

清 华 大 学

综 合 论 文 训 练

题目：基于 FU740 的线上硬件交互实
验平台开发

系 别：计算机科学与技术系

专 业：计算机科学与技术

姓 名：唐皓泓

指导教师：向 勇 副研究员

2022 年 6 月 15 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名： 唐皓冰 导师签名： 何良 日 期： 2022/06/15

中文摘要

本工作旨在研究对 FU740 的控制、建立起一个可行的线上调试流程并实现线上实验平台的搭建。通过研究实现了对 FU740 的多种控制，完整的调试流程并设计出资源调度策略，同时也实现了用户交互调试界面。

对 FU740 的控制达到了以下五项：GPIO 电源控制、HDMI 的输出、TFTP 文件传输、OpenOCD 调试 U-Boot 以及 JTAG 串口读取；线上 Moodle 平台提供了交互界面并支持申请预约功能；服务器端实现预约优先分配算法以及机器状态更新算法。

本工作最终成功搭建了一个基于 FU740 的 Moodle 线上硬件交互实验平台。

关键词：FU740；Moodle；线上实验平台；硬件交互

ABSTRACT

It aims to find the new approach of controlling FU740 and build an executable debugging flow and reliable online experimental platform in this work. Though the research, it comes out several methods of controlling FU740 and completes the debugging flow and a resource scheduling strategy design.

The control of FU740 has reached the following options: GPIO power control, HDMI output, TFTP file transfer, OpenOCD U-Boot debugging and JTAG serial port debugging. The online Moodle platform provides an interactive interface and supports the function of applying for reservation. Server implement Reservation priority strategy algorithm and Machine state updating algorithm.

At the end, this work established a online experimental platform with hardware interaction based on FU740.

Keywords: FU740; Moodle; online experimental platform; hardware interaction

目 录

第 1 章 引言	1
1.1 研究背景	1
1.2 研究内容	1
1.2.1 HiFive Unmatched 开发板的启动方法与远程调试方式探究	1
1.2.2 线上实验交互平台的设计与搭建	1
1.3 主要贡献	2
1.4 本文结构	2
第 2 章 相关工作	3
2.1 已有工作	3
2.1.1 HiFive Unmatched 开发板	3
2.1.2 RaspberryPi 树莓派	4
2.2 复现工作	5
2.2.1 HiFive Unmatched 开发板	5
2.2.2 RaspberryPi 树莓派	6
第 3 章 线上平台设计	7
3.1 系统组成	7
3.1.1 系统模块化功能总结图	7
3.2 改进工作	8
3.2.1 更多的机器控制额	8
3.2.2 获取 IP 地址	9
3.2.3 交换机	9
3.3 线上平台需求分析	9
3.3.1 用户端需求分析	9
3.3.2 服务器端需求分析	11
3.4 模型逻辑构思	12
3.4.1 用户端模型	12
3.4.2 服务器端模型	13

3.4.3 构思流程	13
第 4 章 线上平台实现	16
4.1 线上平台的选择	16
4.2 服务器端	16
4.2.1 Docker 虚拟环境	16
4.2.2 用户文件保存	18
4.2.3 用户连线安全保障	18
4.2.4 用户行为分析记录	19
4.2.5 机器使用预约系统	20
4.2.6 排程服务器后台实现	25
4.2.7 Moodle 安装	25
4.3 用户端	25
4.3.1 响应网站	25
4.3.2 Docker 使用	26
4.3.3 调试方式选择	26
4.3.4 Moodle 交互界面实现	26
第 5 章 线上平台的应用	31
5.1 Hifive Unmatched 远程开发和 NXOS 移植-工作介绍	31
5.2 应用成果	31
第 6 章 结论	33
6.1 工作总结	33
6.2 工作展望	33
插图索引	34
表格索引	35
参考文献	36
致 谢	37
声 明	38
外文资料的书面翻译	39

第 1 章 引言

1.1 研究背景

考虑到现今计算系的原理课与实验环节几乎是密不可分的，就比如计算机组成原理课就有一个在线的硬件仿真实验平台让同学们能够在板子上跑自己的程序，加强同学们的学习成效以及动手能力，让同学们在做实验的过程中学习相关知识。

