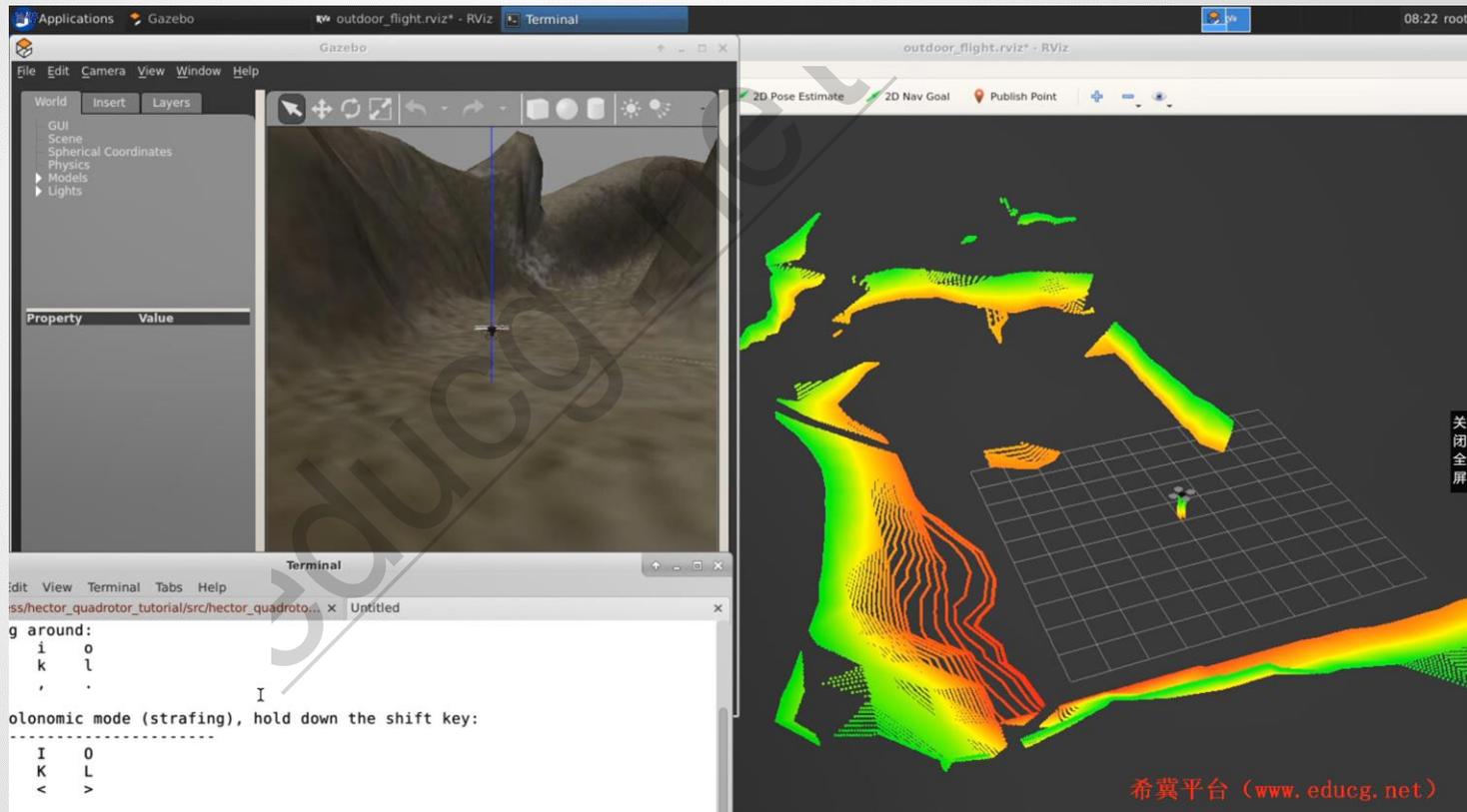
## • 提供科研级仿真实验环境

在云桌面实验环境内,提供智能控制与机器人科研领域相关期刊和会议所广泛采用与认可的仿真环境。



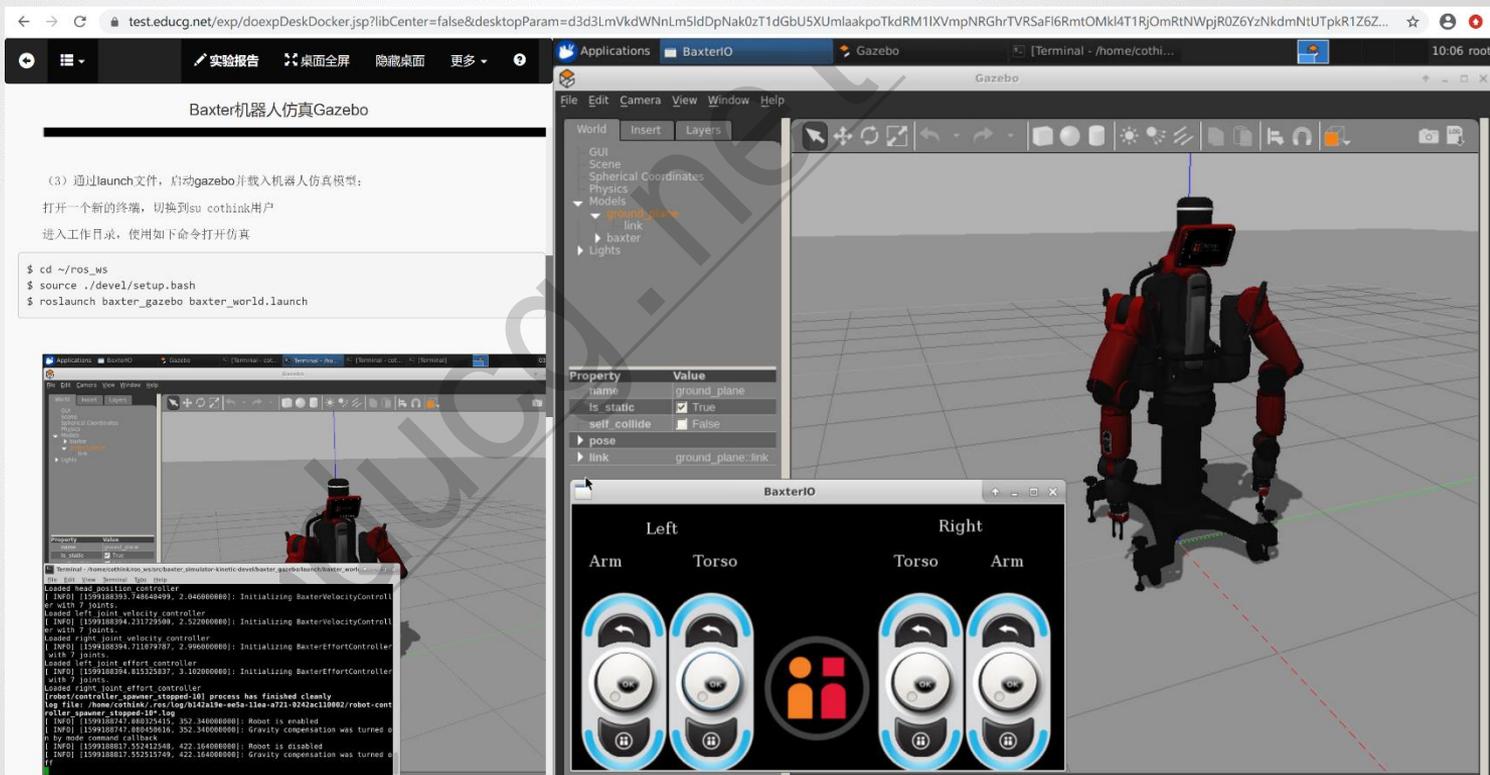
## 支持多种类型机器人

在科研级仿真实验环境内，提供了轮式机器人、机械臂、双足机器人、固定翼无人机、多旋翼无人机、自动驾驶汽车、无人驾驶小车等机器人模型。



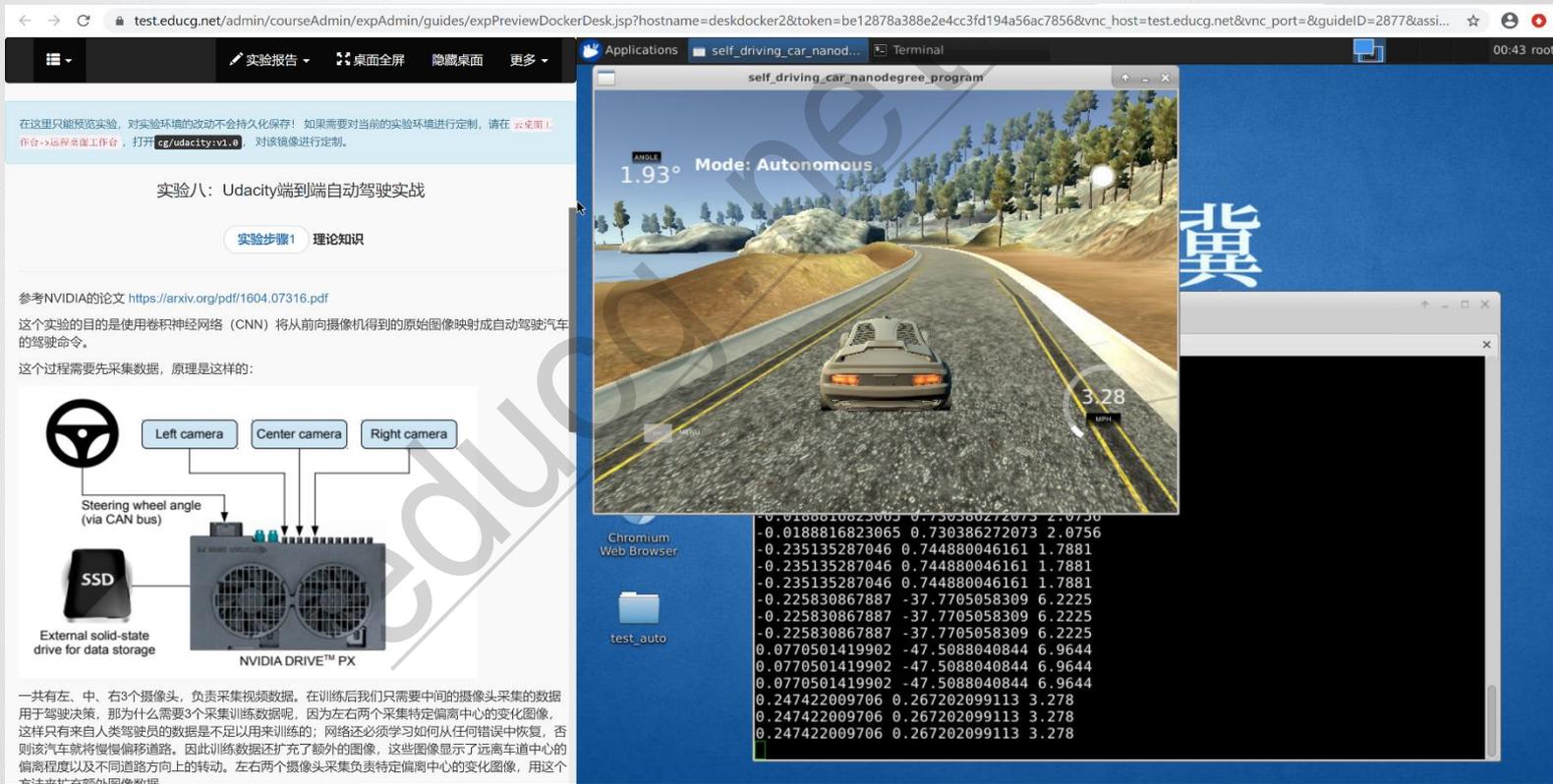
## 支持多种类型机器人

在科研级仿真实验环境内，提供了轮式机器人、机械臂、双足机器人、固定翼无人机、多旋翼无人机、自动驾驶汽车、无人驾驶小车等机器人模型。



## 支持多种类型机器人

在科研级仿真实验环境内，提供了轮式机器人、机械臂、双足机器人、固定翼无人机、多旋翼无人机、自动驾驶汽车、无人驾驶小车等机器人模型。



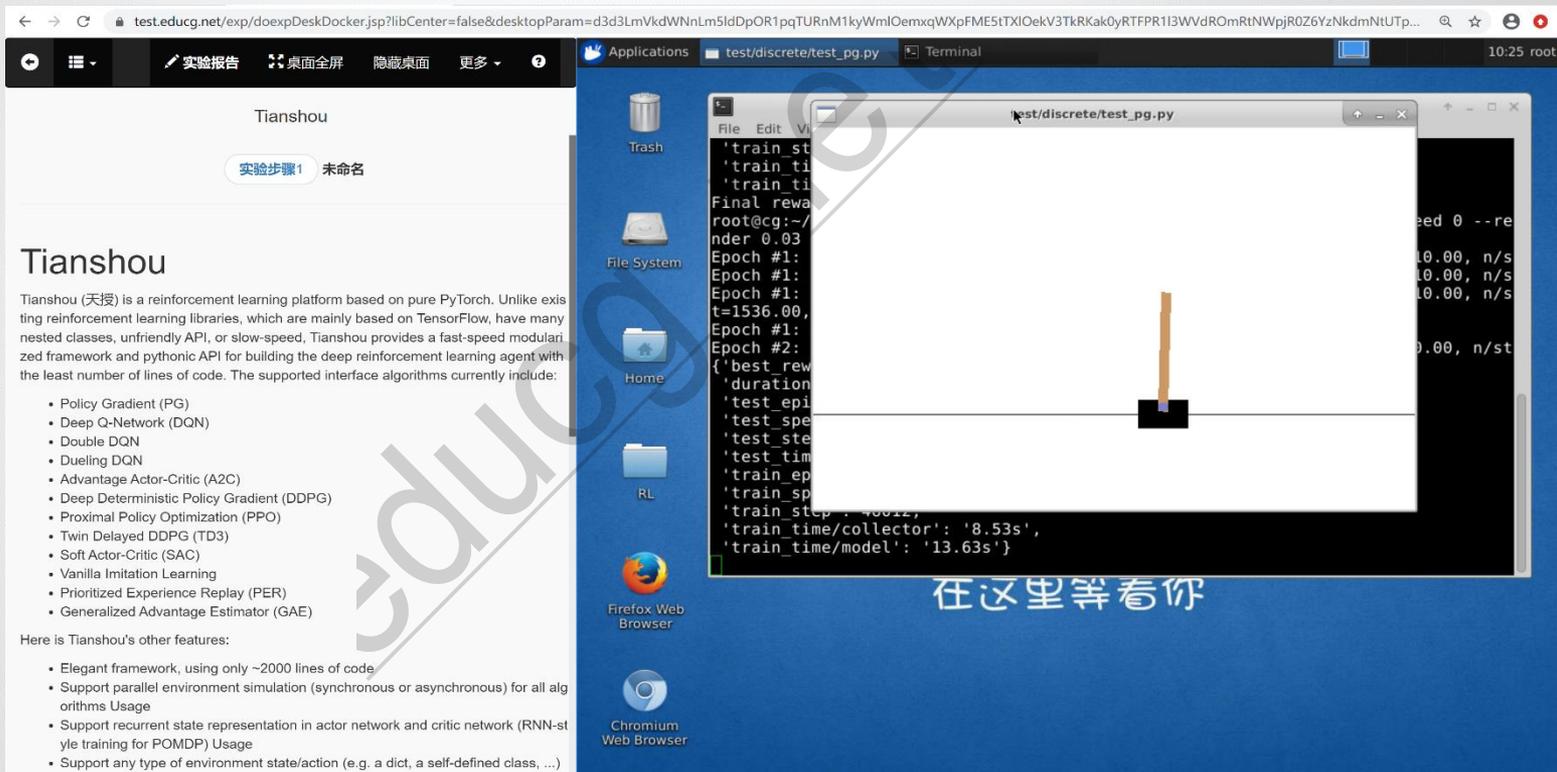
The image shows a web browser window displaying a Udacity course page for "self\_driving\_car\_nanodegree\_program". The page includes a navigation menu, a title "实验八：Udacity端到端自动驾驶实战", and a "理论知识" (Theory Knowledge) section. Below the text, there is a diagram of the sensor setup for the self-driving car simulation, showing three cameras (Left, Center, Right) and a steering wheel connected to an NVIDIA DRIVE PX system. The diagram also shows an external solid-state drive for data storage.

The terminal window shows the simulation running in "Mode: Autonomous" with a steering wheel angle of 1.93 degrees and a speed of 3.28 MPH. The terminal output displays a series of numerical data points, likely representing sensor readings or model outputs.

一共有左、中、右3个摄像头，负责采集视频数据。在训练后我们只需要中间的摄像头采集的数据用于驾驶决策，那为什么需要3个采集训练数据呢，因为左右两个采集特定偏离中心的变化图像，这样只有来自人类驾驶员的数据是不足以来训练的；网络还必须学习如何从任何错误中恢复，否则该汽车就将慢慢偏移道路。因此训练数据还扩充了额外的图像，这些图像显示了远离车道中心的偏离程度以及不同道路方向上的转动。左右两个摄像头采集负责特定偏离中心的变化图像，用这个方法扩充采集图像数据。

## • 支持多种类型机器人

在科研级仿真实验环境内，提供了轮式机器人、机械臂、双足机器人、固定翼无人机、多旋翼无人机、自动驾驶汽车、无人驾驶小车等机器人模型。



The screenshot displays a virtual machine interface. On the left, a web browser shows the Tianshou website, which describes it as a reinforcement learning platform based on pure PyTorch. The website lists various supported algorithms and features. On the right, a terminal window shows the execution of a Python script named 'test\_pg.py'. The terminal output displays training progress, including epoch numbers, time taken, and performance metrics. A large watermark 'educug' is overlaid on the terminal area.

**Tianshou**

Tianshou (天授) is a reinforcement learning platform based on pure PyTorch. Unlike existing reinforcement learning libraries, which are mainly based on TensorFlow, have many nested classes, unfriendly API, or slow-speed, Tianshou provides a fast-speed modularized framework and pythonic API for building the deep reinforcement learning agent with the least number of lines of code. The supported interface algorithms currently include:

- Policy Gradient (PG)
- Deep Q-Network (DQN)
- Double DQN
- Dueling DQN
- Advantage Actor-Critic (A2C)
- Deep Deterministic Policy Gradient (DDPG)
- Proximal Policy Optimization (PPO)
- Twin Delayed DDPG (TD3)
- Soft Actor-Critic (SAC)
- Vanilla Imitation Learning
- Prioritized Experience Replay (PER)
- Generalized Advantage Estimator (GAE)

Here is Tianshou's other features:

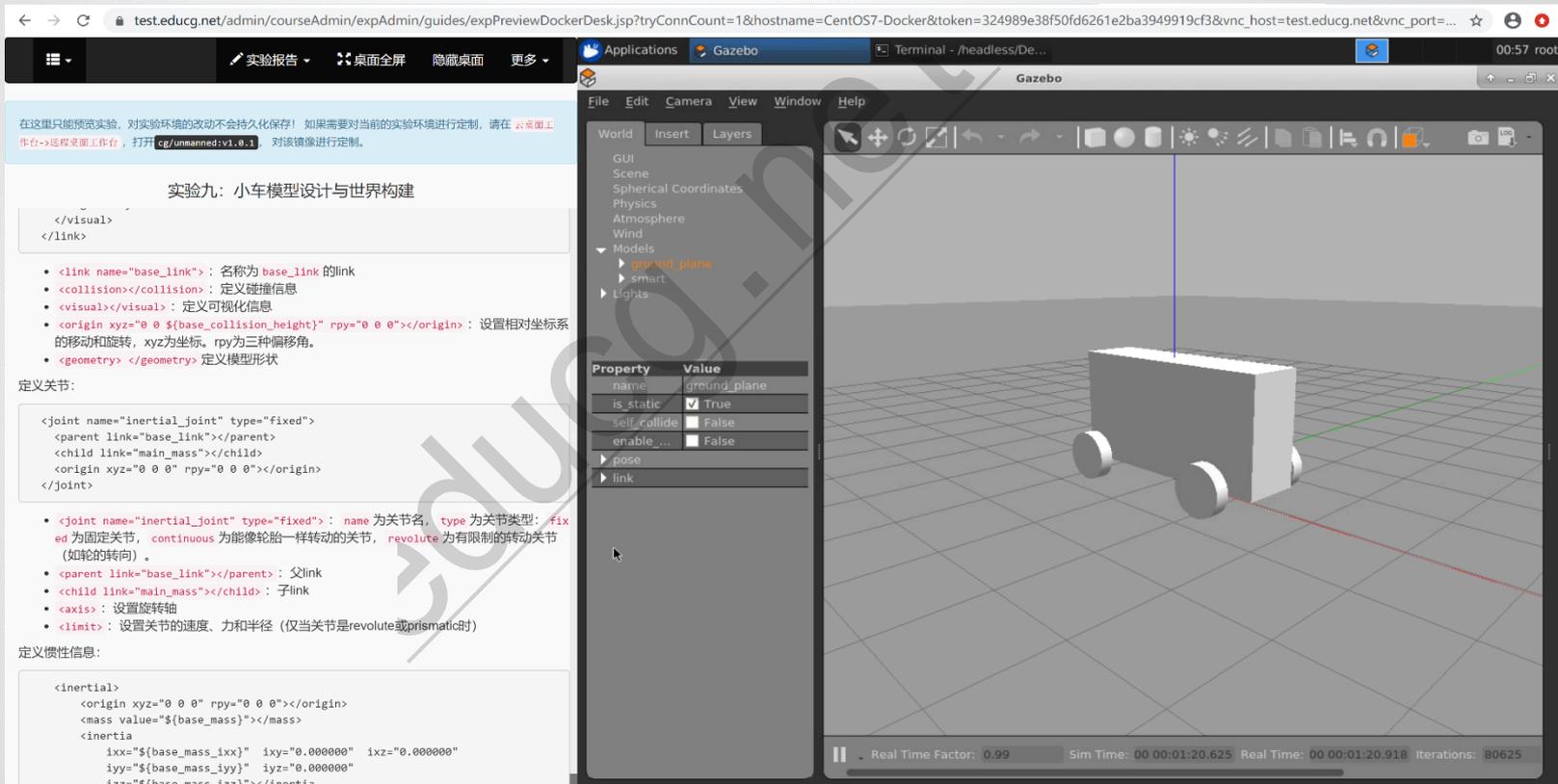
- Elegant framework, using only ~2000 lines of code
- Support parallel environment simulation (synchronous or asynchronous) for all algorithms Usage
- Support recurrent state representation in actor network and critic network (RNN-style training for POMDP) Usage
- Support any type of environment state/action (e.g. a dict, a self-defined class, ...)

```
File Edit View
test/discrete/test_pg.py
train_st
'train ti
'train ti
'train ti
Final rew
root@cg:~/
nder 0.03
Epoch #1:
Epoch #1:
t=1536.00,
Epoch #1:
Epoch #2:
{'best rew
'duration
'test epi
'test spe
'test ste
'test tim
'train ep
'train sp
'train step: 48012,
'train_time/collector': '8.53s',
'train_time/model': '13.63s'}
```

在这里等着你

## • 自定义仿真环境

提供了一系列实验，引导学生自己动手构建机器人模型和仿真世界。



test.educg.net/admin/courseAdmin/expAdmin/guides/expPreviewDockerDesk.jsp?tryConnCount=1&hostname=CentOS7-Docker&token=324989e38f50fd6261e2ba3949919cf3&vnc\_host=test.educg.net&vnc\_port=...