同样是四大原理课中的操作系统课，缺乏一个能够让同学们与实际板子打交道的机会，只能在软件模拟环境中做实验。同学们并不能在实际的板子中解决实际系统的问题。因此，我希望操作系统课也能够拥有一个在线硬件实验平台，让同学们能够从实际系统中学习相关知识，提升课程的多样性以及培养同学们发现问题以及解决实际系统问题的能力。

另外，在疫情的影响下，在线实验平台更显得有价值。能够让学生们安全地，不用到实验室都能够利用实际板子来进行实验学习。

1.2 研究内容

1.2.1 HiFive Unmatched 开发板的启动方法与远程调试方式探究

HiFive Unmatched 开发板，是一个配搭 SiFive Freedom U740 SoC 处理器且支持 RISC-V 开源指令集架构的开发板。我需要从开发板的硬件配置、启动方式开始研究，这对于我来说无疑是一个全新的挑战。在理解基本知识后再去寻找一个目前最合适且可靠的远程调试方式。

1.2.2 线上实验交互平台的设计与搭建

通过调研发现^[1-4]，目前线上平台的使用率处于上升趋势，同时也存在一定的挑战。在线实验方便使用和资源利用效率高，而真实板子实验能更好地体验真实系统中需要解决的问题。这两者的结合是在线的真实环境实验系统设计便是这次工作的目标。

所以，本项工作的目标，旨在建立一个能够让用户操作实际板子进行操作系统实验，平台管理员也能够有系统的进行资源的分配、收集用户信息的一个简易可靠且安全的平台。同时也希望该平台的实现，能够尽量简单高效，让复现的成本降到最低，适合应用在各式各样的平台上，减少重新搭建平台的开销。

1.3 主要贡献

在这项工作里，我的主要贡献有以下几个部分：复现前人工作，包括硬件环境的配置和软件环境的搭建；利用更高效以及合理的方法改进一些已实现的功能；然后是最主要的工作：在树莓派上搭建出一个可以成功与 FU740 进行交互使用的线上平台。

1.4 本文结构

本文会从五个部分展示工作内容以及成果。第二章介绍已有的相关工作。第三章介绍改进工作以及线上平台的设计分析；第四章详细讲解线上平台的实现；第五章是线上平台的应用例子；第六章最后一部分是本文的结论。

第 2 章 相关工作

2.1 已有工作

这项工作主要是在清华大学的曹隽诚同学所完成的工作^①：U740 的远程控制的基础上进行开发。首先学习了开发板的一些基本知识，比如开发板的启动流程、默认 Boot 方式，Console，Flashing 等。然后按照曹隽诚同学所完成的搭建方法进行复现，重新搭建一套配置。在配置的过程中再改进一些实现细节。

2.1.1 HiFive Unmatched 开发板

在开发板的配置上，主要是根据曹隽诚同学的方法来实现的。

1. 远程开机方法

首先是远程的开机方法。因为如果想要启动 U740 开发板的话，必须要手动按开机键才能够进行开机，这样便会使得网上调试变得不可行。因此曹隽诚同学想到了一个解决方法：利用一个中间媒介输出 GPIO 信号，连接继电器，然后利用 GPIO 输出高低电平的方法来模拟按开机键。具体流程见图 2.1