Applications Gazebo Terminal - /headless/De... 00:57 root

Gazebo

File Edit Camera View Window Help

World Insert Layers

GUI  
Scene  
Spherical Coordinates  
Physics  
Atmosphere  
Wind  
Models  
▶ ground\_plane  
▶ smart  
Lights

Property	Value
name	ground_plane
is_static	<input checked="" type="checkbox"/> True
self_collide	<input type="checkbox"/> False
enable...	<input type="checkbox"/> False
▶ pose	
▶ link	

实验九：小车模型设计与世界构建

```
</visual>
</link>
```

- `<link name="base_link">`：名称为 `base_link` 的link
- `<collision></collision>`：定义碰撞信息
- `<visual></visual>`：定义可视化信息
- `<origin xyz="0 0 0" rpy="0 0 0"></origin>`：设置相对坐标系的移动和旋转，xyz为坐标，rpy为三种偏移角。
- `<geometry>` `</geometry>` 定义模型形状

定义关节：

```
<joint name="inertial_joint" type="fixed">
<parent link="base_link"></parent>
<child link="main_mass"></child>
<origin xyz="0 0 0" rpy="0 0 0"></origin>
</joint>
```

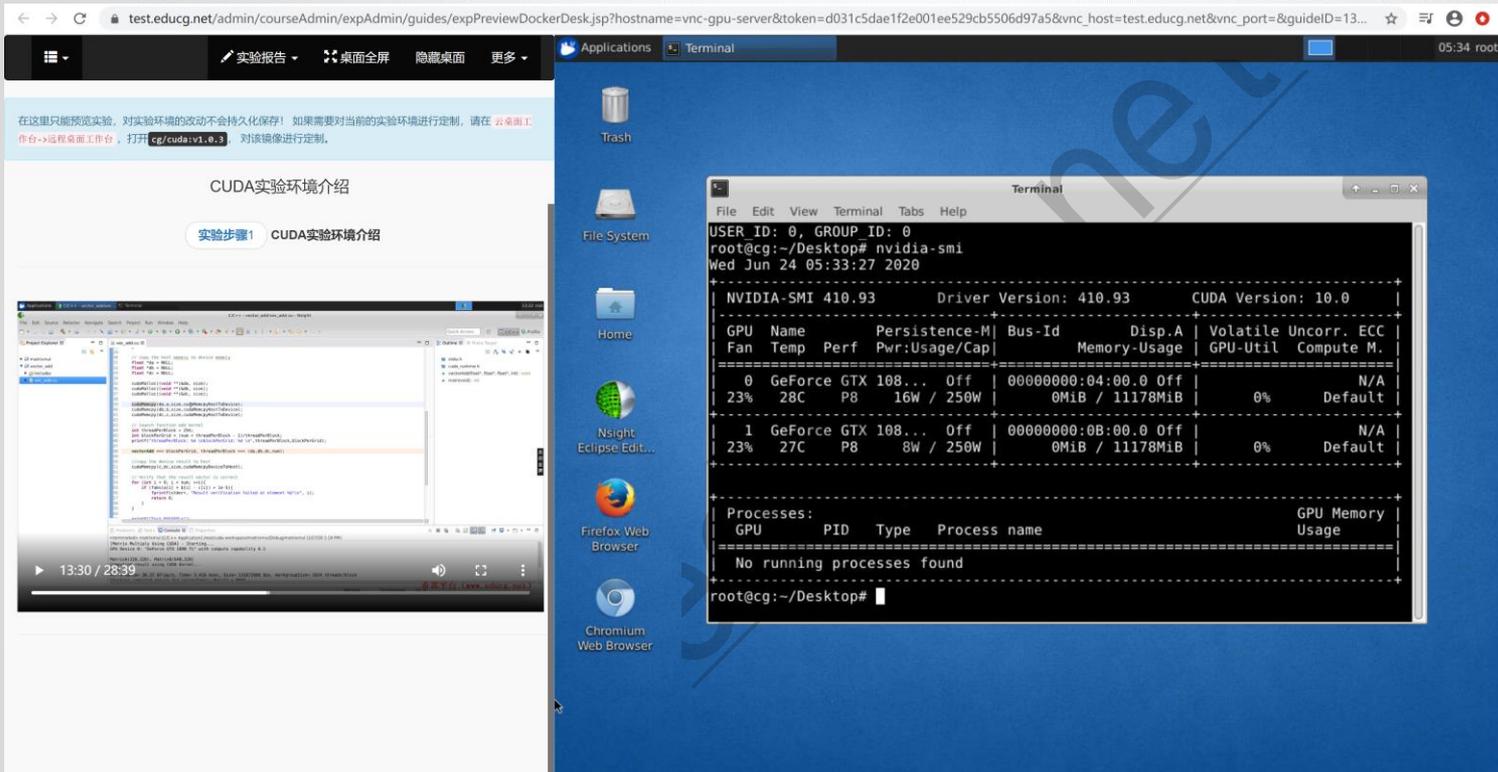
- `<joint name="inertial_joint" type="fixed">`：name 为关节名，type 为关节类型：`fixed` 为固定关节，`continuous` 为能像轮胎一样转动的关节，`revolute` 为有限的转动关节（如轮的转向）。
- `<parent link="base_link"></parent>`：父link
- `<child link="main_mass"></child>`：子link
- `<axis>`：设置旋转轴
- `<limit>`：设置关节的速度、力和半径（仅当关节是revolute或prismatic时）

定义惯性信息：

```
<inertial>
<origin xyz="0 0 0" rpy="0 0 0"></origin>
<mass value="${base_mass}"></mass>
<inertia
  Ixx="${base_mass_Ixx}" Ixy="0.000000" Ixz="0.000000"
  Iyy="${base_mass_Iyy}" Iyz="0.000000"
  Izz="${base_mass_Izz}"></inertia>
```

Real Time Factor: 0.99 Sim Time: 00 00:01:20.625 Real Time: 00 00:01:20.918 Iterations: 80625

## • 仿真环境支持GPU加速



The screenshot displays a cloud desktop interface. On the left, a web browser shows a page titled "CUDA实验环境介绍" (CUDA Experiment Environment Introduction) with a "实验步骤1" (Experiment Step 1) button. The desktop background is blue and features a sidebar with icons for Trash, File System, Home, Nsight, Eclipse Edit..., Firefox Web Browser, and Chromium Web Browser. A terminal window is open in the foreground, displaying the output of the command `cat /dev/nvidia0`. The terminal output includes the following information:

```
USER ID: 0, GROUP ID: 0
root@cg:~/Desktop# nvidia-smi
Wed Jun 24 05:33:27 2020

=====+-----+
| NVIDIA-SMI 410.93      Driver Version: 410.93      CUDA Version: 10.0
|-----+-----+
| GPU Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|-----+-----+
| 0   GeForce GTX 108...    Off      | 00000000:04:00:0 | Off      |          0%      Default |
| 23%   28C    P8     16W / 250W  | 0MiB / 11178MiB |           |                 |
|-----+-----+
| 1   GeForce GTX 108...    Off      | 00000000:0B:00:0 | Off      |          0%      Default |
| 23%   27C    P8     8W / 250W  | 0MiB / 11178MiB |           |                 |
|-----+-----+

Processes:
GPU      PID  Type  Process name                      GPU Memory
Usage
=====+-----+
No running processes found
=====+-----+

root@cg:~/Desktop#
```

◆ 支持多块GPU卡直通云桌面实验环境。

◆ 提升深度学习和强化学习相关算法的训练效率。

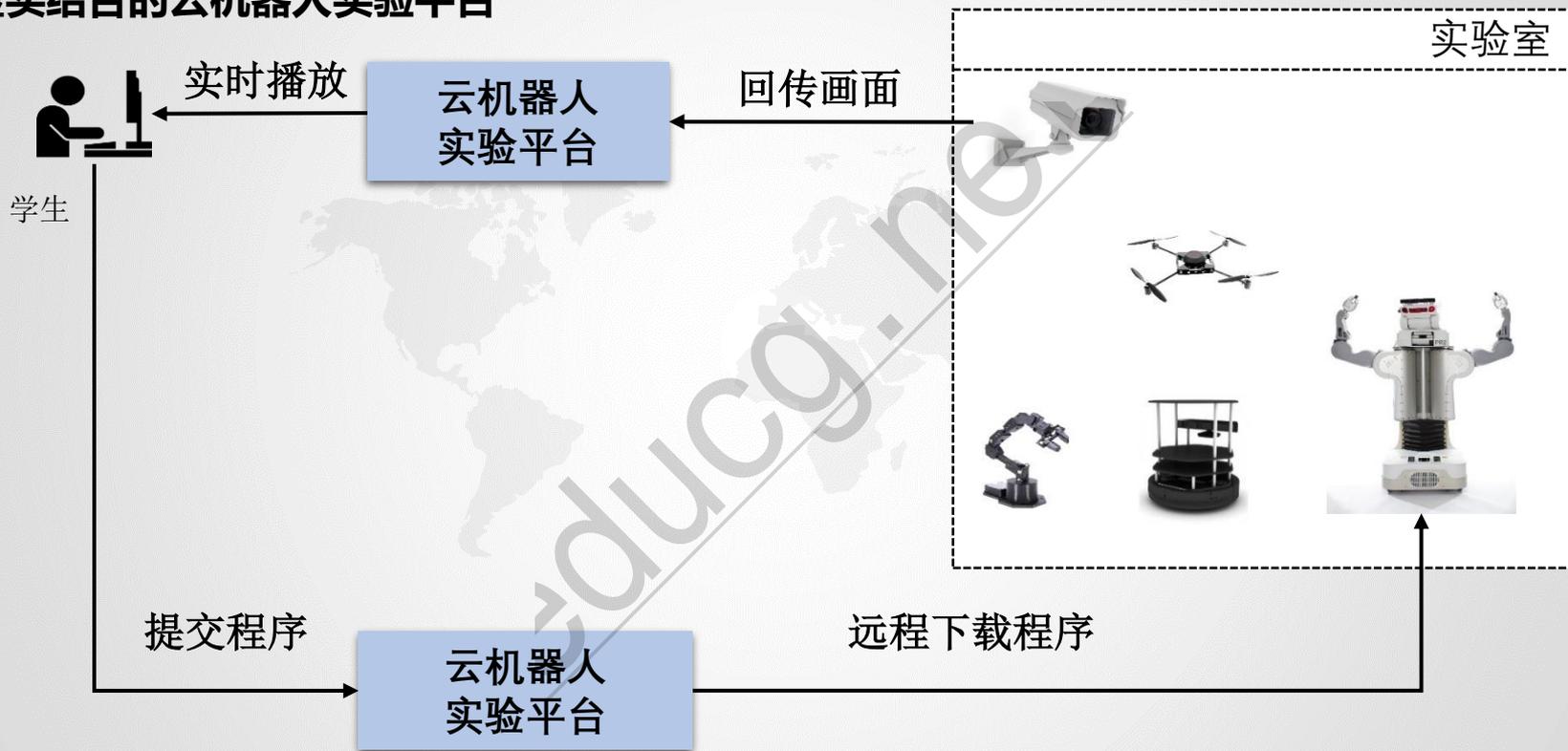
◆ 加快复杂仿真环境的图形渲染速度。

# 云机器人实验平台

## 连接远程机器人开展实验

- ◎ 服务机器人
- ◎ 移动机器人
- ◎ 双臂机器人
- ◎ 可对接任意机器人

## • 虚实结合的云机器人实验平台



◆ 在实体机器人上远程运行程序

◆ 实体机器人运行过程实时播放

◆ 不绑定不依赖特定实体机器人

- 实验实验室场景布局



## • 移动机器人SLAM导航实验



远程桌面工作台

test.educg.net/admin/courseAdmin/expAdmin/guides/expDesktopWorkbench.jsp?hostname=oslbanddesktopdocker&token=71a169ecf295043e5794cid896d8fd63&vnc\_host=test.educg.net&vnc\_port=80imgel...

远程桌面工作台 工作笔记 桌面全屏 隐藏桌面 更多 -

工作笔记 转换为实验指导书

### 云机器人实验

- 双击图标“Cloud Robot”，连接云端机器人上位机
- 启动 roscore
- 启动 turtlebot3
- 启动 RViz 地图
- 启动键盘控制程序
- 启动摄像头
- 重新布局窗口
- 开始进行建图实验

Trash Project

File System Cloud Robot

Home

Firefox Web Browser

Chromium Web Browser

Vivado

Right here waiting for you  
在这里等着你

在这里输入你要搜索的内容

17:20  
2020/7/5

## • 双臂机器人抓取实验



The screenshot displays a remote desktop workstation. On the left, a web browser shows a page titled "机械臂" (Robotic Arm) with the following text:

**机械臂**

机械臂是指高精度、多输入多输出、高度非线性、强耦合的复杂系统。因其独特的操作灵活性，已在工业装配、安全防护等领域得到广泛应用。

机械臂是一个复杂系统，存在着参数摄动、外界干扰及未建模动态等不确定性，因而机械臂的建模模型也存在着不确定性，对于不同的任务，需要规划机械臂关节空间的运动轨迹，从而级联构成末端位姿。

**机械臂系统**

机器人系统是由视觉传感器、机械臂系统及主控计算机组成，其中机械臂系统又包括模块化机械臂和灵巧手两部分。整个系统的构建模型如图1所示。

Figure 1 is a schematic diagram of the robotic arm system. It shows a "天花板" (Ceiling) at the top, with a "视觉传感器" (Visual Sensor) and a "灵巧手" (Dexterous Hand) attached to a horizontal bar. A coordinate system  $(X_c, Y_c, Z_c)$  is centered at the sensor. Below this, a "模块化机械臂" (Modular Robotic Arm) is shown with its own coordinate system  $(X_m, Y_m, Z_m)$ . A "CAN通讯" (CAN Communication) line connects the arm to a "主控计算机" (Main Control Computer) at the bottom right.

On the right side of the workstation, a Linux desktop environment is visible with a blue background. It features the "希冀 Course Grading" logo and the text "Right here waiting for you" and "在这里等着你". The desktop includes icons for "Trash", "cg-ros", "File System", "Home", "Firefox Web Browser", and "Chromium Web Browser". The system tray at the bottom right shows the time as 19:00 on 2020/8/15.

## 课程与实验体系

### 一站式支撑机器人专业核心课程

- ◎ 服务机器人
- ◎ 移动机器人
- ◎ 双臂机器人
- ◎ 可对接任意机器人

## 课程体系与实验列表

ROS编程与实践	机器人系统仿真
机器人运动控制	机器人感知系统
机器人运动规划	机器人自主导航
机械臂智能控制	无人机智能飞行控制
自动驾驶	无人小车智能控制
...	...

课程	实验名称	实验难度	实验学时	实验手册	实验代码	实验数据
ROS编程与实践	ROS 入门	☆☆☆	2	✓	✓	✓
	ROS文件系统	☆☆☆	2	✓	✓	✓
	ROS节点	☆☆☆	2	✓	✓	✓
	ROS话题	☆☆☆	2	✓	✓	✓
	ROS服务	☆☆☆	2	✓	✓	✓
	ROS消息发布与订阅	☆☆☆	2	✓	✓	✓
	ROS服务与客户端	☆☆☆	2	✓	✓	✓
	roscpp程序设计	☆☆☆☆☆	4	✓	✓	✓
	rospy程序设计	☆☆☆☆☆	4	✓	✓	✓
	TF坐标转换	☆☆☆	4	✓	✓	✓
	数据录制与回放	☆☆☆	4	✓	✓	✓
	...	...	...	...	...	...
	机器人系统仿真	Gazebo仿真环境的使用	☆☆☆	4	✓	✓
RViz可视化平台的使用		☆☆☆	4	✓	✓	✓
MoveIt!运动控制平台的使用		☆☆☆☆☆	4	✓	✓	✓
OpenAI Gym 强化学习平台的使用		☆☆☆☆☆	4	✓	✓	✓
Mujoco 机器人仿真环境的使用		☆☆☆☆☆	4	✓	✓	✓
Rotors Simulator 无人机仿真环境的使用		☆☆☆	4	✓	✓	✓
Unity 仿真环境的使用		☆☆☆	2	✓	✓	✓

更多信息，访问：[www.educg.net/robot.html](http://www.educg.net/robot.html)

## 课程体系与实验列表

ROS编程与实践	机器人系统仿真
机器人运动控制	机器人感知系统
机器人运动规划	机器人自主导航
机械臂智能控制	无人机智能飞行控制
自动驾驶	无人小车智能控制
...	...

课程名称	评价	学时	必修	选修	其他
<b>机器人系统仿真</b>					
Rotors Simulator 无人机仿真环境的使用	☆☆☆	4	✓	✓	✓
Unity 仿真环境的使用	☆☆☆	2	✓	✓	✓
URDF 机器人建模	☆☆☆	4	✓	✓	✓
URDF 机器人模型优化	☆☆☆☆	4	✓	✓	✓
...	...	...	...	...	...
<b>机器人运动控制</b>					
位置方位与坐标系	☆☆	2	✓	✓	✓
齐次坐标系转换	☆☆☆	2	✓	✓	✓
运动学建模	☆☆☆	2	✓	✓	✓
正向运动学求解	☆☆☆	4	✓	✓	✓
逆向运动学求解	☆☆☆	4	✓	✓	✓
脉宽调制 PWM	☆☆☆☆	4	✓	✓	✓
PID 控制算法	☆☆☆☆	4	✓	✓	✓
...	...	...	...	...	...
<b>机器人感知系统</b>					
位置感知	☆☆☆	2	✓	✓	✓
红外感知	☆☆☆	2	✓	✓	✓
速度感知	☆☆☆	2	✓	✓	✓
姿态感知	☆☆☆	2	✓	✓	✓
语音感知	☆☆☆☆	4	✓	✓	✓
视觉感知	☆☆☆☆	4	✓	✓	✓

更多信息，访问：[www.educg.net/robot.html](http://www.educg.net/robot.html)

## 课程体系与实验列表

ROS编程与实践	机器人系统仿真
机器人运动控制	机器人感知系统
机器人运动规划	机器人自主导航
机械臂智能控制	无人机智能飞行控制
自动驾驶	无人小车智能控制
...	...

机器人感知系统	姿态感知	☆☆☆	2	✓	✓	✓
	语音感知	☆☆☆☆	4	✓	✓	✓
	视觉感知	☆☆☆☆	4	✓	✓	✓
	...	...	...	...	...	...
机器人运动规划	轨迹规划	☆☆☆	4	✓	✓	✓
	路径规划	☆☆☆	4	✓	✓	✓
	智能规划	☆☆☆☆	4	✓	✓	✓
	...	...	...	...	...	...
机器人自主导航	SLAM 地图系统	☆☆☆	2	✓	✓	✓
	基于激光雷达的 SLAM 地图构建	☆☆☆	2	✓	✓	✓
	基于激光雷达的自主导航	☆☆☆	2	✓	✓	✓
	基于三维视觉的 SLAM 地图构建	☆☆☆☆	4	✓	✓	✓
	基于三维视觉的自主导航	☆☆☆☆	4	✓	✓	✓
...	...	...	...	...	...	...
机械臂智能控制	机械臂仿真环境介绍	☆☆	2	✓	✓	✓
	Movelt!机械臂仿真	☆☆☆	2	✓	✓	✓
	机械臂运动规划	☆☆☆	4	✓	✓	✓
	机械臂语音感知与控制	☆☆☆☆	4	✓	✓	✓
	机械臂视觉感知与控制	☆☆☆☆	4	✓	✓	✓
	机械臂智能抓取	☆☆☆☆	4	✓	✓	✓

更多信息，访问：[www.educg.net/robot.html](http://www.educg.net/robot.html)

## 课程体系与实验列表

ROS编程与实践	机器人系统仿真
机器人运动控制	机器人感知系统
机器人运动规划	机器人自主导航
机械臂智能控制	无人机智能飞行控制
自动驾驶	无人小车智能控制
...	...

	机械臂视觉感知与控制	★★★★★	4	✓	✓	✓
	机械臂智能抓取	★★★★★	4	✓	✓	✓
	...	...	...	...	...	...
无人机智能飞行控制	无人机智能避障	★★★★★	4	✓	✓	✓
	无人机智能导航	★★★★★	4	✓	✓	✓
	无人机姿态调整	★★★★★	4	✓	✓	✓
	无人机飞行规划	★★★★★	4	✓	✓	✓
	无人机目标打击	★★★★★	4	✓	✓	✓
	无人机语音控制飞行	★★★★★	4	✓	✓	✓
	无人机视觉控制与导航	★★★★★	4	✓	✓	✓
	无人机强化学习与实践	★★★★★	4	✓	✓	✓
	...	...	...	...	...	...
		目标检测	★★★★	4	✓	✓
	物体跟踪	★★★★	4	✓	✓	✓
	视觉定位	★★★★	4	✓	✓	✓
	车道检测与识别	★★★★	4	✓	✓	✓
	智能泊车	★★★★★	4	✓	✓	✓
	交通标志识别	★★★★	4	✓	✓	✓
自动驾驶	行人识别	★★★★	4	✓	✓	✓
	周围车辆识别	★★★★	4	✓	✓	✓

更多信息，访问：[www.educg.net/robot.html](http://www.educg.net/robot.html)

## 课程体系与实验列表

ROS编程与实践	机器人系统仿真
机器人运动控制	机器人感知系统
机器人运动规划	机器人自主导航
机械臂智能控制	无人机智能飞行控制
自动驾驶	无人小车智能控制
...	...

自动驾驶	智能泊车	★★★★★	4	✓	✓	✓
	交通标志识别	★★★	4	✓	✓	✓
	行人识别	★★★	4	✓	✓	✓
	周围车辆识别	★★★	4	✓	✓	✓
	天气感知	★★★	4	✓	✓	✓
	路径规划	★★★★★	4	✓	✓	✓
	...	...	...	...	...	...
无人小车智能控制	Unity 仿真环境介绍	★★	2	✓	✓	✓
	无人驾驶小车控制	★★	2	✓	✓	✓
	摄像头数据采集	★★	2	✓	✓	✓
	无人驾驶模型搭建与训练	★★★★★	4	✓	✓	✓
	训练模型控制的无人驾驶	★★★	2	✓	✓	✓
...	...	...	...	...	...	
智能控制与机器人综合应用案例	机械臂跳舞	★★★★★★	8	✓	✓	✓
	移动机械臂物体智能分拣	★★★★★★	8	✓	✓	✓
	酒店机器人送餐系统	★★★★★★	8	✓	✓	✓
	无人机编队飞行	★★★★★★	8	✓	✓	✓
	...	...	...	...	...	...

更多信息，访问：[www.educg.net/robot.html](http://www.educg.net/robot.html)



05

---

**区块链专业**

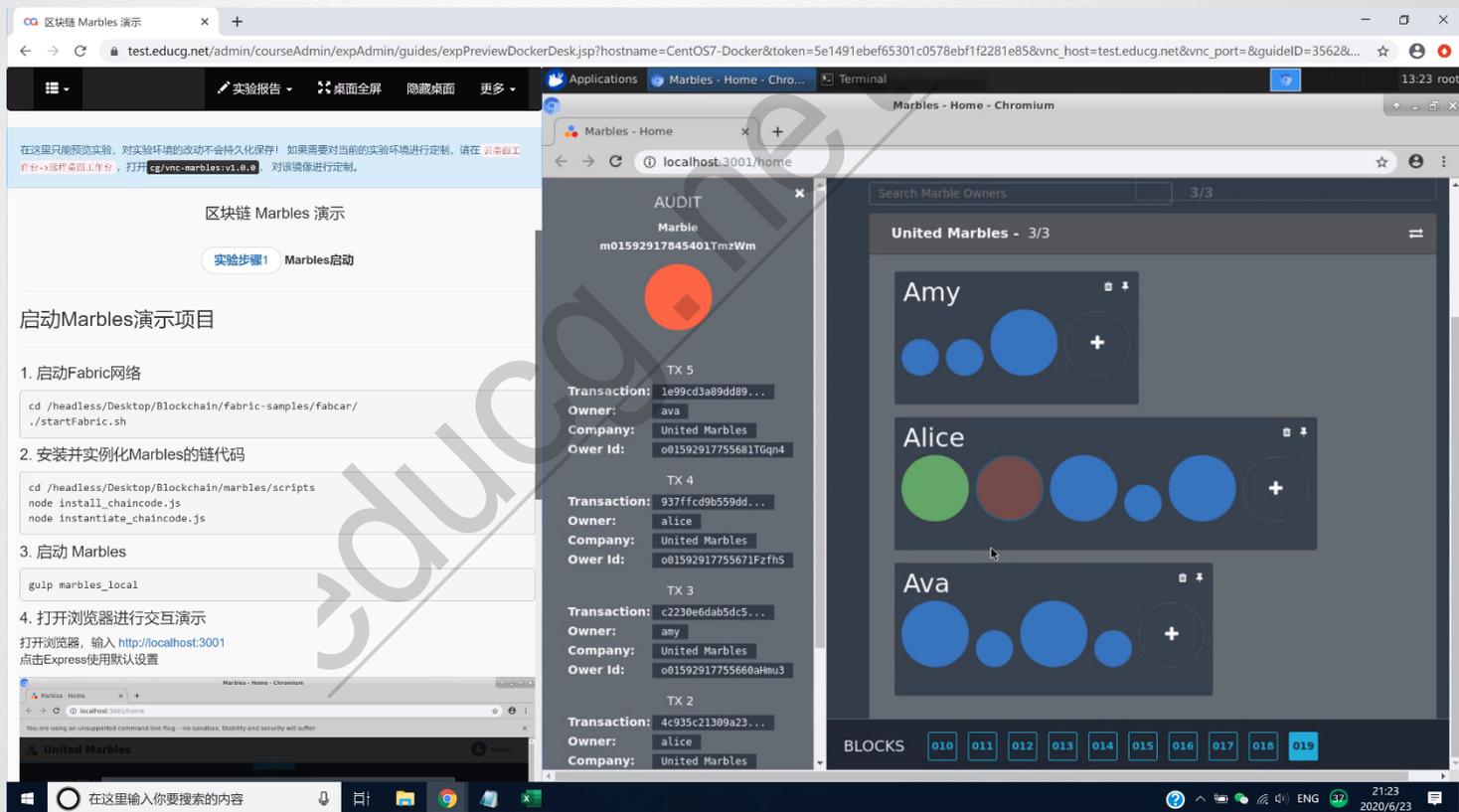
# 区块链在线实验环境

---

随时随地在线实验

## • 随时随地在线实验

基于希冀云桌面实验平台，为学生提供了“开箱即用”的区块链在线实验环境。实验页面分左右两栏，左栏为实验指导手册，右栏为云桌面实验环境。学生可根据左栏的实验指导手册，在右栏的云桌面实验环境中开展区块链实验。



The screenshot displays a web-based interface for a blockchain experiment. The left sidebar contains the following instructions:

- 1. 启动Fabric网络  

```
cd /headless/Desktop/Blockchain/fabric-samples/fabcar/  
./startFabric.sh
```
- 2. 安装并实例化Marbles的链代码  

```
cd /headless/Desktop/Blockchain/marbles/scripts  
node install_chaincode.js  
node instantiate_chaincode.js
```
- 3. 启动 Marbles  

```
gulp marbles_local
```
- 4. 打开浏览器进行交互演示  
打开浏览器，输入 `http://localhost:3001`  
点击Express使用默认设置

The right main area shows a terminal window with the following output:

```
AUDIT  
Marble  
m01592917845401Tm:Wm  
  
TX 5  
Transaction: 1e99cd3a89dd89...  
Owner: ava  
Company: United Marbles  
Owner Id: o01592917755681TGqn4  
  
TX 4  
Transaction: 937ffcd9b559dd...  
Owner: alice  
Company: United Marbles  
Owner Id: o01592917755671Fzfh5  
  
TX 3  
Transaction: c2230e6dab5dc5...  
Owner: amy  
Company: United Marbles  
Owner Id: o01592917755660AHmH3  
  
TX 2  
Transaction: 4c935c21309a23...  
Owner: alice  
Company: United Marbles
```

Below the terminal is a visual representation of the blockchain network, titled "United Marbles - 3/3". It shows three nodes: Amy, Alice, and Ava. Each node is represented by a circle of a different color (blue, green, and blue respectively) and a set of smaller circles representing its neighbors. The nodes are connected in a network structure. At the bottom, there is a "BLOCKS" section with a sequence of blocks numbered 010 to 019.

## • 内置丰富实验环境

希冀区块链在线实验环境为区块链工程专业的每门课程都提供了云桌面实验环境镜像。在实验环境镜像内安装了该课程各实验所需的软件环境，集成了各实验的支撑材料。通过希冀云桌面实验平台，提供的区块链实验环境包括：

Geth, 轻量级以太坊客户端

Mist, 带有图形界面的以太坊客户端

Go, Go语言编译器及开发环境

SoLC, Solidity编译器

Solc-js, Solidity编译器

Web3.js, 基于以太坊的JavaScript API

Marbles Demo, 模拟区块链交易的弹珠游戏

IPFS, 去中心化的文件系统

MetaMask, 轻量级以太坊钱包

Remix IDE, Solidity集成开发环境

HyperLedger Fabric, Linux基金会创建的开源区块链分布式账本

blockchain-explorer, 区块链浏览器

Truffle, 以太坊开发框架

Ganache, 以太坊私有链

Ganache-cli, Ganache命令行版本

OpenZeppelin, 智能合约代码库

Embark, 以太坊DAPP框架

Tierion, 区块链数据验证平台

Ganache-cli, Ganache命令行版本

OpenZeppelin, 智能合约代码库

Embark, 以太坊DAPP框架

Tierion, 区块链数据验证平台

Solium, Solidity代码漏洞检查

VIBES, 大规模区块链网络仿真器

SimBlock, 事件驱动的区块链仿真器

Bitcoin Simulator, 命令行比特币网络仿真器

BLOCKBENCH, 私有区块链仿真与性能评测框架

WebPack, 模块打包器

JDK, Java开发环境

VSCode, 多语言集成开发环境

Node.js, 运行在服务端的JavaScript

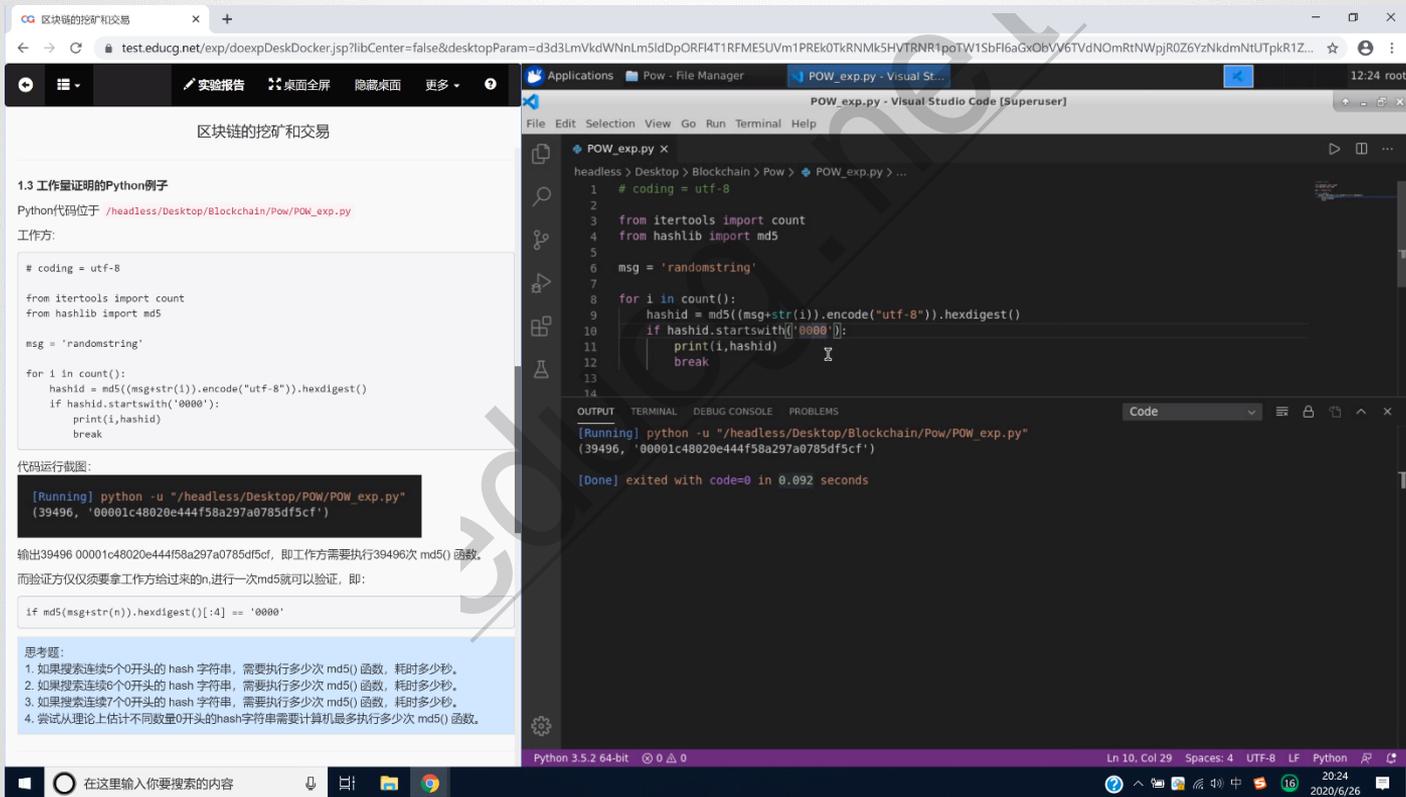
Python, Python语言运行环境

PyCharm, Python语言集成开发环境

...

## • 支持开展多种类型区块链实验

希冀区块链在线实验环境针对公有链、联盟链、私有链等不同类型的区块链，分别提供了对应的实验环境。



区块链的挖矿和交易

### 1.3 工作量证明的Python例子

Python代码位于 `/headless/Desktop/Blockchain/Pow/POW_exp.py`

工作方:

```
# coding = utf-8

from itertools import count
from hashlib import md5

msg = 'randomstring'

for i in count():
    hashid = md5((msg+str(i)).encode("utf-8")).hexdigest()
    if hashid.startswith('0000'):
        print(i,hashid)
        break
```

代码运行截图:

```
[Running] python -u "/headless/Desktop/POW/POW_exp.py"
(39496, '00001c48020e444f58a297a0785df5cf')
```

输出39496 00001c48020e444f58a297a0785df5cf, 即工作需要执行39496次 md5() 函数。  
而验证方仅仅需要拿工作方给过来的n,进行一次md5就可以验证, 即:

```
if md5(msg+str(n)).hexdigest()[0:4] == '0000'
```

思考题:

1. 如果搜索连续5个0开头的 hash 字符串, 需要执行多少次 md5() 函数, 耗时多少秒。
2. 如果搜索连续6个0开头的 hash 字符串, 需要执行多少次 md5() 函数, 耗时多少秒。
3. 如果搜索连续7个0开头的 hash 字符串, 需要执行多少次 md5() 函数, 耗时多少秒。
4. 尝试从理论上估计不同数量0开头的hash字符串需要计算机最多执行多少次 md5() 函数。

```
POW_exp.py X
headless > Desktop > Blockchain > Pow > POW_exp.py > ...
1 # coding = utf-8
2
3 from itertools import count
4 from hashlib import md5
5
6 msg = 'randomstring'
7
8 for i in count():
9     hashid = md5((msg+str(i)).encode("utf-8")).hexdigest()
10    if hashid.startswith('0000'):
11        print(i,hashid)
12        break
13
14

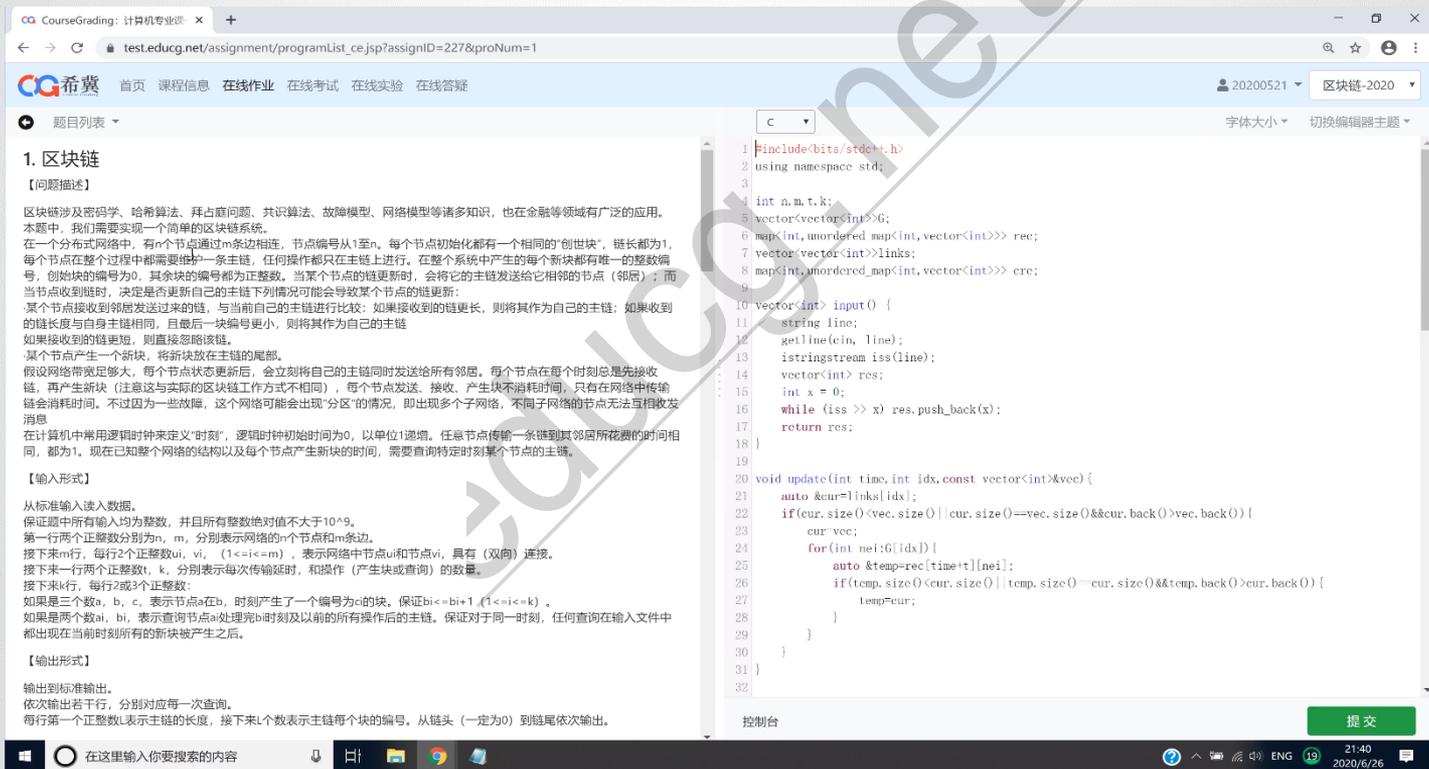
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS Code
[Running] python -u "/headless/Desktop/Blockchain/Pow/POW_exp.py"
(39496, '00001c48020e444f58a297a0785df5cf')

[Done] exited with code=0 in 0.092 seconds

Python 3.5.2 64-bit @ 0 A 0 Ln 10, Col 29 Spaces: 4 UTF-8 LF Python 20:24 2020/6/26
```

## 提供程序自动评判实验环境

区块链工程专业要求学生具备扎实的编程能力，基于希冀的程序自动评判实验环境可发布程序设计相关的实验、作业与考试。



The screenshot displays the CourseGrading web interface. On the left, a sidebar shows the problem list with '1. 区块链' selected. The main content area contains the problem description for 'Blockchain', including a detailed text description of a distributed network and a '【输入形式】' section. On the right, a code editor shows a C++ program for a blockchain simulation. The code includes headers, variable declarations, and functions for input, update, and output. A '提交' (Submit) button is visible at the bottom right of the code editor.

**1. 区块链**

【问题描述】

区块链涉及密码学、哈希算法、拜占庭问题、共识算法、故障模型、网络模型等诸多知识，也在金融等领域有广泛的应用。本题中，我们需要实现一个简单的区块链系统。

在一个分布式网络中，有 $n$ 个节点通过 $m$ 条边相连，节点编号从1至 $n$ 。每个节点初始化都有一个相同的“创世块”，链长都为1。每个节点在整个过程中都需要维护一条主链。任何操作都只在主链上进行。在整个系统中产生的每个新块都有唯一的整数编号，创始块的编号为0。其余块的编号都为正整数。当某个节点的链更新时，会将其的主链发送给它相邻的节点（邻居）；而当节点收到链时，决定是否更新自己的主链下列情况可能会导致某个节点的链更新：

- 某个节点接收到邻居发送过来的链，与当前自己的主链进行比较：如果接收到的链更长，则将其作为自己的主链；如果收到的链长度与自身主链相同，且最后一块编号更小，则将其作为自己的主链
- 如果接收到的链更短，则直接忽略该链。
- 某个节点产生一个新块，将新块放在主链的尾部。

假设网络带宽足够大，每个节点状态更新后，会立刻将自己的主链同时发送给所有邻居。每个节点在每个时刻总是先接收链，再产生新块（注意这与实际的区块链工作方式不相同），每个节点发送、接收、产生块不消耗时间。只有在网络中传输链会消耗时间。不过因为一些故障，这个网络可能会出现“分区”的情况，即出现多个子网络，不同子网络的节点无法互相收发消息

在计算机中常用逻辑时钟来定义“时刻”，逻辑时钟初始时间为0，以单位1递增。任意节点传输一条链到其邻居所花费的时间相同，都为1。现在已知整个网络的结构以及每个节点产生新块的时间，需要查询特定时刻某个节点的主链。

【输入形式】

从标准输入读入数据。

保证题目中所有输入均为整数，并且所有整数绝对值不大于 $10^9$ 。

第一行两个正整数分别为 $n, m$ ，分别表示网络的 $n$ 个节点和 $m$ 条边。

接下来 $m$ 行，每行2个正整数 $u, v$  ( $1 \leq u < v \leq n$ )，表示网络中节点 $u$ 和节点 $v$ ，具有（双向）连接。

接下来一行两个正整数 $k, k'$ ，分别表示每次传输延时，和操作（产生块或查询）的数量。

接下来 $k$ 行，每行2个正整数 $a, b$ ，表示节点 $a$ 在 $b$ 时刻产生了一个编号为 $c$ 的块。保证 $bic = bi + 1$  ( $1 \leq i \leq k$ )。

如果是三个数 $a, b, c$ ，表示节点 $a$ 在 $b$ 时刻产生了一个编号为 $c$ 的块。保证 $bic = bi + 1$  ( $1 \leq i \leq k$ )。

如果是两个数 $a, b$ ，表示查询节点 $a$ 处理完 $b$ 时刻及以前的所有操作后的主链。保证对于同一时刻，任何查询在输入文件中都出现在当前时刻所有的新块被产生之后。

【输出形式】

输出到标准输出。

依次输出若干行，分别对应每一次查询。

每行第一个正整数 $L$ 表示主链的长度，接下来 $L$ 个数表示主链每个块的编号。从链头（一定为0）到链尾依次输出。

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int n, m, t, k;
5 vector<vector<int>>> G;
6 map<int, unordered_map<int, vector<int>>>> rec;
7 vector<vector<int>>> links;
8 map<int, unordered_map<int, vector<int>>>> cre;
9
10 vector<int> input() {
11     string line;
12     getline(cin, line);
13     istringstream iss(line);
14     vector<int> res;
15     int x = 0;
16     while (iss >> x) res.push_back(x);
17     return res;
18 }
19
20 void update(int time, int idx, const vector<int>&vec) {
21     auto &cur = links[idx];
22     if (cur.size() < vec.size() || cur.size() == vec.size() && cur.back() > vec.back()) {
23         cur = vec;
24         for (int nei: G[idx]) {
25             auto &tmp = rec[time + t][nei];
26             if (tmp.size() < cur.size() || tmp.size() == cur.size() && tmp.back() > cur.back()) {
27                 tmp = cur;
28             }
29         }
30     }
31 }
32
```

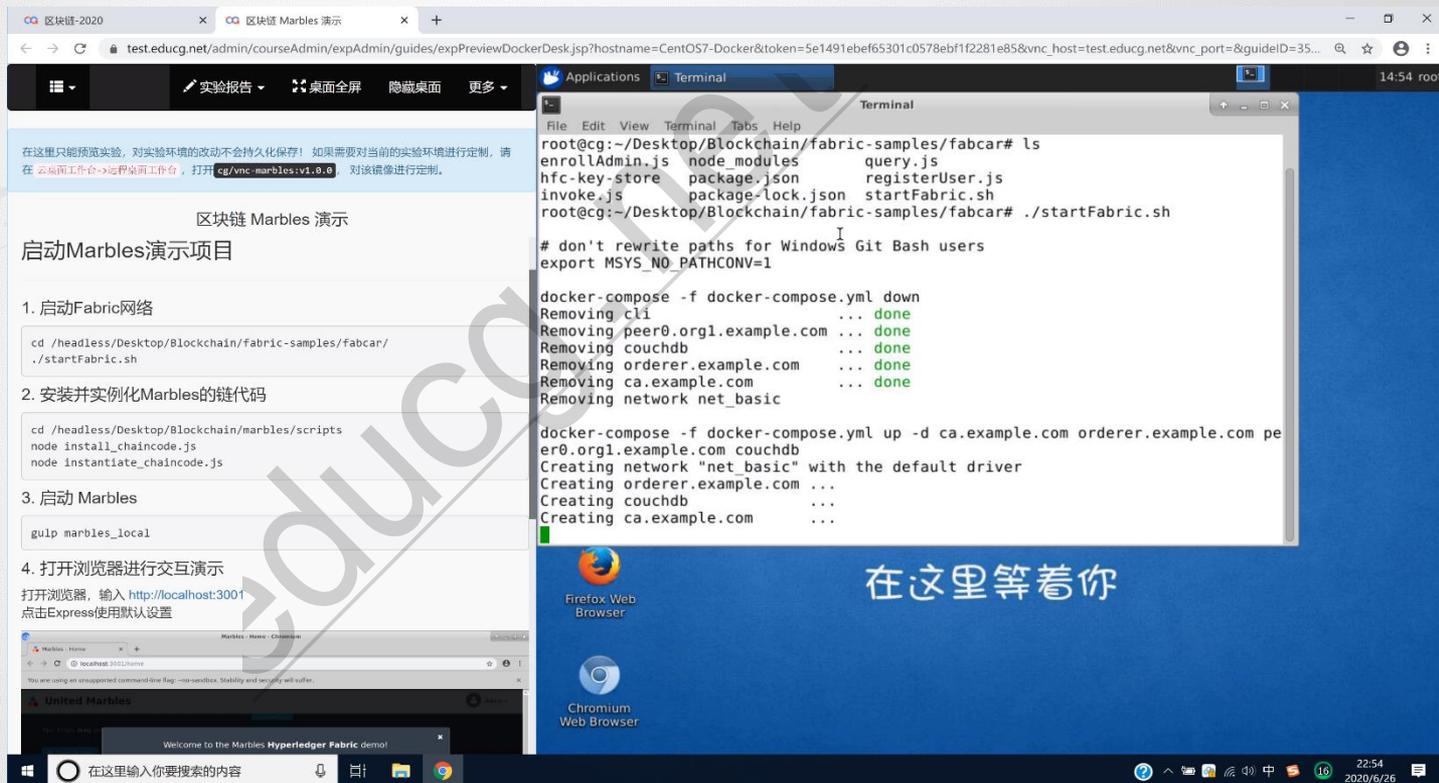
# 分布式区块链实验环境

---

运行真实的区块链分布式系统

## 提供基于容器的分布式实验环境

学生可在云桌面实验环境中，基于 Docker 容器云快速编排和构建面向集群的分布式区块链系统。基于容器云实验环境，支持学生采用开源区块链项目代码构建多节点的公有链系统、联盟链系统以及私有链系统，学生对所构建的区块链系统拥有完全的掌控权，可创建、销毁、停用、重启该区块链系统，可修改源代码重新编译和构建区块链。



区块链 Marbles 演示

启动 Marbles 演示项目

- 启动 Fabric 网络