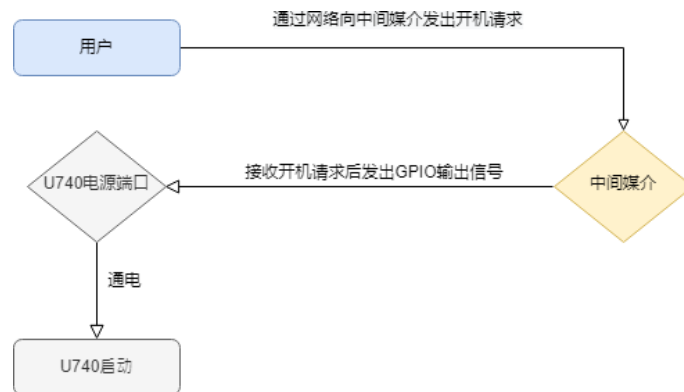


图 2.1 远程开机流程示意图

2. 启动流程

HiFive Unmatched 开发板的默认启动流程如图 2.2。

对于启动 (Boot)，有两种可选的方法：第一种是把 U-Boot 烧录到 SD 卡，在启动的时候通过 SD 卡启动，这种是官方默认的启动方法，但这对于需要使用自定义 U-Boot 的远程用户来说是不可能的，因为远程用户无法对 SD 卡进行读写操作。因此曹隽诚同学对此提出了一种方法，使用 RISC-V OpenOCD

^① U740 的远程控制：<https://nichic.co/articles/rpi-bmc.html>

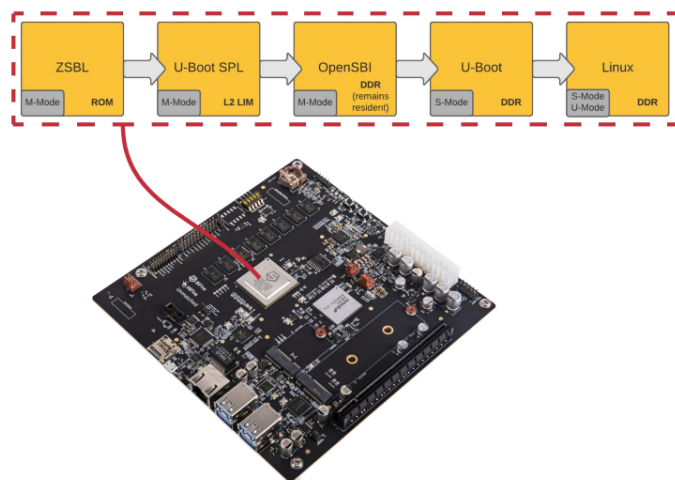


图 2.2 HiFive Unmatched 引导流程示意图

远程直接将 U-Boot 写入到 Flash，然后更改开发板的默认启动方式，改成从 Flash 启动，这样便可以让远程用户利用自定义的 U-Boot 启动开发板。

而对于需要调试内核的远程用户，开发板的标准开机流程是将内核写入到 SD 卡的分区中，在 U-Boot 启动后读取在 SD 卡分区中的内核并加载运行。但是考虑到用户需要进行远程开发，无法对 SD 卡进行读写操作，因此需要用到 U-Boot 的动态配置启动特性：选择从网络加载。在 U-Boot 启动时选择从服务器中下载内核镜像文件，然后加载运行。因此，需要配置一个服务器让用户能够上传自己的内核镜像文件到开发板。

3. 串口 (Console)

开发板上有内置的 UART/JTAG 到 USB 适配器，只需连接 USB 线即可，无需作特殊配置。

2.1.2 RaspberryPi 树莓派

在曹隽诚同学的搭建方法里，选用了 RaspberryPi(树莓派) 作为中间媒介，然后连接开发板，使远程用户可以通过访问树莓派来间接访问开发板。

1. 远程控制开发板启动方法

首先物理上连接开发板：利用杜邦线，中间连接一个继电器，连接树莓派上的 GPIO 端口与开发板的电源端口。之后可以通过程序控制树莓派的 GPIO 端口输出，从而达到控制开发板电源开关的效果。

对于远程用户，曹隽诚同学编写了一个简易的 API，利用 HTTP GET 请求来实现对远程开发板的开关。

2. 远程串口访问方法

首先利用官方的 USB 线连接树莓派 USB 端口与开发板的 Console 端口。之后为了能够使用户在本地远程访问到开发板的端口。利用内核模块 USBIP 协议，将树莓派上的端口暴露到网络上，使得远程用户可以透过网络访问到开发板上的串口，将串口连接到本地再进行操作。