```
cd /headless/Desktop/Blockchain/fabric-samples/fabcar/  
./startFabric.sh
```

- 安装并实例化 Marbles 的链代码

```
cd /headless/Desktop/Blockchain/marbles/scripts  
node install_chaincode.js  
node instantiate_chaincode.js
```

- 启动 Marbles

```
gulp marbles_local
```

- 打开浏览器进行交互演示

打开浏览器，输入 <http://localhost:3001>  
点击 Express 使用默认设置

```
root@cg:~/Desktop/Blockchain/fabric-samples/fabcar# ls  
enrollAdmin.js  node_modules  query.js  
hfc-key-store  package.json  registerUser.js  
invoke.js      package-lock.json  startFabric.sh  
root@cg:~/Desktop/Blockchain/fabric-samples/fabcar# ./startFabric.sh  
  
# don't rewrite paths for Windows Git Bash users  
export MSYS_NO_PATHCONV=1  
  
docker-compose -f docker-compose.yml down  
Removing cli ... done  
Removing peer0.org1.example.com ... done  
Removing couchdb ... done  
Removing orderer.example.com ... done  
Removing ca.example.com ... done  
Removing network net_basic  
  
docker-compose -f docker-compose.yml up -d ca.example.com orderer.example.com peer0.org1.example.com couchdb  
Creating network "net_basic" with the default driver  
Creating orderer.example.com ...  
Creating couchdb ...  
Creating ca.example.com ...
```

在这里等着你

Firefox Web Browser

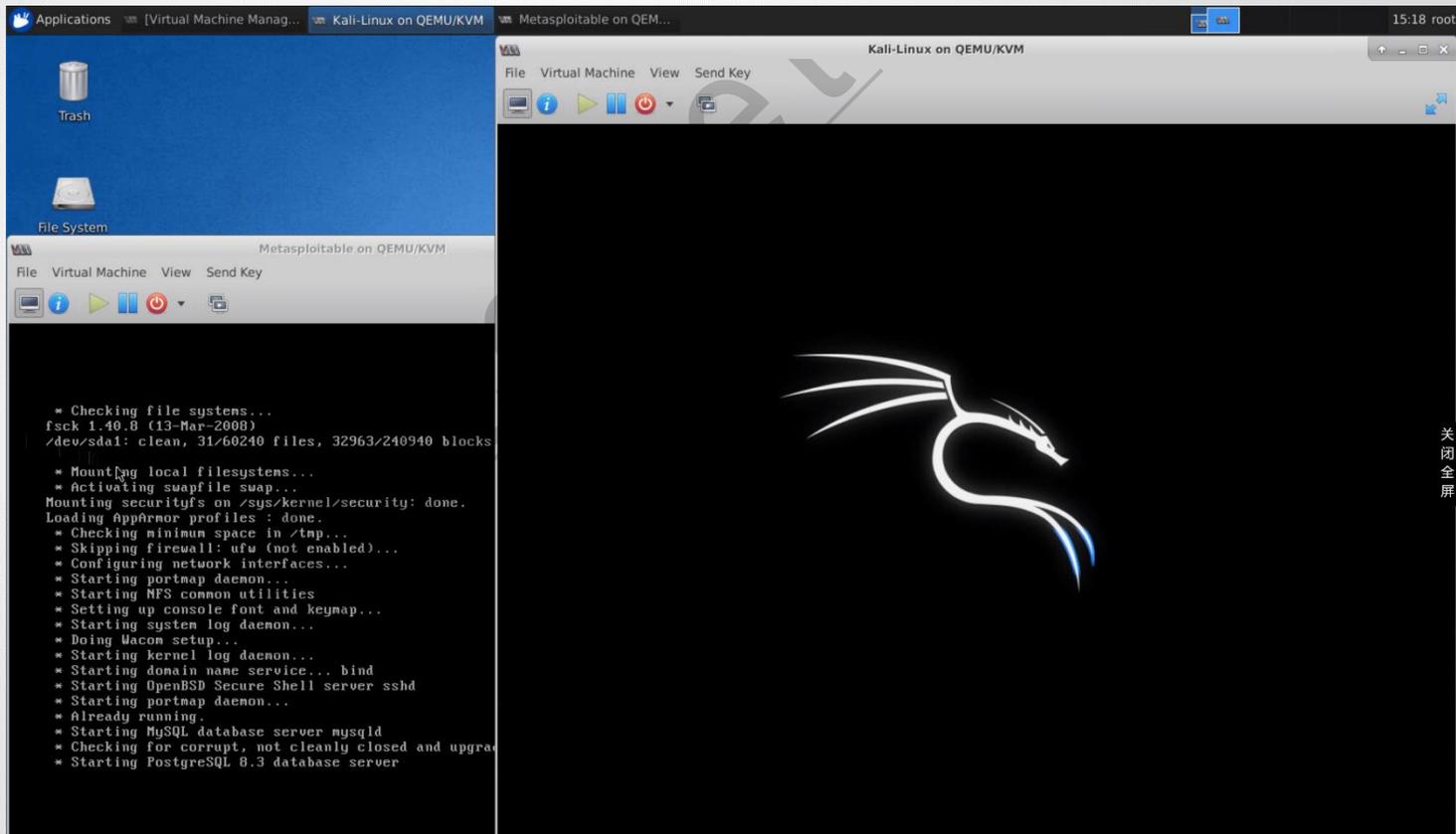
Chromium Web Browser

在这里输入你要搜索的内容

22:54  
2020/6/26

## • 提供基于虚机的分布式实验环境

学生可在云桌面实验环境中，基于多台虚拟机快速构建面向集群的分布式区块链系统。基于由虚拟机组成的集群实验环境，支持学生采用开源区块链项目代码构建多节点的公有链系统、联盟链系统以及私有链系统，学生对所构建的区块链系统拥有完全的掌控权，可创建、销毁、停用、重启该区块链系统，可修改源代码重新编译和构建区块链。



- 验支持区块链专业其他课程实验的开展

基于所提供的容器实验环境和虚拟机实验环境，可支撑学生开展分布式系统、分布式计算、分布式存储、P2P网络、信息安全、网络安全、数字身份认证等相关课程的实验。学生可以用多个容器或虚拟机构建同构或异构的集群计算环境，也可以用多个容器或虚拟机构建多节点的分布式系统。由于在实验环境内提供了已封装好实验软件和实验材料的Docker镜像或虚拟机模板，学生启动集群或分布系统后，可立即得到一个完整的、可用的、可靠的课程实验环境。

# 课程与实验体系

---

一站式支撑区块链专业核心课程

## 课程体系与实验列表

Go语言程序设计	密码学基础原理
分布式系统	区块链原理
区块链系统开发	Solidity程序设计
智能合约	区块链案例实践
...	...

课程	实验名称	实验难度	实验学时	实验手册	实验代码	实验数据
Go语言程序设计	Go语言语法、变量、语句	☆☆	2	✓	✓	✓
	Go语言函数	☆☆☆	2	✓	✓	✓
	Go语言数组、字符串、指针	☆☆☆	2	✓	✓	✓
	Go语言Range和Map	☆☆☆☆	4	✓	✓	✓
	Go语言错误处理	☆☆☆	2	✓	✓	✓
	Go语言并发编程	☆☆☆☆	4	✓	✓	✓
	100道Go语言自动评判编程题	☆☆☆☆☆	50	✓	✓	✓
	...	...	...	...	...	...
密码学基础原理	MD5信息摘要算法原理与实践（分组级联）	☆☆☆	4	✓	✓	✓
	MD5密码暴力破解	☆☆☆☆	4	✓	✓	✓
	RSA算法最大公约数算法和快速幂算法	☆☆☆☆	4	✓	✓	✓
	对称加密算法-AES加解密算法（128）	☆☆☆☆	4	✓	✓	✓
	DES算法密钥编排	☆☆☆☆	4	✓	✓	✓
	AES的轮函数	☆☆☆☆	4	✓	✓	✓
	60道密码学自动评判编程题	☆☆☆☆☆	100	✓	✓	✓
...	...	...	...	...	...	
...	共享内存与消息队列	☆☆	2	✓	✓	✓
	MPI多机多进程通信	☆☆☆☆	4	✓	✓	✓
	CORBA公共对象请求代理体系结构	☆☆☆	2	✓	✓	✓
	Web Service系统	☆☆☆☆	4	✓	✓	✓

更多信息，访问：[www.educg.net/blockchain.html](http://www.educg.net/blockchain.html)

## 课程体系与实验列表

Go语言程序设计	密码学基础原理
分布式系统	区块链原理
区块链系统开发	Solidity程序设计
智能合约	区块链案例实践
...	...

分布式系统	Web Service系统	★★★★	4	✓	✓	✓
	发布订阅系统	★★★	2	✓	✓	✓
	远程调用：RPC与RMI	★★★★	4	✓	✓	✓
	一致性：Paxos与Raft	★★★★★	4	✓	✓	✓
	微架构与REST风格接口	★★★★	4	✓	✓	✓
	...	...	...	...	...	...
区块链原理	弹珠游戏模拟资产转移	★★★	2	✓	✓	✓
	比特币数据结构	★★★	2	✓	✓	✓
	共识协议及系统实现	★★★	2	✓	✓	✓
	挖矿算法及难度调整	★★★★	4	✓	✓	✓
	匿名和隐私保护	★★★★	4	✓	✓	✓
	基于账户的分布式账本	★★★★	4	✓	✓	✓
	状态树、交易树、数据树	★★★★	4	✓	✓	✓
	GHOST协议	★★★★	4	✓	✓	✓
	权益证明	★★★★	4	✓	✓	✓
	智能合约	★★★★	4	✓	✓	✓
	...	...	...	...	...	...
	私有链构建	★★★	2	✓	✓	✓
	联盟链构建	★★★	2	✓	✓	✓
公有链构建	★★★	2	✓	✓	✓	
多机分布式私有链构建	★★★★	4	✓	✓	✓	

更多信息，访问：[www.educg.net/blockchain.html](http://www.educg.net/blockchain.html)

## 课程体系与实验列表

Go语言程序设计	密码学基础原理
分布式系统	区块链原理
区块链系统开发	Solidity程序设计
智能合约	区块链案例实践
...	...

区块链系统开发	联盟链构建	☆☆☆	2	✓	✓	✓
	公有链构建	☆☆☆	2	✓	✓	✓
	多机分布式私有链构建	☆☆☆☆	4	✓	✓	✓
	多机分布式联盟链构建	☆☆☆☆	4	✓	✓	✓
	多机分布式公有链构建	☆☆☆☆	4	✓	✓	✓
	使用Go语言从零开发区块链系统	☆☆☆☆	4	✓	✓	✓
	使用Python语言从零开发区块链系统	☆☆☆☆	4	✓	✓	✓
	使用Java语言从零开发区块链系统	☆☆☆☆	4	✓	✓	✓
	...	...	...	...	...	...
Solidity程序设计	Solidity开发环境配置与选择	☆☆☆	2	✓	✓	✓
	以太坊虚拟机	☆☆☆☆	4	✓	✓	✓
	Solidity注释、数据类型、变量、语句	☆☆☆	2	✓	✓	✓
	Solidity函数、函数修饰符、视图函数	☆☆☆☆	4	✓	✓	✓
	Solidity纯函数、回退函数、函数重载	☆☆☆☆	4	✓	✓	✓
	Solidity数学函数、加密函数	☆☆☆☆	4	✓	✓	✓
	Solidity内联汇编、事件、错误处理	☆☆☆☆	4	✓	✓	✓
	Solidity程序调试	☆☆☆☆	4	✓	✓	✓
	...	...	...	...	...	...
	合约创建与构造函数	☆☆☆	2	✓	✓	✓
合约组合	☆☆☆	2	✓	✓	✓	
合约的封装、继承、多态	☆☆☆	2	✓	✓	✓	

更多信息，访问：[www.educg.net/blockchain.html](http://www.educg.net/blockchain.html)

## 课程体系与实验列表

Go语言程序设计	密码学基础原理
分布式系统	区块链原理
区块链系统开发	Solidity程序设计
智能合约	区块链案例实践
...	...

Solidity程序设计	以太坊虚拟机	★★★★☆	4	✓	✓	✓
	Solidity注释、数据类型、变量、语句	★★★☆☆	2	✓	✓	✓
	Solidity函数、函数修饰符、视图函数	★★★★☆	4	✓	✓	✓
	Solidity纯函数、回退函数、函数重载	★★★★☆	4	✓	✓	✓
	Solidity数学函数、加密函数	★★★★☆	4	✓	✓	✓
	Solidity内联汇编、事件、错误处理	★★★★☆	4	✓	✓	✓
	Solidity程序调试	★★★★☆	4	✓	✓	✓
...	...	...	...	...	...	...
智能合约	合约创建与构造函数	★★★☆☆	2	✓	✓	✓
	合约组合	★★★☆☆	2	✓	✓	✓
	合约的封装、继承、多态	★★★☆☆	2	✓	✓	✓
	抽象合约、接口、库	★★★★☆	4	✓	✓	✓
	实现Hello World智能合约	★★★★☆	4	✓	✓	✓
	Truffle基础与单元测试	★★★★☆	4	✓	✓	✓
	ERC721合约模板	★★★★☆	4	✓	✓	✓
...	...	...	...	...	...	...
区块链案例实践	基于智能合约的宠物交易项目实战	★★★★★☆☆	8	✓	✓	✓
	基于智能合约的商品拍卖项目实战	★★★★★☆☆	8	✓	✓	✓
	基于智能合约的捐赠系统项目实战	★★★★★☆☆	8	✓	✓	✓
	基于智能合约的保险系统项目实战	★★★★★☆☆	8	✓	✓	✓
	...	...	...	...	...	...

更多信息，访问：[www.educg.net/blockchain.html](http://www.educg.net/blockchain.html)



06

---

**集成电路专业**

- 课程体系

数字集成电路设计

CMOS模拟集成电路设计

VLSI数字信号处理

VLSI数字通信原理与设计

处理器设计

智能处理器设计

FPGA技术

计算机组成原理

电路与Spice

数字电路与Verilog

集成电路专业

## 支持Spice在线实验环境

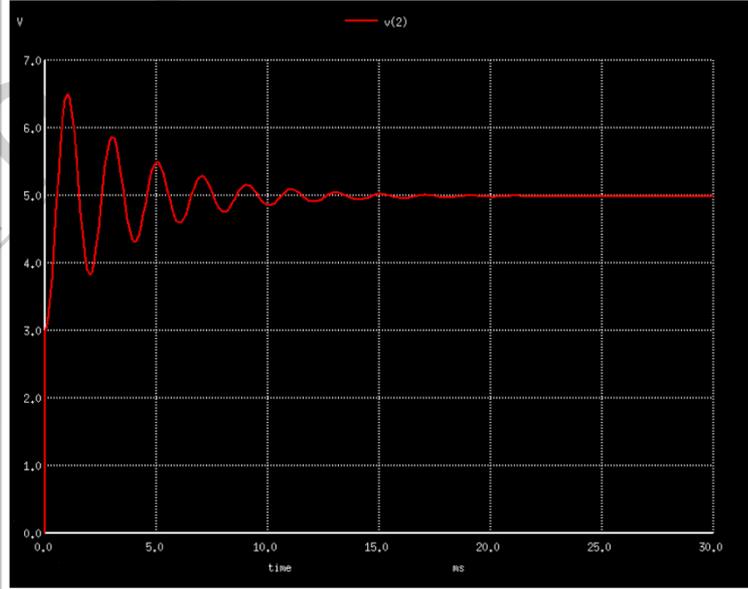
远程桌面工作

远程桌面工作

远程桌面工作

```

R3 3 0 4K
L 2 3 0.2 IC=0.2
C 3 0 0.5U IC=2
.TRAN 0.1M 30M UIC
.END
root@cg:~/Desktop/project/spice# cat ex3.cir
EX3
VIN 1 0 AC 10V
L 1 2 0.2 IC=0
R 3 0 500
C 2 3 0.1U IC=0
.AC DEC 20 100 10K
.END
root@cg:~/Desktop/project/spice# cat ex2.cir
EX2
VS 1 0 10
R1 1 2 20
R2 2 0 20
R3 3 0 4K
L 2 3 0.2 IC=0.2
C 3 0 0.5U IC=2
.TRAN 0.1M 30M UIC
.END
root@cg:~/Desktop/p
    
```



简介

简介

简介

SPICE: SPICE: SPICE: CM

如今, 每一天都件SPICE. SPICE广泛应

如今, 每一天都件SPICE. SPICE广泛应

如今, 每一天都件SPICE. SPICE广泛应

Initial Transient Solution

Node	Voltage
1	0
2	0
3	0
t#branch	0
vs#branch	0

No. of Data Rows : 312

ngspice 1 -> plot v(2)

ngspice 1 -> [ ]

代码

演示程序1:

```

EX1
VIN 1 0 AC 2
R1 1 2 0.45K
R2 2 0 1K
RI 3 0 1MEG
R0 5 4 100
R3 3 4 500
R4 4 0 1K
C1 2 3 4U
C2 2 4 4U
E1 5 0 3 0 5
.AC DEC 20 1
.END
        
```

代码

演示程序1:

```

EX1
VIN 1 0 AC 2V
R1 1 2 0.45K
R2 2 0 1K
RI 3 0 1MEG
R0 5 4 100
R3 3 4 500
R4 4 0 1K
C1 2 3 4U
C2 2 4 4U
E1 5 0 3 0 500
.AC DEC 20 1
.END
        
```

代码

演示程序1:

```

EX1
VIN 1 0 AC 2V
R1 1 2 0.45K
R2 2 0 1K
RI 3 0 1MEG
R0 5 4 100
R3 3 4 500
R4 4 0 1K
C1 2 3 4U
C2 2 4 4U
E1 5 0 3 0 500K
.AC DEC 20 1 10K
.END
        
```

演示程序2:

EX2

演示程序2:

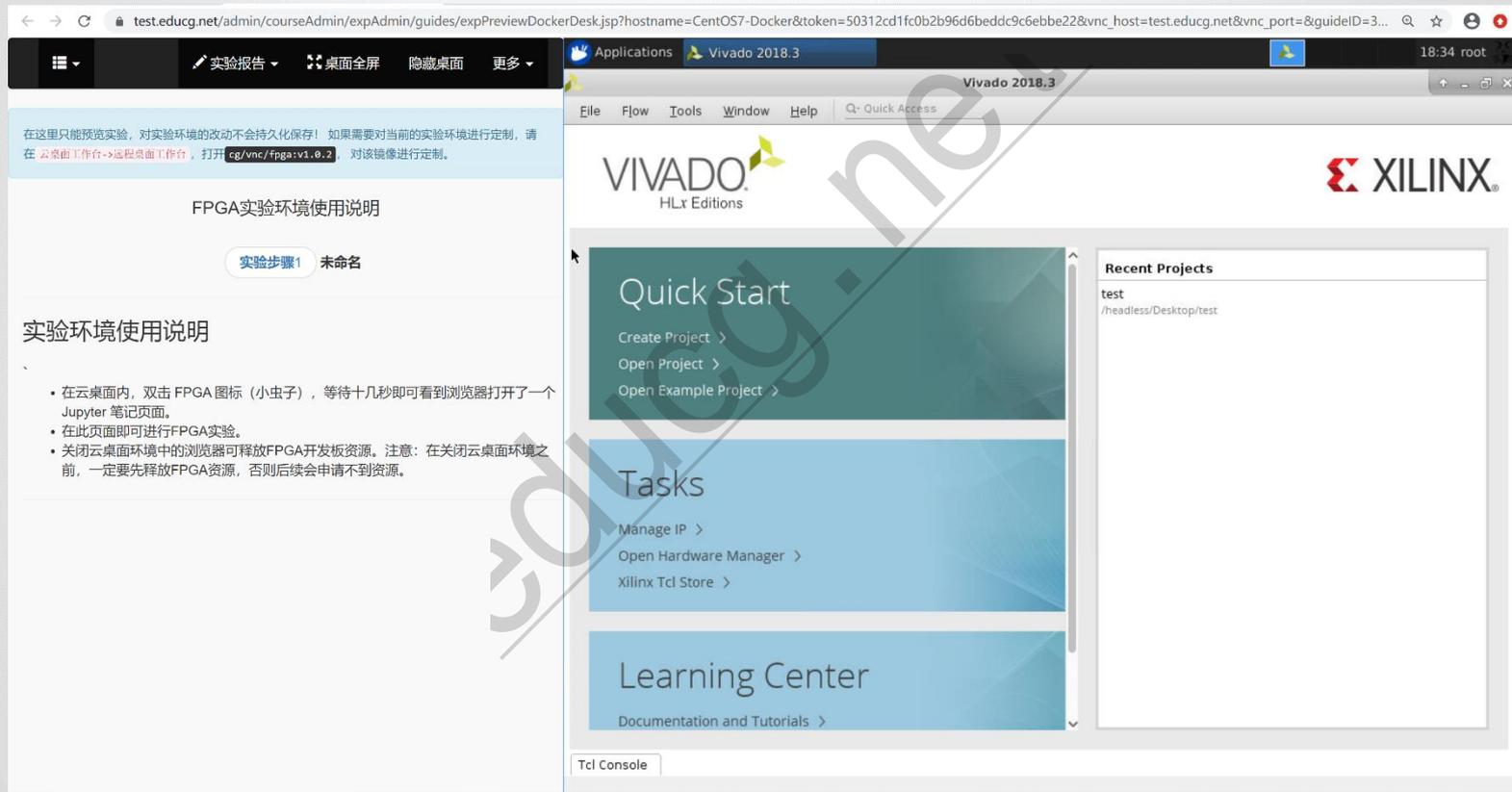
EX2

演示程序2:

EX2

关闭全屏

## • 支持Vivado在线实验环境



test.educg.net/admin/courseAdmin/expAdmin/guides/expPreviewDockerDesk.jsp?hostname=CentOS7-Docker&token=50312cd1fc0b2b96d6beddc9c6ebbe22&vnc\_host=test.educg.net&vnc\_port=&guidelD=3...

Applications Vivado 2018.3 18:34 root

File Fjow Tools Window Help Quick Access

VIVADO HLx Editions XILINX

FPGA实验环境使用说明

实验步骤1 未命名

### 实验环境使用说明

- 在云桌面内，双击 FPGA 图标（小虫子），等待十几秒即可看到浏览器打开了一个 Jupyter 笔记页面。
- 在此页面即可进行FPGA实验。
- 关闭云桌面环境中的浏览器可释放FPGA开发板资源。注意：在关闭云桌面环境之前，一定要先释放FPGA资源，否则后续会申请不到资源。

### Quick Start

- Create Project >
- Open Project >
- Open Example Project >

### Tasks

- Manage IP >
- Open Hardware Manager >
- Xilinx Tcl Store >

### Learning Center

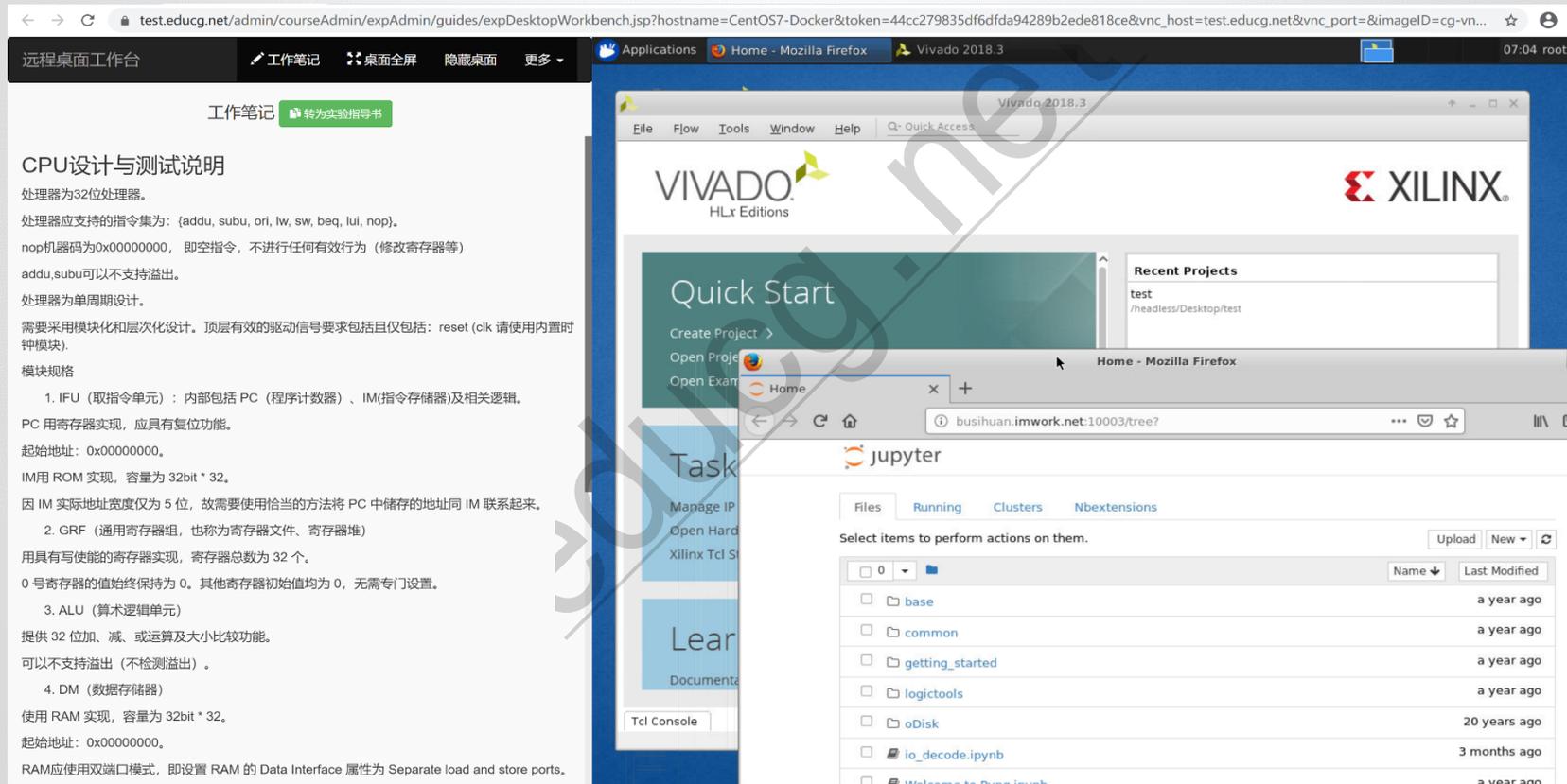
- Documentation and Tutorials >

#### Recent Projects

- test  
/headless/Desktop/test

Tcl Console

## 支持FPGA在线实验环境



The screenshot displays a remote desktop session. On the left, a document titled "工作笔记" (Work Notes) contains technical details about CPU design and testing. On the right, a Jupyter notebook interface is open, showing a file browser with a directory structure including 'base', 'common', 'getting\_started', 'logictools', 'oDisk', 'io\_decode.ipynb', and 'Welcome to Pure IPynb'. The background shows the Vivado 2018.3 software interface with a 'Quick Start' panel and a 'Recent Projects' list.

远程桌面工作台

工作笔记 [转为实验指导书](#)

### CPU设计与测试说明

处理器为32位处理器。  
处理器应支持的指令集为: {addu, subu, ori, lw, sw, beq, lui, nop},  
nop机器码为0x00000000, 即空指令, 不进行任何有效行为 (修改寄存器等)  
addu,subu可以不支持溢出。  
处理器为单周期设计。  
需要采用模块化和层次化设计。顶层有效的驱动信号要求包括且仅包括: reset (clk 请使用内置时钟模块)。  
模块规格

1. IFU (取指令单元): 内部包括 PC (程序计数器)、IM(指令存储器)及相关逻辑。  
PC 用寄存器实现, 应具有复位功能。  
起始地址: 0x00000000。  
IM用 ROM 实现, 容量为 32bit \* 32。  
因 IM 实际地址宽度仅为 5 位, 故需要使用恰当的方法将 PC 中储存的地址同 IM 联系起来。
2. GRF (通用寄存器组, 也称为寄存器文件、寄存器堆)  
用具有写使能的寄存器实现, 寄存器总数为 32 个。  
0 号寄存器的值始终保持为 0。其他寄存器初始值均为 0, 无需专门设置。
3. ALU (算术逻辑单元)  
提供 32 位加、减、或运算及大小比较功能。  
可以不支持溢出 (不检测溢出)。
4. DM (数据存储器)  
使用 RAM 实现, 容量为 32bit \* 32。  
起始地址: 0x00000000。  
RAM应使用双端口模式, 即设置 RAM 的 Data Interface 属性为 Separate load and store ports,

VIVADO HLx Editions XILINX

### Quick Start

Create Project >  
Open Project >  
Open Example >

### Recent Projects

test  
/headless/Desktop/test

### Task

Manage IP  
Open Hardware  
Xilinx Tcl S...

### Learn

Documents

Tcl Console

Jupyter

Files Running Clusters Nnextensions

Select items to perform actions on them.

Name	Last Modified
base	a year ago
common	a year ago
getting_started	a year ago
logictools	a year ago
oDisk	20 years ago
io_decode.ipynb	3 months ago
Welcome to Pure IPynb	a year ago

## 支持Verilog评测

### 当前通用评测题：9091. 8位逐位进位加法器

学习逐位进位加法器的原理，完成8位行逐位进位加法器的设计与功能仿真验证。

顶层模块声明如下：

```
module adder_top(
    input [7:0]a,
    input [7:0] b,
    input cin,
    output [7:0]s,
    output cout
);
```

信号名称	信号作用
a[7:0]	加数
b[7:0]	被加数
cin	进位输入
s[7:0]	和
cout	进位输出

提交源文件，只能提交以 rar、zip 为后缀的文件：

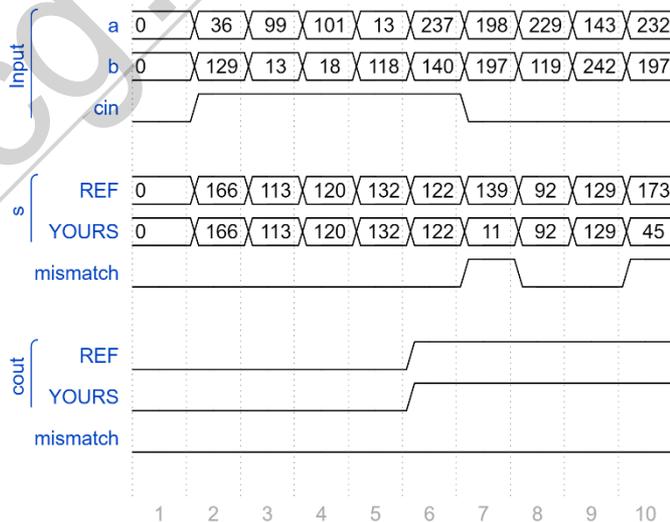
未选择任何文件

[下载源文件](#)

更新时间：2019-09-03 09:23:26

answer error

answer error



## 支持Logisim在线实验环境

远程桌面工作台

工作笔记 转为实验指导书

### CPU设计与测试说明

处理器为32位处理器。

处理器应支持的指令集为：{addu, subu, ori, lw, sw, beq, lui, nop}。

nop机器码为0x00000000，即空指令，不进行任何有效行为（修改寄存器等）

addu,subu可以不支持溢出。

处理器为单周期设计。

需要采用模块化和层次化设计。顶层有效的驱动信号要求包括且仅包括：reset (clk 请使用内置时钟模块)。

模块规格

- IFU (取指令单元)：内部包括 PC (程序计数器)、IM(指令存储器)及相关逻辑。
- GRF (通用寄存器组，也称为寄存器文件、寄存器组)
- ALU (算术逻辑单元)

PC用寄存器实现，应具有复位功能。

起始地址：0x00000000。

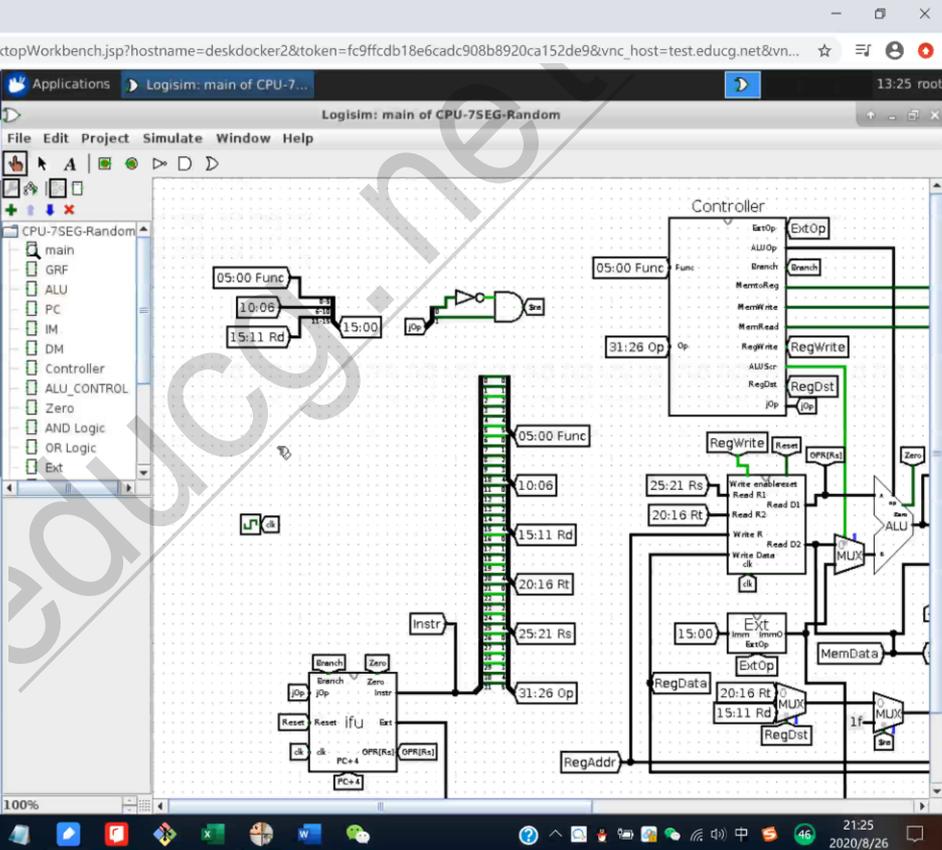
IM用ROM实现，容量为 32bit \* 32。

因 IM 实际地址宽度仅为 5 位，故需要使用恰当的方法将 PC 中存储的地址同 IM 联系起来。

0 号寄存器的值始终保持为 0。其他寄存器初始值均为 0，无需专门设置。

可以提供 32 位加、减、或运算及大小比较功能。

可以不支持溢出（不检测溢出）。



The screenshot shows a Logisim project titled "Logisim: main of CPU-7SEG-Random". The circuit diagram is a complex digital logic design for a CPU. It features several main components:
 

- IFU (Instruction Fetch Unit):** Located at the bottom left, it includes a PC register, a branch/jump control logic, and an instruction memory (IM) connected to a ROM.
- GRF (General Register File):** A central vertical bus structure that manages 32 registers. It includes control logic for writing data to registers and reading data from them.
- ALU (Arithmetic Logic Unit):** A large block on the right that performs operations on data from registers. It includes control logic for selecting the operation (add, subtract, etc.) and for setting flags like Zero.
- Controller:** A block at the top right that coordinates the flow of data and control signals between the IFU, GRF, and ALU.

## 支持Logisim评测

当前通用评测题: 33834. 斐波那契数列 (64个周期内计算出

黄小板同学在沙河东门半导体公司的工作终于到了尾声, 这一周内去计算斐波那契数。这时, 公司负责人提出了自己的看法, 那契数, 说着, 他就开始搭建电路

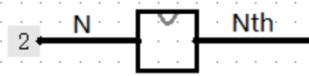
公司负责人开始紧张地思考...

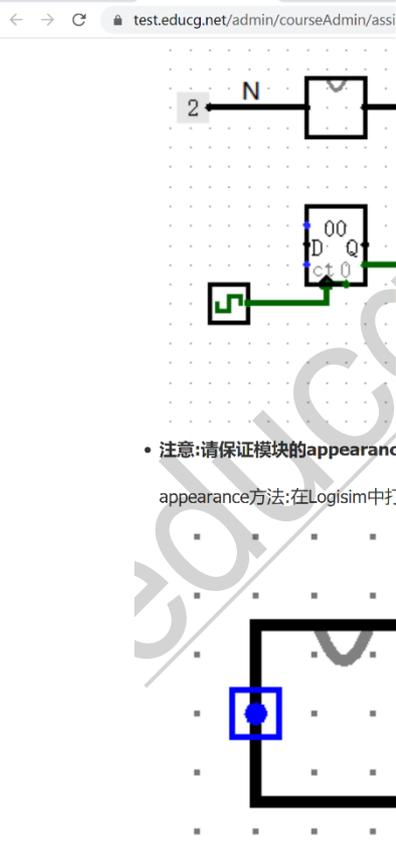
**TO BE CONTINUED...**

### 提交要求

使用Logisim搭建一个根据输入序号x计算对应序号斐波那契数1对应输出数1, 输入序号2对应输出数1, 以此类推) 并提交

- 输入: N (3bit)
- 输出: Nth (4bit)
- 文件内模块名: main
- 测试要求: 每次给定一个固定输入保持不结果未计算出之前输出端口输出0
- 测试电路图:





注意: 请保证模块的appearance  
appearance方法: 在Logisim中打

提交源文件, 只能提交以 `circ` 为后缀的文件:

选择文件 未选择任何文件

**提交** 调试信息

[下载源文件](#) [在线浏览源代码](#) 更新时间: 2020-08-18 13:45:38

WA

**verdict: WA**

**score: 14**

name	verdict	score	time	details
TestCase1	Accept	14.285714285714286	635	expand/collapse
TestCase2	Wrong Answer	0	658	expand/collapse
TestCase3	Wrong Answer	0	718	expand/collapse
TestCase4	Wrong Answer	0	1074	expand/collapse
TestCase5	Wrong Answer	0	1072	expand/collapse
TestCase6	Wrong Answer	0	921	expand/collapse
TestCase7	Wrong Answer	0	608	expand/collapse

## 提供数字逻辑与Verilog自动评测实验体系

实验专题	知识点	实验名称	实现特定电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测							
数字逻辑基础	Verilog 基础	1、输出 0	组合电路	28、简单电路 A	组合电路 2	42、组合电路 2	实现特定组合电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		2、其他的门		29、简单电路 B		43、组合电路 3	实现特定组合电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		3、两个门		30、简单电路 AB		44、组合电路 4	实现特定组合电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		4、wire 类型		31、always-组合逻辑		45、组合电路 5	实现特定组合电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		5、Four wires		32、if 语句		46、组合电路 6	实现特定组合电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		6、非门		33、if 语句引发锁存		47、三个模块-移位寄存器	实现特定时序电路。	掌握	2	2	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		7、与门		34、case 语句		48、模块与 vector	实现特定时序电路。	掌握	2	2	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		8、或非门		35、简单编码器		49、D 触发器	实现特定时序电路。	掌握	2	2	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
				36、避免锁存		50、D 触发器组合	实现特定时序电路。	掌握	2	2	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
				37、2 位计数器		51、异步复位触发器	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
						52、同步复位触发器	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
			89、字符自动机-Verrilog	实现特定电路。	掌握	4	4	提供	√	cg/judge/ise:1.0.1	通用评测						
			90、格雷码计数器	实现特定电路。	掌握	2	3	提供	√	cg/judge/ise:1.0.1	通用评测						
			91、表达式状态机	实现特定电路。	掌握	2	3	提供	√	cg/judge/ise:1.0.1	通用评测						
			92、表达式状态机-允许括号	实现特定电路。	掌握	4	4	提供	√	cg/judge/ise:1.0.1	通用评测						
组合电路	Verilog 基础	16、vector 连接	时序逻辑	44、组合电路 4	组合电路 3	60、多路选择器与触发器 2	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		17、vector 反转		45、组合电路 5		61、根据真值表实现电路	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		18、模块		46、组合电路 6		62、上升沿检测	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		19、按位置连接端口		47、三个模块-移位寄存器		63、时序电路 7	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		20、按名称连接端口		48、模块与 vector													
		21、三目算符		49、D 触发器		64、时序电路 8	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		22、优化运算 1		50、D 触发器组合		65、纠错 键盘数字检测	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		23、优化运算 2		51、异步复位触发器		66、算数左右移	实现特定时序电路。	掌握	2	3	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		24、或非门		52、同步复位触发器		67、 fsm1	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		25、两位相等		53、同步复位触发器组		68、 fsm1s	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		26、tb 时钟		54、同步复位特殊值触发器组		69、 fsm3comb	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		27、tb 与门		55、可异步复位的触发器组		70、 fsm3	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		28、简单电路 A		56、带控制端的触发器组		71、 鼠标 1	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		29、简单电路 B		57、D 锁存器		72、 鼠标 2	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		30、简单电路 AB		58、门控触发器		73、 鼠标 3	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		31、always-组合逻辑		59、多路选择器与触发器 1		74、 鼠标 4	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		32、if 语句		60、多路选择器与触发器 2		75、序列识别	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		33、if 语句引发锁存		61、根据真值表实现电路		76、验证 1000 序列 (Moore)	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
		34、case 语句		62、上升沿检测		77、验证 1000 序列 (Mealy)	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测			
	63、时序电路 7	78、验证 1101 序列 (Mealy)	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测							
		79、验证 1101 序列 (Moore)	实现特定状态机。	掌握	2	4	提供	√	cg/judge/verilog-judge:1.0.2	通用评测							

## 提供数字部件设计自动评测实验体系

实验专题	知识点	实验名称	实验任务	考核要求	学时	难度	参考答案	自动评测	镜像名称	实验环境	
数字部件设计	计数器	1、四位二进制计数器	实现特定计数器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		2、十进制计数器 0	实现特定计数器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		3、十进制计数器 1	实现特定计数器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		4、慢速十进制计数器	实现特定计数器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		5、counter1-12	实现特定计数器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		6、计数器 1000	实现特定计数器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		7、四位移位寄存器和递减计数器	实现特定计数器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
	加法器	8、加法器 1	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		9、加法器 2	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		10、进位选择加法器	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		11、加减法器	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		12、半加器	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		13、全加器	实现特定加法器。	掌握	2	3	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		14、3 位加法器	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		15、adder 练习	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		16、溢出检测	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		17、bcd 加法器	实现特定加法器。	掌握	2	3	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		18、改错_加法器减法器	实现特定加法器。	掌握	2	2	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		19、32 位逐位进位加法器	实现特定加法器。	掌握	2	3	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		20、32 位选择进位加法器	实现特定加法器。	掌握	2	3	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		21、32 位超前进位加法器	实现特定加法器。	掌握	2	3	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
		乘法器	22、8 位乘法器	实现特定乘法器。	掌握	2	3	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
			23、8 位流水乘法器	实现特定乘法器。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
数字部件综合设计	24、4 位比较器	实现特定比较器。	掌握	2	3	提供	√	cg/judge/ise:1.0.1	通用评测		
	25、4 位全加器	实现特定加法器。	掌握	2	3	提供	√	cg/judge/logisim:v1.0.4	通用评测		
	26、4 位加法器	实现特定加法器。	掌握	2	3	提供	√	cg/judge/ise:1.0.1	通用评测		
	27、4 位 ALU	实现特定 ALU。	掌握	2	3	提供	√	cg/judge/logisim:v1.0.4	通用评测		
	28、32 位六运算 ALU	实现特定 ALU。	掌握	2	3	提供	√	cg/judge/ise:1.0.1	通用评测		

## 提供处理器设计自动评测实验体系

CPU 设计	29、实验 1: 取指单元	实现 CPU 取指单元。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	30、实验 2: 寄存器堆	实现 CPU 寄存器堆。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	31、实验 3: ALU	实现 CPU 的算术逻辑运算器。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	32、实验 4: DataMem	实现 CPU 的数据存储器。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	33、实验 5: ALU 控制模块	实现 CPU 的 ALU 控制模块。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	34、实验 6: 控制器	实现 CPU 的控制器。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	35、实验 7: CPU	实现完整的 CPU。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
CPU 设计	36、寄存器堆(RegFile)	实现寄存器堆。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	37、取指级(PC)的基本实现	实现取指。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	38、取指-译码中间级(IFID)	实现取指-译码中间级。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	39、译码级(ID)的基本实现	实现译码级。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	40、执行级(EX)的基本实现	实现执行级。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	41、访存级(MEM)的基本实现	实现访存级。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	42、写回级(WB)的基本实现	实现写回级。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	43、解决寄存器读写冲突	解决寄存器读写冲突。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	44、让取指级(PC)支持分支	让取指级支持分支。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	45、让译码级(ID)支持分支	让译码级支持分支。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	46、让译码级(ID)支持访存	让译码级支持访存。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	47、让执行级(EX)支持访存	让执行级支持访存。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	48、访存级(MEM)的完整实现	实现完整的访存级。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	49、写回级(WB)的完整实现	实现完整的写回级(WB)。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	50、让取指级(PC)支持暂停	让取指级支持暂停。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
51、译码级(ID)冲突检测	实现译码级冲突检测。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
52、执行级(EX)冲突检测	实现执行级冲突检测。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测	
CPU 综合设计	53、访存级(MEM)冲突检测	实现访存级冲突检测。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	54、解决访存冲突	解决访存冲突。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	55、实现流水线控制器	实现流水线控制器。	掌握	2	4	提供	√	cg/judge/verilog-judge: 1.0.2	通用评测
	56、单周期 CPU (logisim)	实现单周期 CPU。	掌握	8	4	提供	√	cg/judge/logisim:v1.0.4	通用评测
	57、单周期 CPU (verilog)	实现单周期 CPU。	掌握	8	4	提供	√	cg/judge/ise:1.0.1	通用评测
	58、多周期 CPU (MIPS-lite4 指令集)	实现多周期 CPU。	掌握	16	5	提供	√	cg/judge/ise:1.0.1	通用评测
	59、多周期 CPU (MIPS-C3 指令集)	实现多周期 CPU。	掌握	16	5	提供	√	cg/judge/ise:1.0.1	通用评测

## 提供智能处理器设计实验体系

实验概览 / 共 5 / 5 个实验

排序 ▾



远程桌面



Jupyter实验

实验专题 ▾

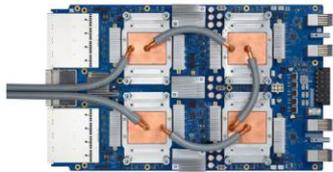
批量导入实验

重置实验环境

更多 ▾

远程桌面工作台 ▾ ?

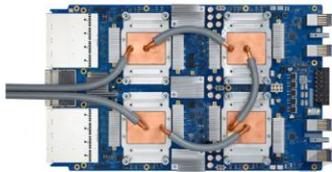
Jupyter工作台 ▾ ?



实验5: 运行MLP图像分类应用

删除

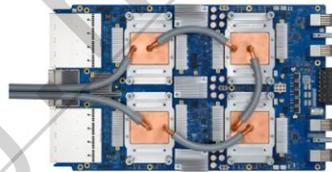
编辑



实验4: MyTPU整体设计与性能评估

删除

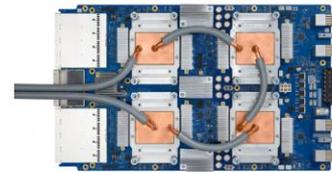
编辑



实验3: 指令并行和数据并行

删除

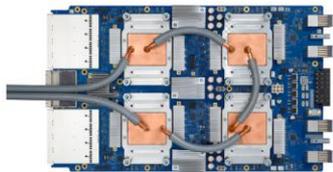
编辑



实验2: 脉动阵列及其实现

删除

编辑



实验1: 谷歌TPU概述和简化

删除

编辑

07

---

**金融科技专业**

- 课程体系



- 对程序设计类课程提供自动评测环境