3. 网络配置

在树莓派上有一套完整网络配置。

- 能够开机自动连接 Tsinghua-Secure WiFi。
- 配有 TFTP 服务器，能够让用户上传内核镜像文件到树莓派上，供开发板调试用。
- 利用网线连接开发板，与开发板共享一个局域网，可以使用 TFTP 服务。
- 配有 DHCP 服务器，能够分配静态 IP 给开发板，方便开发调试过程。
- 配有 NAT 服务，使开发板能够访问外网。
- 配置了一个响应网站，用来接收用户对开发板的远程控制请求。

4. HDMI 输出

首先将开发板的 HDMI 输出，用 HDMI 线加上 HDMI 转 USB 适配器，接入到树莓派的 USB 端口上。然后利用 *µstreamer*，把 HDMI 输出进行视频串流，远程用户可以通过网络访问到 HDMI 输出。

2.2 复现工作

以上的工作我都重新复现一遍，但有些地方是不同的，比如曹隽诚同学在树莓派上是使用 NIX 系统的。而考虑到大多数用户对于 NIX 系统都是不熟悉的，因此我选用了比较多人使用的 Ubuntu 系统来进行复现工作，也减低了之后会按照我的方案来实现的成本。

2.2.1 HiFive Unmatched 开发板

按照曹隽诚同学的方案来实现了如下功能：

1. 电源端口连接了树莓派的 GPIO 端口，能够成功被远程启动。
2. 安装了 U-Boot，能够成功启动并进入 Boot。
3. Console 端口连接了树莓派的 USB 端口，能够被远程访问，获取串口信息及写入内核。

2.2.2 RaspberryPi 树莓派

由于在树莓派上安装了不同的系统，因此即使有一个可执行的方案，所有的功能也必须要重新自己实现。在环境配置这里的工作量也是挺大的。

1. 远程控制开发板启动方法

编写了一个 Python 脚本，用于控制树莓派上的 GPIO 端口输出。之后编写了一个 Flask 网页接收用户的 GET 请求，成功通过树莓派启动开发板。

2. 远程串口访问方法

按照曹隽诚同学的方法，在 Ubuntu 上配置 USBIP 协议服务，把端口暴露到网络上，成功把串口连接到本地进行调试。

3. 网络配置

- 编写 wpa_supplicant 文件，使 Ubuntu 能够在开机时自动登陆并连接 Tsinghua-Secure。
- 编写系统配置文件，配置了 TFTP 服务器，成功上传内核镜像文件到树莓派上。
- 编写系统配置文件，配置了 DHCP 服务器，通过网线连接，可动态为每个开发板分配 IP。
- 编写路由配置文件，配置了 NAT 服务，使开发板能够成功访问外网。
- 配置了一个响应网站，成功接收用户对开发板的远程控制请求。

4. HDMI 输出

利用 *μstreamer*，成功进行视频串流。

实现以上功能后，基本复现了曹隽诚同学的方案。复现后的整体架构见图 2.3。

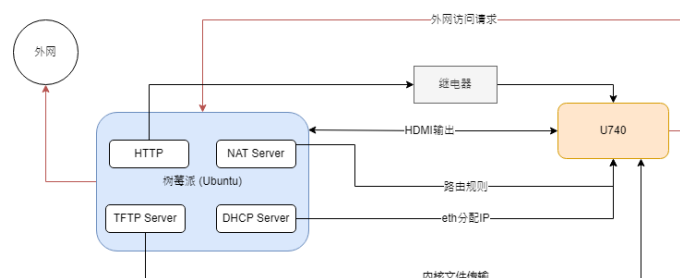


图 2.3 复现整体架构示意图

第 3 章 线上平台设计

3.1 系统组成

首先介绍平台的系统组成。本次工作的系统组成如下图 3.1。在前端中主要负责与用户的交互，比如接收用户信息，展示网页让用户进行在线的预约操作等。前端所使用的环境为 Moodle 3.9。在后端中主要负责管理用户信息，包括接收并更新数据，储存用户信息、预约信息、Docker 信息，并分配虚拟环境以及对开发板的资源调度等。使用的环境为树莓派，使用的系统为 Ubuntu Server 20.04。虚拟环境的搭建是使用 Docker 来完成的，数据库选用 MySQL。

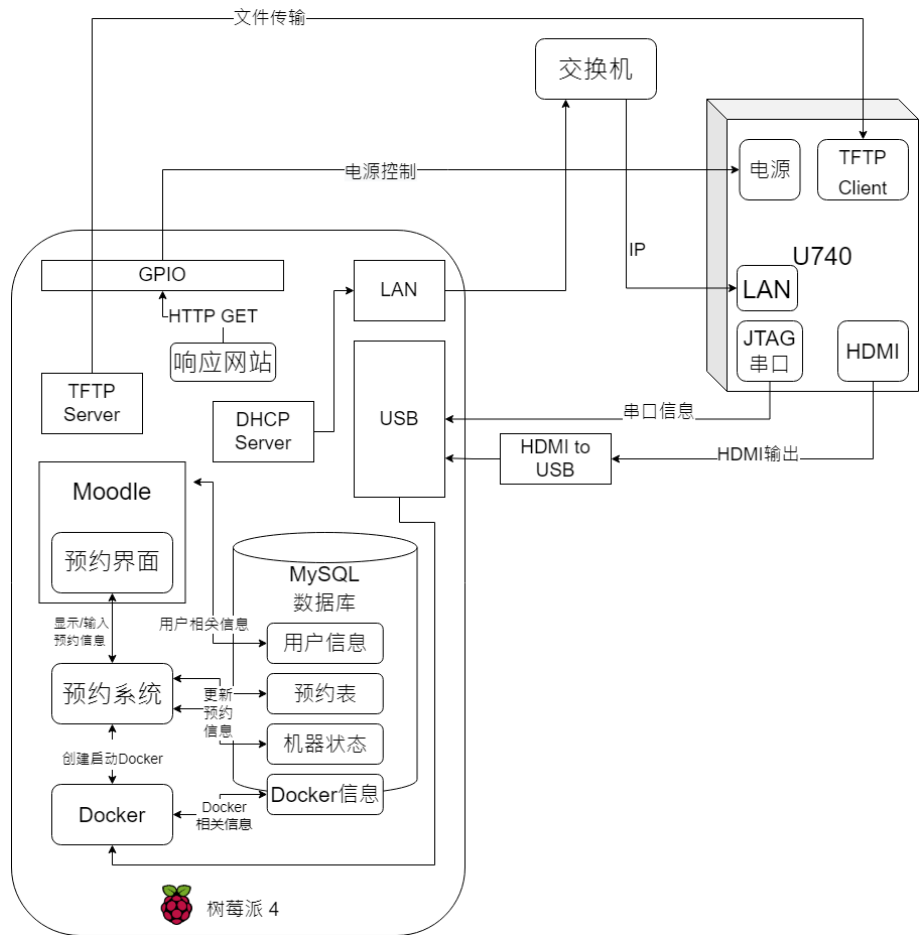


图 3.1 系统组成示意图

3.1.1 系统模块化功能总结图

整个系统大概可分为 9 个模块，其功能如下：

1. 用户交互模块：供用户申请以及预约使用机器
2. USBIP 调试模块：提供用户串口调试
3. Docker 调试模块：提供用户调试环境以及串口调试
4. OpenOCD 模块：提供用户 U-Boot 调试
5. 网络模块：提供文件传输以及分配 IP 功能
6. 数据管理模块：保存 Docker 信息、用户信息以及机器预约信息
7. 资源管理模块：提供机器使用调度
8. HDMI 模块：提供 HDMI 输出功能
9. FU740 控制模块：控制 FU740 的电源开关