```

程序片段编程题样例
rng = numpy.random

# Parameters
learning_rate = 0.01
training_epochs = 1000
display_step = 50

# Training Data
train_X = numpy.asarray([3.3,4.4,5.5,6.71,6.93,4.168,9.779,6.182,7.59,2.167,
7.042,10.791,5.313,7.997,5.654,9.27,3.1])
train_Y = numpy.asarray([1.7,2.76,2.09,3.19,1.694,1.573,3.366,2.596,2.53,1.
2.827,3.465,1.65,2.904,2.42,2.94,1.3])
n_samples = train_X.shape[0]

# tf Graph Input
X = tf.placeholder("float")
Y = tf.placeholder("float")

# Set model weights
W = tf.Variable(rng.randn(), name="weight")
b = tf.Variable(rng.randn(), name="bias")

# Construct a linear model
pred = tf.add(tf.multiply(X, W), b)

# Mean squared error
cost = tf.reduce_sum(tf.pow(pred-Y, 2))/(2*n_samples)

# Gradient descent
# Note minimize() knows to modify W and b because Variable objects are tra

```

```

程序片段编程题样例

print("Optimization Finished!")
training_cost = sess.run(cost, feed_dict={X: train_X, Y: train_Y})
print("Training cost=", training_cost, "W=", sess.run(W), "b=", sess.run(b), '\n')

# Testing example, as requested (Issue #2)
test_X = numpy.asarray([6.83, 4.668, 8.9, 7.91, 5.7, 8.7, 3.1, 2.1])
test_Y = numpy.asarray([1.84, 2.273, 3.2, 2.831, 2.92, 3.24, 1.35, 1.03])

print("Testing... (Mean square loss Comparison)")
testing_cost = sess.run(
    tf.reduce_sum(tf.pow(pred - Y, 2)) / (2 * test_X.shape[0]),
    feed_dict={X: test_X, Y: test_Y}) # same function as cost above
print("Testing cost=", testing_cost)
print("Absolute mean square loss difference:", abs(
    training_cost - testing_cost))

```

[下载源文件](#)   [在线浏览源代码](#)   更新时间：2018-08-25 09:24:03   [详细评判结果 >>](#)

共有测试数据:5  
平均占用内存:9.621K   平均运行时间:0.06000S

测试数据	评判结果
测试数据1	输出错误
测试数据2	输出错误
测试数据3	输出错误
测试数据4	输出错误
测试数据5	输出错误

trainable=True by default  
h) '\n')

- 提供区块链类课程的实验体系

区块链与数字经济

区块链案例分析与设计

区块链原理与系统

矿机技术与开发实践

共识机制与博弈论

Solidity与智能合约

密码学基础

信息安全与数字认证

Go语言程序设计

分布式系统与P2P网络

**区块链专业**

## • 基于Jupyter提供智能金融实验体系

test.educg.net/admin/courseAdmin/expAdmin/guides/expPreviewJupyter.jsp?guidelD=3019

1.2 SDE模拟与统计 此处仅预览实验，对实验指导书和环境的改动不会保存！ 定制Jupyter实验，请进 [云桌面工作台->Jupyter工作台](#)。

提交实验笔记

jupyter d\_statistics 最后检查: 2020年7月16日 (未保存改变)

File Edit View Insert Cell Kernel Widgets Help 不可信 Python 3

运行 标记

### SDE 模拟和统计

#### Contents

- [布朗运动](#)
  - [置信区间](#)
  - [假设检验](#)
- [几何布朗运动](#)
- [CIR 过程](#)
  - [Euler-Maruyama 算法](#)
  - [参数估计](#)
  - [变量变化](#)

```
In [2]: import numpy as np
import scipy as scp
import scipy.stats as ss
import matplotlib.pyplot as plt
from statsmodels.graphics.gofplots import qqplot
from scipy.special import iv
from scipy.optimize import minimize
```

### 布朗运动

让我们模拟一些布朗路径。请记住(标准)布朗运动  $\{X_t\}_{t \geq 0}$  是一个连续时间的随机过程，它满足以下特性:

## • 基于Jupyter提供智能金融实验体系

test.educg.net/admin/courseAdmin/expAdmin/guides/expPreviewJupyter.jsp?guideID=1529

06\_Financial\_Time\_Series 此处仅预览实验, 对实验指导书和环境的改动不会保存! 定制Jupyter实验, 请进 [云桌面工作台](#)->[Jupyter工作台](#). 提交实验

jupyter 06\_Financial\_Time\_Series 最后检查: 2020年4月19日 (未保存改变) Python 3

File Edit View Insert Cell Kernel Widgets Help 不可信 Python 3

运行

### Financial Data

```
In [48]: # data from Thomson Reuters Eikon API
raw = pd.read_csv('source/tr_eikon_eod_data.csv',
                 index_col=0, parse_dates=True)
raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1972 entries, 2010-01-04 to 2017-10-31
Data columns (total 12 columns):
AAPL.0    1972 non-null float64
MSFT.0    1972 non-null float64
INTC.0    1972 non-null float64
AMZN.0    1972 non-null float64
GS.N      1972 non-null float64
SPY       1972 non-null float64
.SPX      1972 non-null float64
.VIX      1972 non-null float64
EUR=      1972 non-null float64
XAU=      1972 non-null float64
GDY       1972 non-null float64
GLD       1972 non-null float64
dtypes: float64(12)
memory usage: 200.3 KB
```

```
In [49]: data = pd.DataFrame(raw['.SPX'])
data.columns = ['Close']
```

```
In [50]: data.tail()
```

```
Out[50]:
```

	Close
Date	
2017-10-25	2557.15



08

---

**信息安全专业**

应用领域	实训案例	培养目标
信息安全防护	信息安全防护体系搭建	<ol style="list-style-type: none"><li>1. 了解防护体系基础</li><li>2. 了解防护体系的组成</li></ol>
安全攻击	攻击技术实训	<ol style="list-style-type: none"><li>1. 攻击原理的掌握。</li><li>2. 攻击工具的掌握</li></ol>
加密算法	了解各种加密算法	<ol style="list-style-type: none"><li>1. 掌握加密算法数学原理</li><li>2. 加密算法实现</li><li>3. 加密算法的应用</li></ol>
安全评级	信息安全等级案例	<ol style="list-style-type: none"><li>1. 掌握信息安全评级标准。</li><li>2. 掌握信息安全等级系统</li></ol>

## • 实验资源体系

- 信息安全前沿案例
- 北航+国防科大
- 资源动态更新
- 适合教学
- 自动化评分



操作系统基础



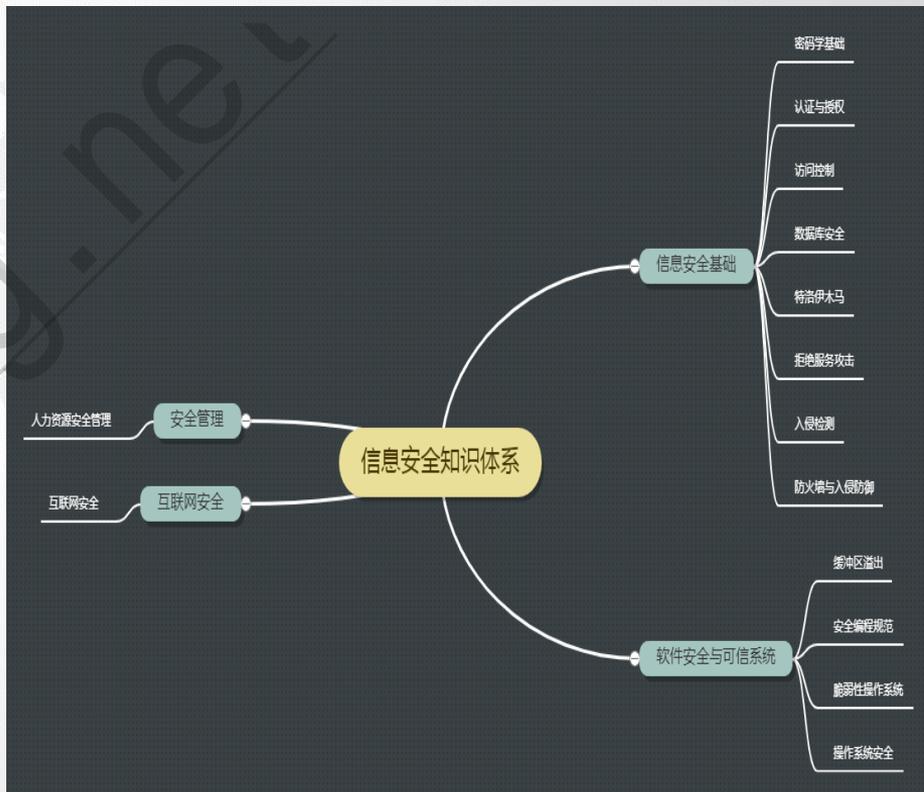
编译及调试技术



网络基础

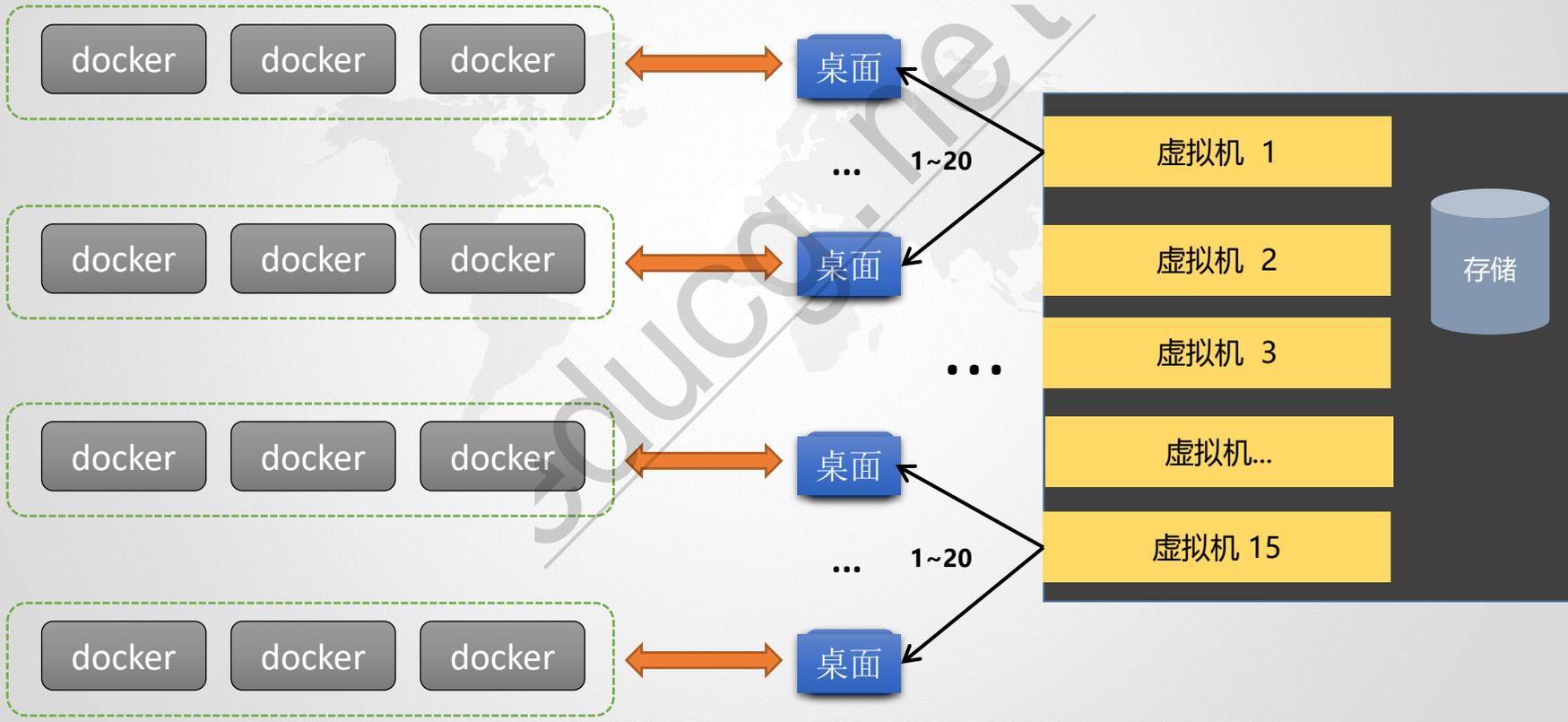


数学基础





- 高并发：单台服务器**300**并发量。
- 多级隔离：支撑学生开展破坏性实验
- 实验自动化评测：减轻教师负担，给学生及时反馈
- 流畅实验体验：**1秒**内打开桌面环境，支持全屏及分辨率自适应

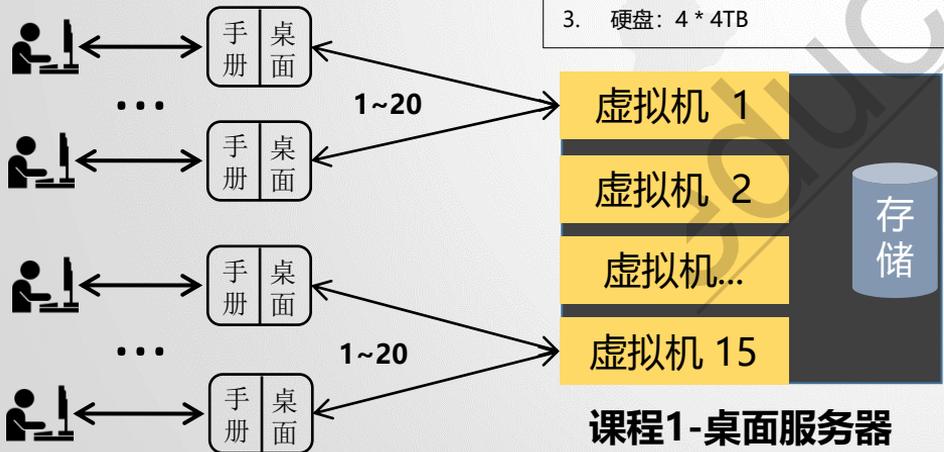


## • 硬件配置测算

希冀在线实验环境

一门课 (每届300人)

1台服务器, 15个虚拟机, 20用户/虚拟机



其它在线实验

[http://www.cstor.cn/proTextdetail\\_12031.html](http://www.cstor.cn/proTextdetail_12031.html)

服务器配置参数

产品型号	详细配置	单位	数量
CPU	E5-2650V4	颗	2
内存	32G DDR4 RECC	根	8
SSD	480G SSD	块	1
硬盘	4TB SATA	块	4
GPU	1080P(型号可选)	块	8

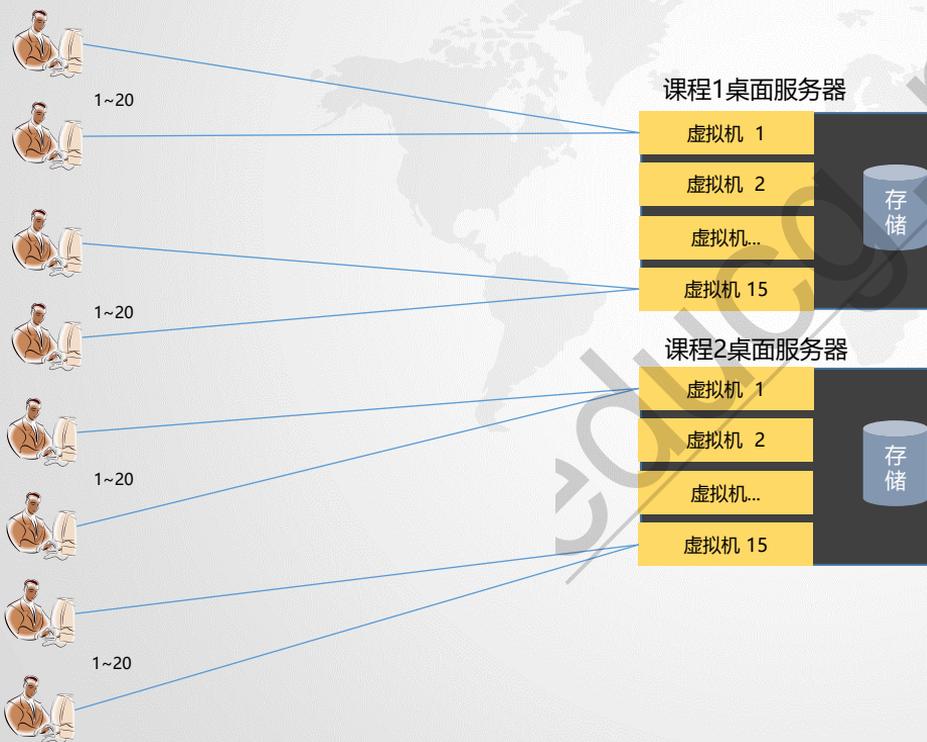
集群配置参数

某厂商实验资源配置

	极简型	经济型	标准型	增强型
上机人数	8人	24人	48人	72人
服务器	1台	3台	6台	9台
交换机	无	S5720-30C-SI	S5720-30C-SI	S5720-30C-SI
CPU	2*E5-2650V4	6*E5-2650V4	12*E5-2650V4	18*E5-2650V4
GPU	8*1080P(型号可选)	24*1080P(型号可选)	48*1080P(型号可选)	72*1080P(型号可选)
内存	8*32G DDR4 RECC	24*32G DDR4 RECC	48*32G DDR4 RECC	72*32G DDR4 RECC
SSD	1*480G SSD	3*480G SSD	6*480G SSD	9*480G SSD
硬盘	4*4TB SATA	12*4TB SATA	24*4TB SATA	36*4TB SATA

## 同时开10门实验课（每届300人）应该部署多少台虚拟机？以漏洞挖掘课程为例

10台服务器：150个虚拟机，20用户/虚拟机，每人存储资源一般为10GB（虚拟机实例）



## 一门课只需要一台服务器

### 说明

1. 一门课一般只有一个镜像，每位学生使用一个实例完成所有的实验。
2. 一门课对应一个学期，最大并发人数为学院每届招生人数。
3. CG实验环境支持一台虚拟机多个用户，相比独占虚拟机，对资源的需求可以呈几十倍的减少。

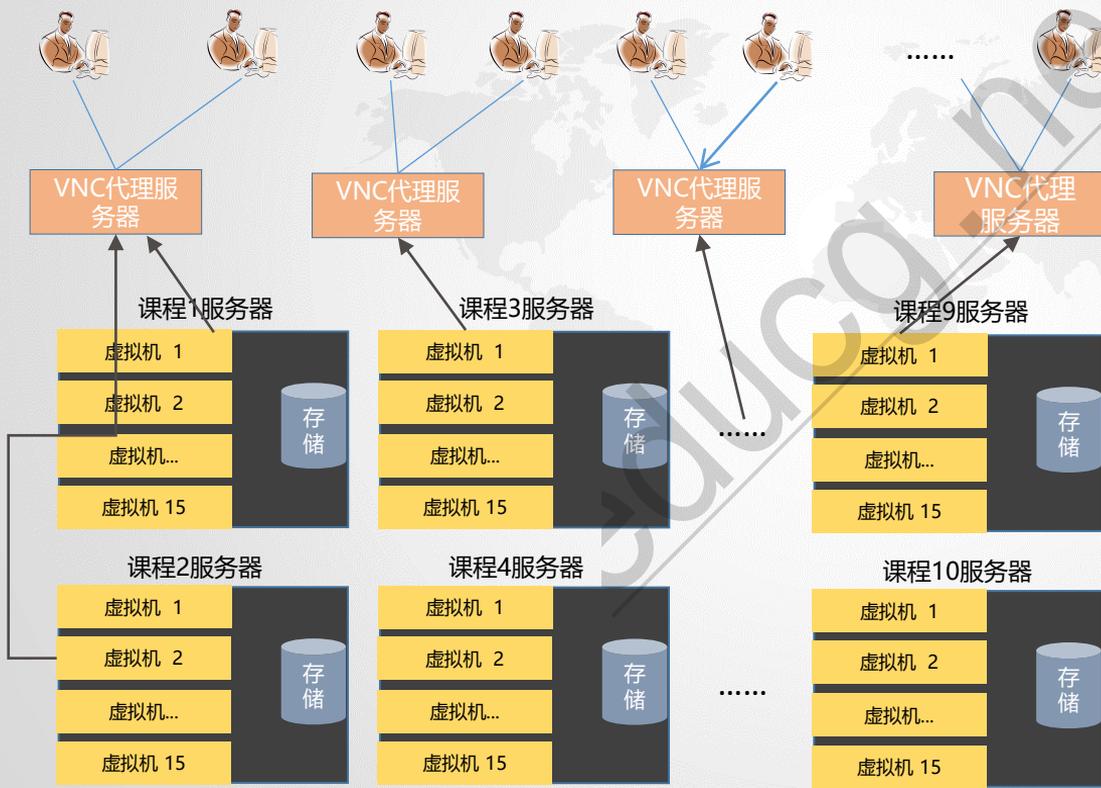
### 服务器推荐配置

1. CPU: 2 \* xeon E7, 15核
2. 内存: 512 GB.
3. 硬盘: 4 \* 4TB

## 若有3000人同时在线实验，如何部署？

保证桌面的流畅性，每用户1Mb的网络带宽。总带宽需求为3Gb，假设服务器为千兆网卡。

10台课程服务器，提供虚拟机环境。4台VNC代理服务器，总的理论带宽为4Gb。



## 一门课只需要一台服务器

### 说明

1. 一门课一般只有一个镜像，每位学生使用一个实例完成所有的实验。
2. 一门课对应一个学期，最大并发人数为学院每届招生人数。
3. CG实验环境支持一台虚拟机多个用户，相比独占虚拟机，对资源的需求可以呈几十倍的减少。

### 课程服务器推荐配置

1. CPU: 2 \* xeon E7, 15核
2. 内存: 256GB。
3. 硬盘: 4 \* 4TB
4. 网卡: 1Gb

### VNC服务器推荐配置

1. CPU: 1 \* xeon E5, 8核
2. 内存: 32GB
3. 硬盘: 300GB
4. 网卡: 1Gb

## 若有3000人同时在线实验，如何部署？

保证桌面的流畅性，每用户1Mb的网络带宽。总带宽需求为3Gb，假设服务器为千兆网卡。

10台课程服务器，提供虚拟机环境。4台VNC代理服务器，总的理论带宽为4Gb。



demo.cjudge.net:8081/admin/index.jsp

系统管理

系统公告 教师账号与权限 课程设置 系统设置 答疑论坛 实验环境管理

在线实验环境

- 镜像管理
- VNC代理
- 实例管理

SQL评测环境

- 评测数据库

串行程序评测环境

- C/C++编译选项
- Python运行环境
- C#编译环境

并行程序评判环境

- 多线程评判
- MPI评判

一般情况下不用配置VNC代理！CG Web服务器已经自动部署了VNC服务。

有两种情况需要设置VNC代理：

- 实验的并发访问量超过1000人。一个用户一般需要1Mb的网络带宽，才能保证实验桌面的流畅性，超过1000人，需要部署若干台VNC专用服务器，分流网络带宽。
- 实例在远程的内网内，只能通过一台有公网IP的机器访问，可以将配有公网IP的机器作为VNC代理服务器。

VNC代理服务器要求：操作系统版本仅限于Centos6、Centos7、Ubuntu14、Ubuntu16、Ubuntu18。

VNC服务器基本配置 VNC代理详细指南

主机ID	IP地址或域名	root密码	操作系统	备注	状态	部署	删除
CGVNCProxyServer	演示版隐藏IP	演示版隐藏密码	Ubuntu14		运行中, 查看日志	打开部署窗口	

+ 添加VNC代理服务器

## 一门课只需要一台服务器

### 说明

1. 一门课一般只有一个镜像，每位学生使用一个实例完成所有的实验。
2. 一门课对应一个学期，最大并发人数为学院每届招生人数。
3. CG实验环境支持一台虚拟机多个用户，相比独占虚拟机，对资源的需求可以呈几十倍的减少。

### 课程服务器推荐配置

1. CPU: 2 \* xeon E7, 15核
2. 内存: 256GB。
3. 硬盘: 4 \* 4TB
4. 网卡: 1Gb

### VNC服务器推荐配置

1. CPU: 1 \* xeon E5, 8核
2. 内存: 32GB
3. 硬盘: 300GB
4. 网卡: 1Gb

五

---

通用科研平台

## 项目管理

Jupyter Notebook

计费管理

计算资源分配

版本管理

## 数据集管理



社交网络



语音视频



计算机视觉



自然语言



智能医疗



智能金融

## 运行环境管理



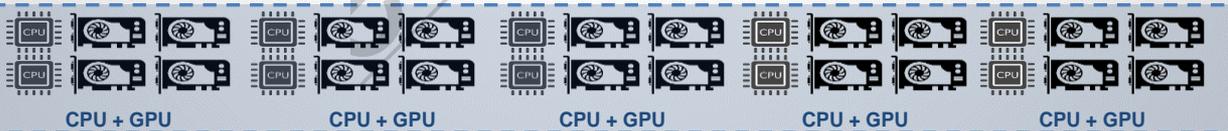
docker

CG-PAI: 人工智能科研集群管理与调度平台



kubernetes

## 集群配置管理 以及存储管理



GPU + k8s + Docker + DL Tools+Jupyter

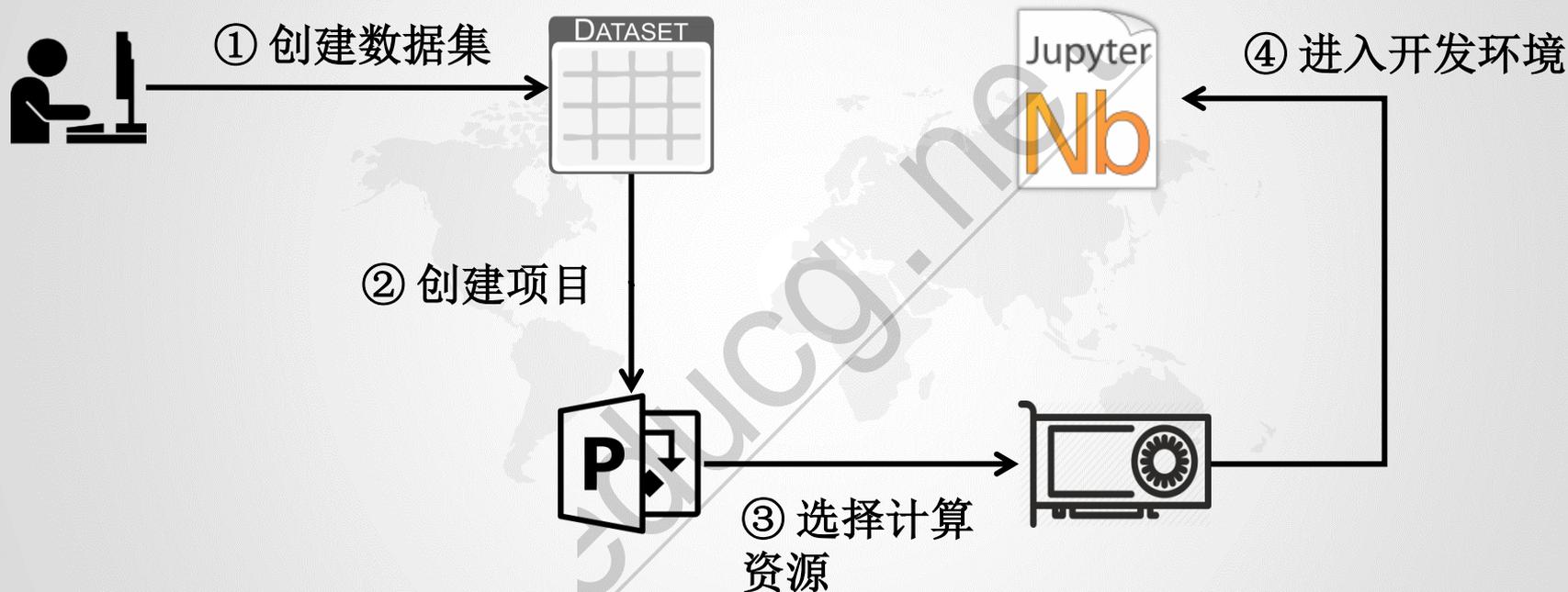


01

---

## 科研项目管理

## • 科研平台使用流程



1分钟内学会如何使用。

1分钟内开启一个科研项目。

## 创建数据集

名称 数据集 jupyter Untitled Last Checkpoint: 2018年10月10日 (unsaved changes)

### 创建项目

项目名称

项目描述

默认环境

挂

数据集

数

笔记文件

数

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

Markdown

数据向隔视为分析观测窗口，在窗口的所有数据即源数据。其中包含了会员卡号、入会时间、性别、年龄、会员卡级别、工作地城市、工作地所在省份、工作地所在国家、观测窗口结束时间、观测窗口乘机积分、飞行公里数、飞行次数、飞行时间、成绩时间间隔和平均折扣率等44个属性。

### 数据分析

对于原始数据，我们总要要进行探索分析，即对数据进行缺失值分析和异常分析，分析出数据的规律以及异常值。在我们的数据中，通过观察发现部分票价属性为空，或者有票价最小值为0、折扣率最小值为0、总飞行公里数却大于0的记录。票价为空值的数据可能是客户不存在乘机记录造成的，而其他的数据是客户乘坐0折机票或者积分兑换产生的。如何处理这些数据，取决于其在总数据中占的比重有多大。首先来查找每列属性观测值中空值个数、最大值、最小值等。新建文件 data\_explore.py,添加代码如下：

```
In [1]:  -*- coding: utf-8 -*-
#对数据进行基本的探索
#返回缺失值个数以及最大最小值

import pandas as pd

datafile = '../data/air_data.csv' #航空原始数据,第一行为属性标签
resultfile = '../tmp/explore.xls' #数据探索结果表

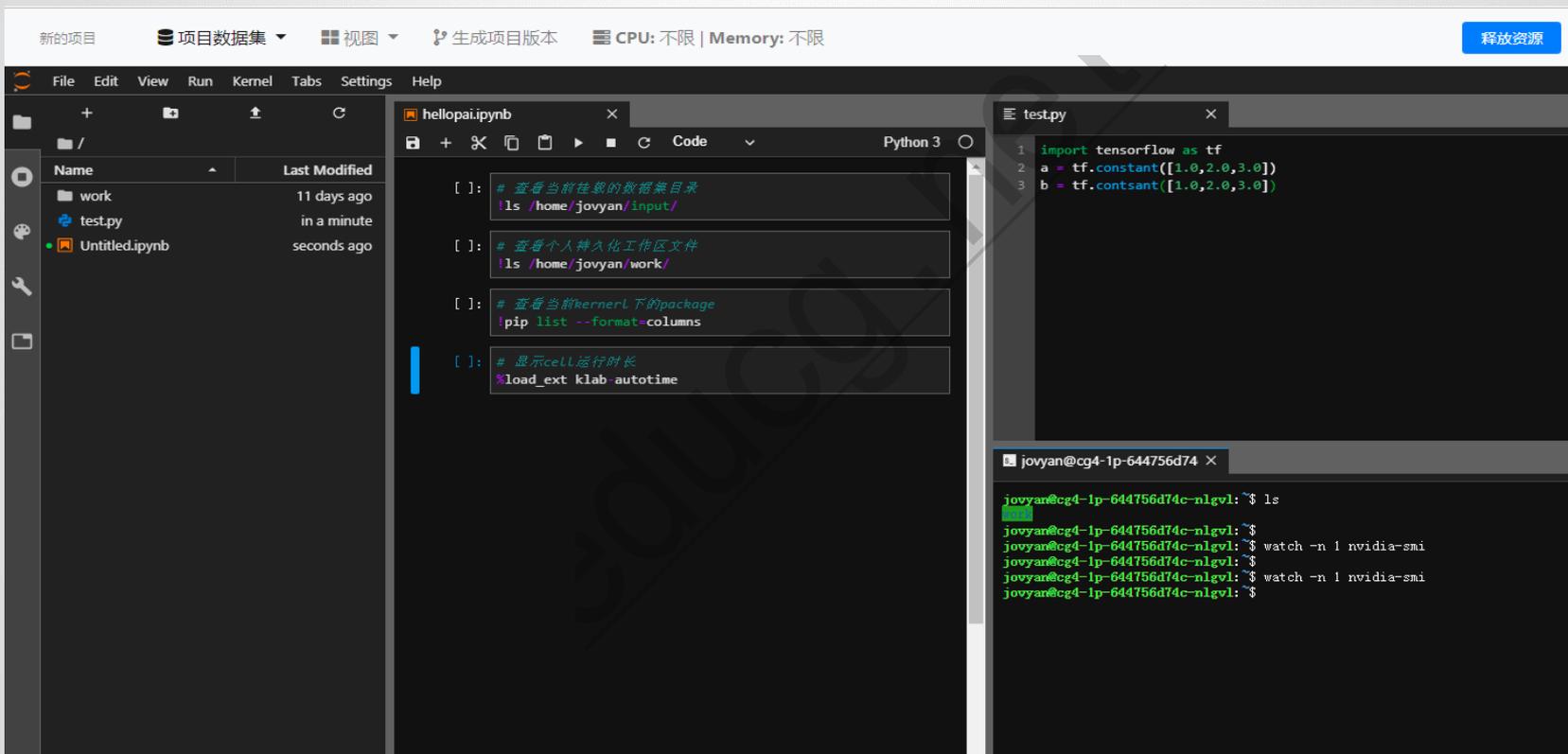
data = pd.read_csv(datafile, encoding = 'utf-8') #读取原始数据,指定UTF-8编码(需要用文本编辑器将数据转换为UTF-8编码)

explore = data.describe(percentiles = [], include = 'all').T #包括对数据的基本描述,percentiles参数是指定计算多少的分位数表(如1/4分位数、中位数)
explore['null'] = len(data)-explore['count'] #describe()函数自动计算非空值数,需要手动计算空值数

explore = explore[['null', 'max', 'min']]
explore.columns = ['空值数', '最大值', '最小值'] #表头重命名
'''这里只选取部分探索结果。
describe()函数自动计算的字段有count(非空值数)、unique(唯一值数)、top(频数最高者)、freq(最高频数)、mean(平均值)、std(方差)、min(最小值)'''

explore.to_excel(resultfile) #导出结果
```

## • Jupyter开发环境



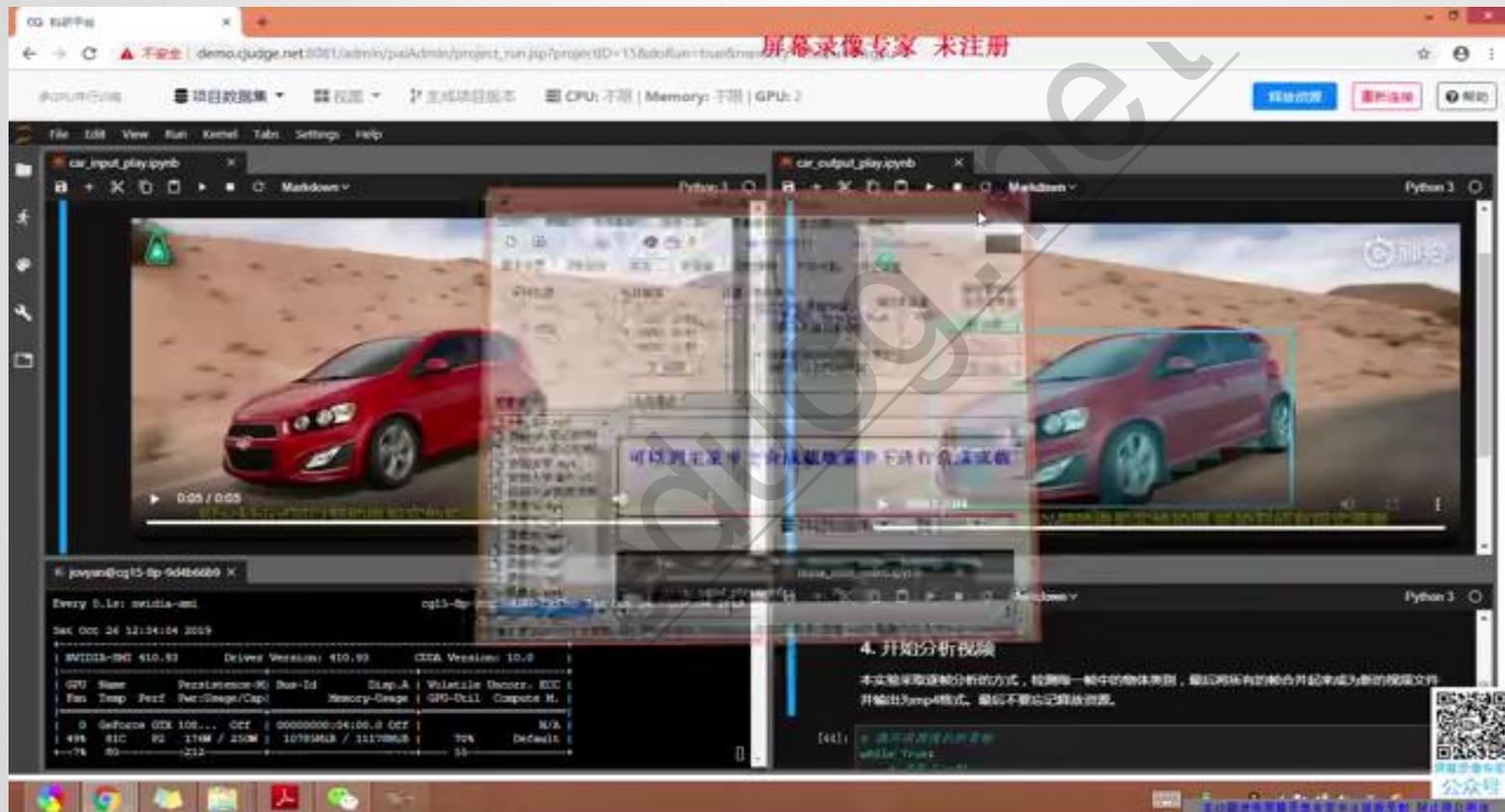
The screenshot displays the JupyterLab interface. At the top, there is a navigation bar with options like '新的项目', '项目数据集', '视图', '生成项目版本', and resource usage 'CPU: 不限 | Memory: 不限'. A '释放资源' button is on the right. Below this is a menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. The main workspace is divided into three panes:

- Left Pane (File Browser):** Shows a file tree with 'work' (modified 11 days ago), 'test.py' (modified in a minute), and 'Untitled.ipynb' (modified seconds ago).
- Center Pane (Notebook):** Displays a notebook named 'helloworld.ipynb' with four code cells:
  - Cell 1: `# 查看当前挂载的数据集目录`  
`!ls /home/jovyan/input/`
  - Cell 2: `# 查看个人持久化工作区文件`  
`!ls /home/jovyan/work/`
  - Cell 3: `# 查看当前kernel下的package`  
`!pip list --format=columns`
  - Cell 4: `# 显示cell运行时长`  
`%load_ext klab-autotime`
- Right Pane (Code Editor):** Shows a Python file named 'test.py' with the following code:

```
1 import tensorflow as tf
2 a = tf.constant([1.0,2.0,3.0])
3 b = tf.contsant([1.0,2.0,3.0])
```
- Bottom Pane (Terminal):** Shows a terminal window with the following output:

```
jovyan@cg4-1p-644756d74 ~$ ls
jovyan@cg4-1p-644756d74c-nlglv:~$
jovyan@cg4-1p-644756d74c-nlglv:~$ watch -n 1 nvidia-smi
jovyan@cg4-1p-644756d74c-nlglv:~$
jovyan@cg4-1p-644756d74c-nlglv:~$ watch -n 1 nvidia-smi
jovyan@cg4-1p-644756d74c-nlglv:~$
```

- Jupyter开发环境



demo.cugrading.net/3361/admin/pasAdmin/project\_run.jsp?projectID=15&doRun=true&mp=

屏幕录像专家 未注册

项目数据集 权限 生成项目版本 CPU: 不限 | Memory: 不限 | GPU: 2

File Edit View Run Kernel Tabs Settings Help

car\_input\_play.ipynb Python 3

car\_output\_play.ipynb Python 3

可以调至全屏，合成视频素材下进行录像或截

4. 开始分析视频

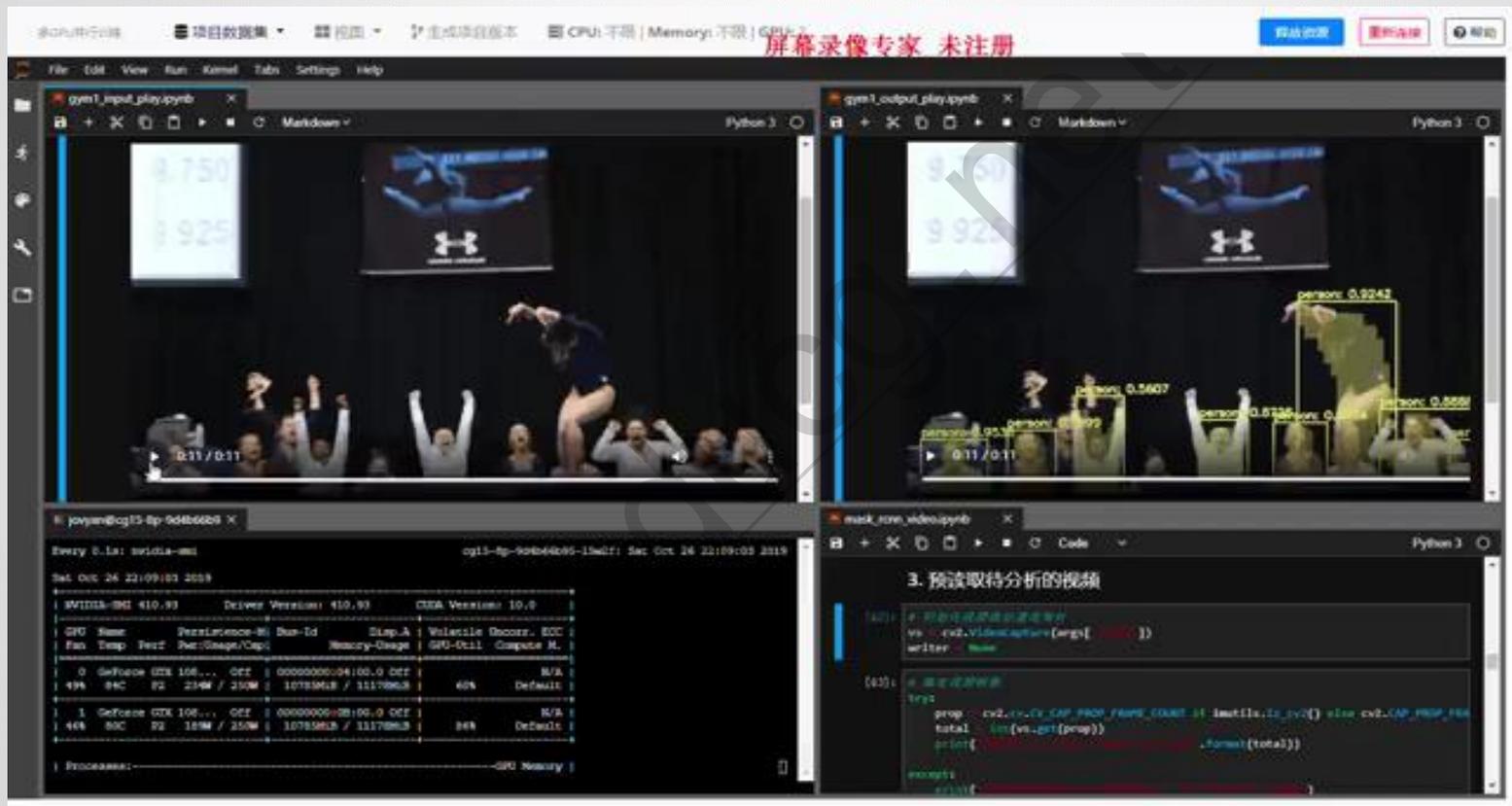
本实验采取逐帧分析的方式，检测每一帧中的物体类别，最后将所有帧合并起来成为新的视频文件并输出为mp4格式。最后不要忘记记录数据来源。

```
[44]: # 遍历所有帧并检测物体类别
while True:
    # 读取帧并检测物体类别
```

NVIDIA-SMI 410.83		Driver Version: 410.83		CUDA Version: 10.0			
GPU Name	Ferris(GeForce-M)	Bus-Id	Slap-A	Wolfslake	Shoortz, EOC		
Fan Temp	Perf	Perf	Perf	Usage/Cap	Memory-Usage	GPU-Driver	Compute M.
0	GeForce GTX 108...	0cf	00000000:04:00:0	0cf			N/A
49%	81C	W	1788 / 250W	1079MHz	3 / 11770MB	70%	Default
---	78	80	---	---	---	---	---

公众号

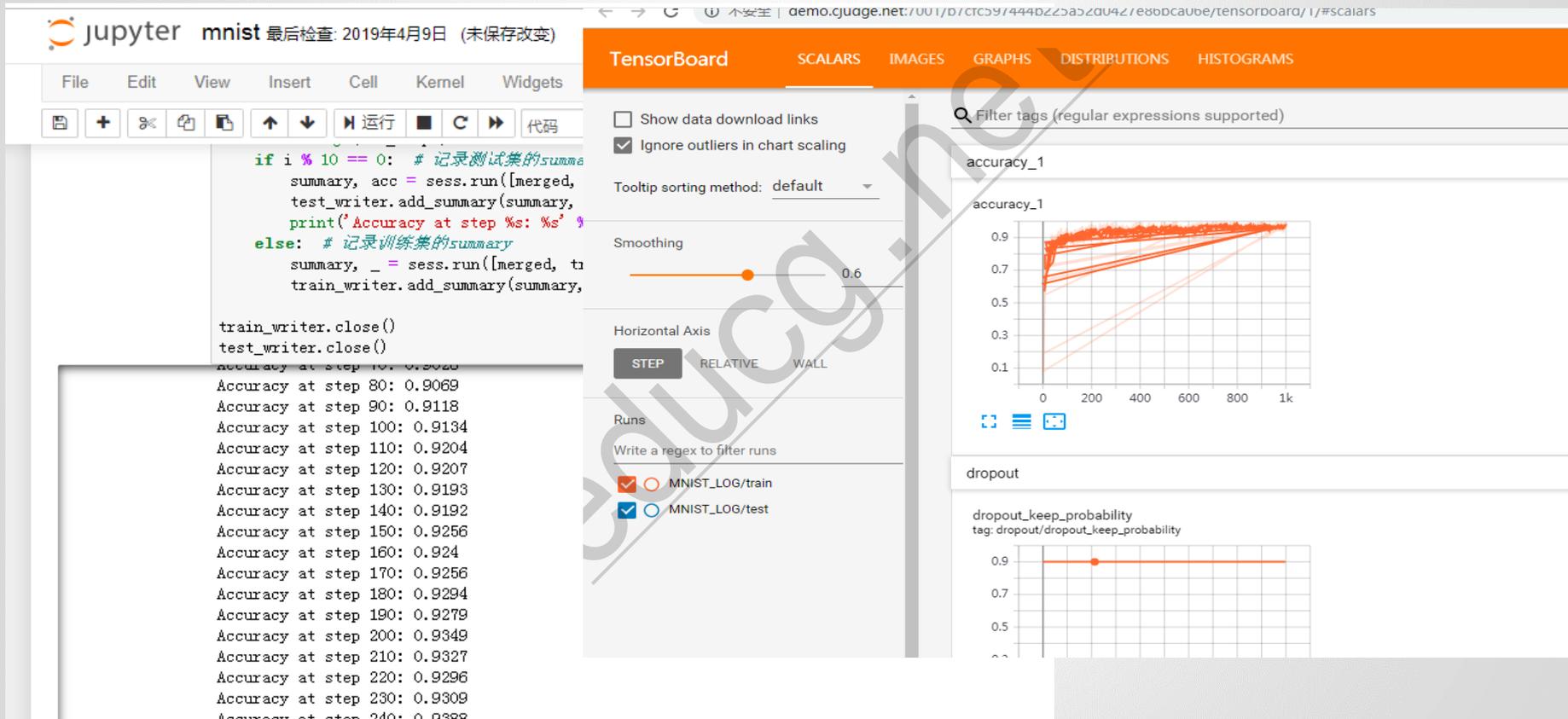
- Jupyter开发环境



The screenshot displays a JupyterLab environment with four main components:

- Top Bar:** Includes navigation icons, a project name '项目数据集', and system status 'CPU: 不眠 | Memory: 不眠 | GPU: 不眠'. A red watermark '屏幕录像专家 未注册' is visible.
- Left Panel:** A terminal window showing system information for 'joyan@cg15-tp-9046608'. It includes the command 'cat /etc/os-release' and the output of 'nvidia-smi', which lists GPU details for two GeForce GTX 1080 Ti cards.
- Top Row:** Two video player windows. The left window, titled 'gym1\_input\_play.ipynb', shows a video of a gymnast performing a vault with a score of 8.750 and a difficulty of 9.925. The right window, titled 'gym1\_output\_play.ipynb', shows the same video with yellow bounding boxes around the gymnast and labels such as 'person: 0.9242'.
- Bottom Row:** A code editor window titled 'mask\_store\_videos.ipynb' containing Python code for video analysis. The code includes comments in Chinese and uses OpenCV and NumPy to process video frames.

## • Jupyter开发环境



The screenshot displays a Jupyter Notebook environment on the left and the TensorBoard interface on the right. The Jupyter Notebook shows Python code for training an MNIST model and its output, which includes accuracy values at various steps.

```
if i % 10 == 0: # 记录测试集的summary
    summary, acc = sess.run([merged,
                             test_writer.add_summary(summary,
                                                         print('Accuracy at step %s: %s' %
else: # 记录训练集的summary
    summary, _ = sess.run([merged, t
                           train_writer.add_summary(summary,

train_writer.close()
test_writer.close()
Accuracy at step 80: 0.9069
Accuracy at step 90: 0.9118
Accuracy at step 100: 0.9134
Accuracy at step 110: 0.9204
Accuracy at step 120: 0.9207
Accuracy at step 130: 0.9193
Accuracy at step 140: 0.9192
Accuracy at step 150: 0.9256
Accuracy at step 160: 0.924
Accuracy at step 170: 0.9256
Accuracy at step 180: 0.9294
Accuracy at step 190: 0.9279
Accuracy at step 200: 0.9349
Accuracy at step 210: 0.9327
Accuracy at step 220: 0.9296
Accuracy at step 230: 0.9309
Accuracy at step 240: 0.9388
```

The TensorBoard interface shows the following configuration and data:

- TensorBoard navigation: SCALARS, IMAGES, GRAPHS, DISTRIBUTIONS, HISTOGRAMS
- Settings:  Show data download links,  Ignore outliers in chart scaling, Tooltip sorting method: default
- Smoothing: 0.6
- Horizontal Axis: STEP, RELATIVE, WALL
- Runs:  MNIST\_LOG/train,  MNIST\_LOG/test
- Filter tags: Filter tags (regular expressions supported)
- Chart 1: accuracy\_1 (Line graph showing accuracy over 1000 steps, with values ranging from approximately 0.6 to 0.95)
- Chart 2: dropout (Line graph showing dropout\_keep\_probability over 1000 steps, with a value of 0.9)

- 云桌面开发环境



- 1000+种数据集。
- 数据集持续更新。

项目 数据集

数据集 +新建 全部数据集

6000条周杰伦微博超话数据！

剪贴板 字体 对齐方式 数字 样式 单元

A	B	C
金融	无	深证主板日线数据, 截止 2017.05.05, 原始价、前复权价、后复权价, 466支股票
		深证中小板日线数据, 截止 2017.05.05, 原始价、前复权价、后复权价, 852支股票
		深证创业板日线数据, 截止 2017.05.05, 原始价、前复权价、后复权价, 636支股票
		上证A股日线数据, 1999.12.09至 2016.06.08, 前复权, 1095支股票
		深证A股日线数据, 1999.12.09至 2016.06.08, 前复权, 1766支股票
		深证创业板日线数据, 1999.12.09 至2016.06.08, 前复权, 510支股票
		MT4平台外汇交易历史数据
		Forex平台外汇交易历史数据
		几组外汇交易逐笔 (Ticks) 数据
		美国股票新闻数据【Kaggle数据】
		美国医疗保险市场数据【Kaggle数据】
		美国金融客户投诉数据【Kaggle数据】
		Lending Club 网贷违约数据【Kaggle数据】
		信用卡欺诈数据【Kaggle数据】
		美国股票数据XBRL【Kaggle数据】
纽约股票交易所数据【Kaggle数据】		
交通	无	贷款违约预测竞赛数据【Kaggle竞赛】
		Zillow 网站房地产价值预测竞赛数据【Kaggle竞赛】
		Sberbank 俄罗斯房地产价值预测竞赛数据【Kaggle竞赛】
		Homesite 保险定价竞赛数据【Kaggle竞赛】
		Winton 股票回报率预测竞赛数据【Kaggle竞赛】
		房屋租赁信息查询次数预测竞赛【Kaggle竞赛】
		2013年纽约出租车行驶数据
		2013年芝加哥出租车行驶数据
		Udacity自动驾驶数据
		纽约Uber 接客数据【Kaggle数据】
		英国车祸数据(2005-2015)【Kaggle数据】
		芝加哥汽车超速数据【Kaggle数据】
		德国交通标志识别数据
		交通信号识别数据
		芝加哥Divvy共享自行车骑行数据(2013年至今)
美国查塔努加市共享单车骑行数据		
Capital 共享单车骑行数据		
Bay Area 共享单车骑行数据		

## 自己生产真实数据

课程公告
课程信息
学生
作业

**作业题库**

- 题目概览
- 录入题目
- 编程题
- 回收站

**作业管理**

- 作业概览
- 布置作业
- 回收站

**数据统计**

- 作业活动数据
- 答题明细数据
- 常见问题

批量导出至CSV(UTF-8编码)

#	学号	姓名	题目ID
1	518021910001	冯尊	64664
2	518021910001	冯尊	64663
3	518021910001	冯尊	69246
4	518021910001	冯尊	69247
5	518021910001	冯尊	68865

### CG-Dashboard

F11切换全屏 | 服务器时间: 2019-10-15 15:55:32 | CourseGrading 4.3.1 旗舰版

**396**

在线人数

**30**

在线课程

**313326**

评测次数

**6928937**

总代码行

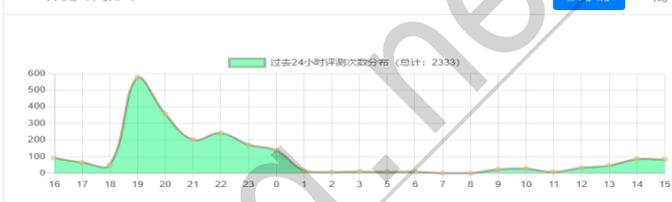
**2832 / 14700**

今年注册人数 / 累计人数

**CPU: 6.3%**

内存 42.6% | 网络收发 0/0(KB/s)

#### 评测时间分布



**正在进行的考试**

编译技术 | 期中考试熟悉环境 | 0人

4/145 (分钟)

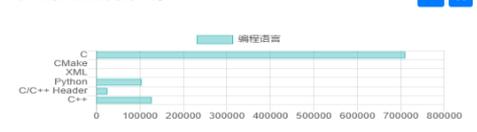
**正在进行的作业**

数据结构 (高工)   2019秋季第7周练习   78人	20/167 (小时)
数据结构 (高工)   历史作业练习   78人	1071/8784 (小时)
高级语言程序设计 (高工)   19级第三次作业   1人	165/509 (小时)
高级语言程序设计 (高工)   19级第二次作业   1人	501/509 (小时)
编译技术   测试错误处理   4人	301/687 (小时)
编译技术   2019年语法分析作业   0人	463/520 (小时)
编译技术   2019年词法分析作业   0人	509/518 (小时)
编译技术   词法分析和语法分析测试   9人	972/2684 (小时)
编译技术   词法分析_测试分数算法   4人	664/1253 (小时)
软件测试技术 (研)   2015级 (研) 软件测试大作业   18人	30135/108000 (小时)

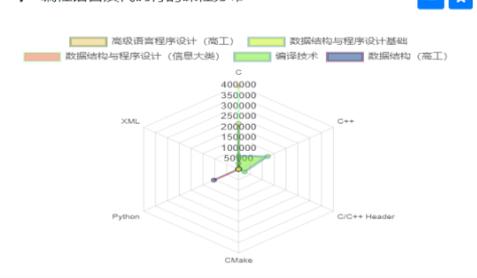
#### 最近一个月的活跃课程



#### 编程语言及代码行



#### 编程语言及代码行的课程分布





02

---

**计算资源管理**

## • 计算资源计费管理

#	用户	项目名	
1	教师   teacher	多GPU并行训练	<a href="#">明细 &gt;</a>
2	教师   teacher	车牌号码识别-GPU	<a href="#">明细 &gt;</a>
3	教师   teacher	GAN手写数字生成	<a href="#">明细 &gt;</a>
4	教师   teacher	手写数字识别-TensorBoard可视化	<a href="#">明细 &gt;</a>
5	教师   teacher	CGPAI-常用命令	<a href="#">明细 &gt;</a>
6	教师   teacher	手写数字识别	<a href="#">明细 &gt;</a>
7	admin   admin	迁移学习提供「借贷风险评估」解决方案   Baseline	

项目计费明细

教师 | teacher : 多GPU并行训练

#	时间(天)	总用时(分钟)	CPU时间(CPU核数 * 分钟)	内存时间(内存GB * 分钟)	GPU时间(GPU卡 * 分钟)
1	2019/07/26	314	20	83	628
2	2019/07/26	1	0	0	2
3	2019/07/27	210	14	56	420

Close

## • 计算资源配额管理

如果增加或者更换集群节点，务必注意进入 [运行环境](#)，点击按钮“重新分发镜像”，在新节点上部署运行环境。

主节点名

IP地址或者主机名，如果用主机名注意在/etc/hosts内配置域名解析。

主节点Root密码   显示密码

从节点列表

```
cgpai-node1
cgpai-node2
cgpai-gpu1
```

推荐使用主机名，每个节点占一行，主机名不必配置域名解析，从节点的主机名通过主节点来解析。

如果主节点也参与计算，注意把主节点名加入该列表。

资源配额 节点数上限 7 内存上限 64 GB CPU上限 8 GPU上限 4

用户能够申请的最大资源，根据集群节点配置填写。节点数指的是，用户一次能够申请的集群服务器数量。别的选项是单个节点的最大资源量，其中CPU为核数或超线程。

确定修改

NAME	STATUS	ROLES	AGE	VERSION
cgpai-gpu1	Ready		114d	v1.13.5
cgpai-master	Ready	master	114d	v1.13.5
cgpai-node1	Ready		114d	v1.13.5
cgpai-node2	Ready		114d	v1.13.5

NAME	STATUS	AGE	READY	STATUS	RESTARTS	AGE
cg-pai	Active	38d				

NAME	READY	STATUS	RESTARTS	AGE
cg59-8p-deploy-656c4cc974-nhzm1	1/1	Running	0	26d

停止所有正在运行的作业/服务

查看集群详细信息

## • 计算资源扩展

### 科研平台集群配置

如果增加或者更换集群节点，务必注意进入 [运行环境](#)，点击按钮“重新分发镜像”，在新节点上部署运行环境。

主节点名

IP地址或者主机名，如果用主机名注意在/etc/hosts内配置域名解析。

主节点Root密码

从节点列表

只需添加计算节点名称  
即可完成计算资源扩展

推荐使用主机名，每个节点占一行，主机名不必配置域名解析，从节点的主机名通过主节点来解析。

如果主节点也参与计算，注意把主节点名加入该列表。

NAME	STATUS	ROLES	AGE	VERSION
cgpai-gpu1	Ready	61d	v1.13.5	
cgpai-master	Ready	master	61d	v1.13.5
cgpai-node1	Ready	61d	v1.13.5	
cgpai-node2	Ready	61d	v1.13.5	

NAME	STATUS	AGE
cg-pai	Active	4d2h

NAME	READY	STATUS	RESTART
cg16-1p-55876fc95c-ctdct	1/1	Running	0
cg17-1p-68d878698d-dkj9h	1/1	Running	0
cg4-1p-644756d74c-nlgvl	1/1	Running	0

停止所有正在运行的作业/服务

查

极简的计算资源扩展方式，无缝对接现有服务器

## • 存储资源管理与扩展

[+ 教师账号与权限](#)[课程设置](#)[? 答疑论坛](#)[系统设置](#)[实验环境管理](#)

### 科研平台存储管理

系统目前支持NFS、GlusterFS存储系统，科研数据与执行环境都存放在该路径下。系统至少有一个存储目录，如果有多个存储目录，自动在多个存储间进行容量均衡。

存储ID	存储路径
straoge2	/cgsvdir

录入新的存储

存储ID

不超过32个字符，只能包括子母和数字

存储ID是为了区分多个存储设备。

存储路径

/CGPAI

该存储路径一定要与集群节点的挂载路径相同。

文件系统类型

NFS

确定



03

---

**科研环境管理**

## 运行环境扩展

系统目前只支持Jupyter官方Docker镜像和以此为基础构建的Docker镜像。

+ 增添Jupyter Docker镜像

重新分发镜像

DockerID	Docker标签(tag)	镜像描述	镜像文件包	镜像分发状态
jupyter-datascience-notebook	jupyter/datascience-notebook:latest			
jupyter-tensorflow-notebook	jupyter/tensorflow-notebook:latest			
jupyter-r-notebook	jupyter/r-notebook:latest			
jupyter-gpu-datascience	jupyter/cggpu-datascience:v03			
jupyter-tensorboard-notebook	coursegrading/tensorboard-notebook:v03			

录入新的镜像

**Docker ID**   
Docker ID全局唯一的标识符。

**Docker 标签(tag)**   
Docker标签一定保持全局的唯一性，镜像包上传之后，将会以该标签重命名。标签一旦设定将不能被修改。

**Docker描述**   
介绍一下Docker镜像内部署的组件和版本等信息。

依赖GPU 【若勾选，集群必须有GPU才能使用该Docker镜像】

确定

## • 一键制作科研环境



The screenshot displays a remote desktop interface. On the left, a web browser window shows the 'Tianshou' page, which describes a reinforcement learning platform. A system menu is open over the browser, with a red arrow pointing to the '保存镜像' (Save Image) option. The desktop background features the '希冀 Course Grading' logo and the text 'Right here waiting for you' and '在这里等着你'. Desktop icons include Home, File System, Trash, Firefox Web Browser, Chromium Web Browser, and a folder named 'prolog'.

test.educg.net/admin/courseAdmin/expAdmin/guides/expDesktopWorkbench.jsp?hostname=deskdocker2&token=d32b9d0ce0f16964d8f87bc57d45cbd2&vnc\_host=test.educg.net&vnc\_port=&imageID=cg-v...

远程桌面工作台 工作笔记 桌面全屏 隐藏桌面 更多

工作笔记 转为实验相

重新连接桌面

剪切板

保存镜像

共享桌面

下载远程桌面内的文件

上传文件至远程桌面

评测脚本管理

重新启动

桌面还原

帮助

### Tianshou

Tianshou (天授) is a reinforcement learning platform based on PyTorch, supporting reinforcement learning libraries, which are mainly based on nested classes, unfriendly API, or slow-speed, Tianshou provides a unified framework and pythonic API for building the deep reinforcement learning environment. The supported interfaces include:

- Policy Gradient (PG)
- Deep Q-Network (DQN)
- Double DQN
- Dueling DQN
- Advantage Actor-Critic (A2C)
- Deep Deterministic Policy Gradient (DDPG)
- Proximal Policy Optimization (PPO)
- Twin Delayed DDPG (TD3)
- Soft Actor-Critic (SAC)
- Vanilla Imitation Learning
- Prioritized Experience Replay (PER)
- Generalized Advantage Estimator (GAE)

Here is Tianshou's other features:

- Elegant framework, using only ~2000 lines of code
- Support parallel environment simulation (synchronous or asynchronous) for all algorithms Usage
- Support recurrent state representation in actor network and critic network (RNN-style training for POMDP) Usage
- Support any type of environment state/action (e.g. a dict, a self-defined class, ...) Usage
- Support customized training process Usage
- Support n-step returns estimation and prioritized experience replay for all Q-learning algorithms Usage

ms; GAE, nstep and PER are very fast thanks to numba jit function

Applications 14:41 root

Home

File System

Trash

Firefox Web Browser

Chromium Web Browser

prolog

Right here waiting for you

在这里等着你

## • 一键添加科研环境

test.educg.net/admin/index.jsp

希冀 系统管理 课程管理 CG Online Judge 科研平台 工程教育认证 GitLab 帮助 云桌面实验平台 (2020)

公告与问卷 课程信息 学生 作业 考试 在线实验 在线课堂 答疑 汇总成绩 助教 设置 进入学生端

实验资源库  
实验概述  
添加实验  
回收站

实验作业管理  
实验作业概述  
布置实验作业  
回收站  
计算资源管理

云桌面工作台  
远程桌面工作台  
Jupyter工作台

数据统计  
在线实验活动记录  
在线实验明细数据  
常见问题

远程桌面工作台 / 在桌面工作台可以制作实验环境或者进行教学演示。 / 镜像加入公共镜像库之后, 才能被用作实验环境, 并注意提醒管理员部署该镜像。

打开远程桌面工作台

#	保存的镜像	镜像描述	桌面镜像	加入公共实验镜像库	加入科研平台镜像库	加入通用评测环境	删除	编辑	最近修改时间
1		From cg/vnc/logisim:v1.0.3   所在主机 deskdocker2   Docker启动选项 --privileged --user 0 --hostname=cg   By admin   By admin	cg/vnc/logisim:v1.0.4 1.30 GB <a href="#">查看日志</a>						admin 2020-08-26 20:58:40
2		From cg/ubuntu-1804:v1.0.2   所在主机 oslabanddeskdocker   Docker启动选项 --privileged --user 0 --hostname=cg	cg/vnc/ic:v1.0.1 38.32 GB <a href="#">查看日志</a>						admin 2020-08-25 10:55:48
3		From cg/logisim:v1.0.2   所在主机 deskdocker2   Docker启动选项 --privileged --user 0 --hostname=cg   By admin   By admin	cg/vnc/logisim:v1.0.3 1.30 GB <a href="#">查看日志</a>	已加入 <a href="#">查看日志</a>					admin 2020-08-10 23:02:25
4		From cg/kvm:v1.0.4   所在主机 deskdocker2   Docker启动选项 --privileged --user 0 --hostname=cg	cg/vnc/rhl9:v1.0.1 4.21 GB <a href="#">查看日志</a>	已加入 <a href="#">查看日志</a>					admin 2020-07-20 21:20:36

正在等待 test.educg.net 的响应



---

## 建设方案总结

## 所有专业课程一站式解决方案

基础课 + 专业核心课 + 综合实训项目

## 唯一具备完善的课程管理

支持所有的课程、极佳的用户体验、完备的功能

## 在线虚拟桌面实验环境

随时随地流畅实验，屏幕分辨率自适应。

## 符合教学实践过程

完善的实验过程管理机制。

## 轻负载设计架构

节省几倍到几十倍的服务器数量，降低维护成本。

## 工业生产级实验环境

可开展大型实验，真正体会大数据工具的魅力

## 完备实验体系

覆盖各专业各课程的知识体系。

## 软件硬件松耦合

独立升级换代，独立维护

## 开放可扩展

拓展课程，拓展题库，拓展实验资源，拓展MOOC资源

## 代码查重

督促学生独立完成代码编写

## 文档查重

督促学生独立完成实验报告。

## 自动评测机制

减轻教师工作量。

## 实验过程追踪和数据分析

可对学生和教学过程进行评价

## 支持自动在线运维

保持系统长期、可靠运行

## 最具技术深度的平台

每一门课程的支持技术都处于国内顶尖水平！

## 唯一的一体化支撑平台

良好体验和稳定性的前提下，有机融合各个专业功能

- 您的参与是希冀平台发展的动力!

**QQ群: 214085215**

**实名: 学校+姓名**



群名称:希冀CourseGrading高校用...  
群 号:214085215

➤ 希冀平台官方网站: [www.educg.net](http://www.educg.net)

---

➤ 在线演示系统地址: [demo.educg.net](http://demo.educg.net)

---

➤ 在线开课系统地址: [course.educg.net](http://course.educg.net)

---

A background image showing a business meeting with people in suits looking at documents. The entire image is overlaid with a semi-transparent red color. In the center, the Chinese characters '谢谢!' are written in white. Below them, the English text 'THANKS FOR YOUR ATTENTION' is written in white. A faint watermark '3d100.com' is visible diagonally across the center.

# 谢谢!

THANKS FOR YOUR ATTENTION