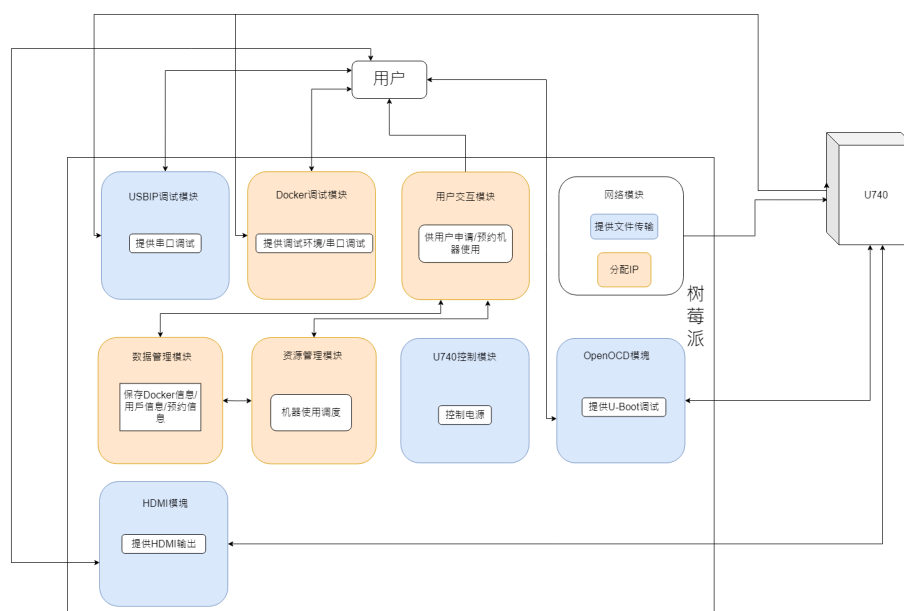


图 3.2 系统模块化功能总结图

其中，USBIP 调试模块、HDMI 模块、FU740 控制模块、OpenOCD 模块、TFTP 文件传输模块皆为前人工作，经改良后重新实现。其余为本工作的原创部分。

3.2 改进工作

基于上一章节的复现工作，我对此功能进行了一些改进。

3.2.1 更多的机器控制额

已实现的方案里最多支持两块开发板同时工作。现在把两个单路继电器换成一个四路继电器，使得理论上最多可支持四块开发板同时工作。增加更多用户调试的可能性。目前由于设备数量有限只配置了三块开发板，经测试可同时工作。

3.2.2 获取 IP 地址

由于校园网是动态的分配 IP 地址。因此每天树莓派启动时都有机会分配到不同的 IP 地址。因此编写了一个可以获取 IP 地址的脚本，在树莓派开机时会发送一封带 IP 地址的邮件到邮箱里。这样便可以从邮件得知树莓派 IP 地址然后远程登录到树莓派进行维护管理。该脚本适用于任何的网路场景。

3.2.3 交换机

由于树莓派原生的 LAN 端口只有一个，为支持更多的开发板能够使用网络来调试，添加了一台交换机。用树莓派作为每个开发板的 IP 分配 DHCP 服务器以及可以使用 TFTP 服务器。改进后的整体架构见图 3.3

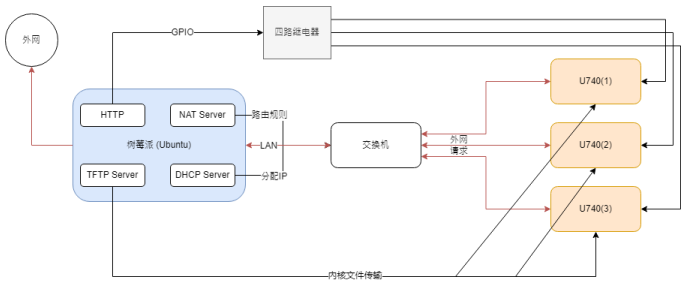


图 3.3 改进后整体架构示意图

3.3 线上平台需求分析

在进行平台实现之前，先对线上平台的执行方案进行构思，将会对线上平台所需的功能进行详细的分析。主要会从两个角度出发：第一个角度是从用户出发，分析用户在使用平台所需要的功能以及所遇到的问题和困难；另一角度是从平台管理员出发，分析管理员所需要用到的功能并尽量减低对技术的要求，能够更轻易上手进行开发和维护管理。

3.3.1 用户端需求分析

3.3.1.1 用户使用流程

从用户角度来看。首先最需要的是有一个能够连接调试环境的方法。以目前已搭建的环境来说，用户端的环境基本已经可以正常使用了。目前用户端的使用流程：

1. 等待服务器把串口暴露到网络。
2. 利用 USBIP 协议把空闲的串口连接到本地。

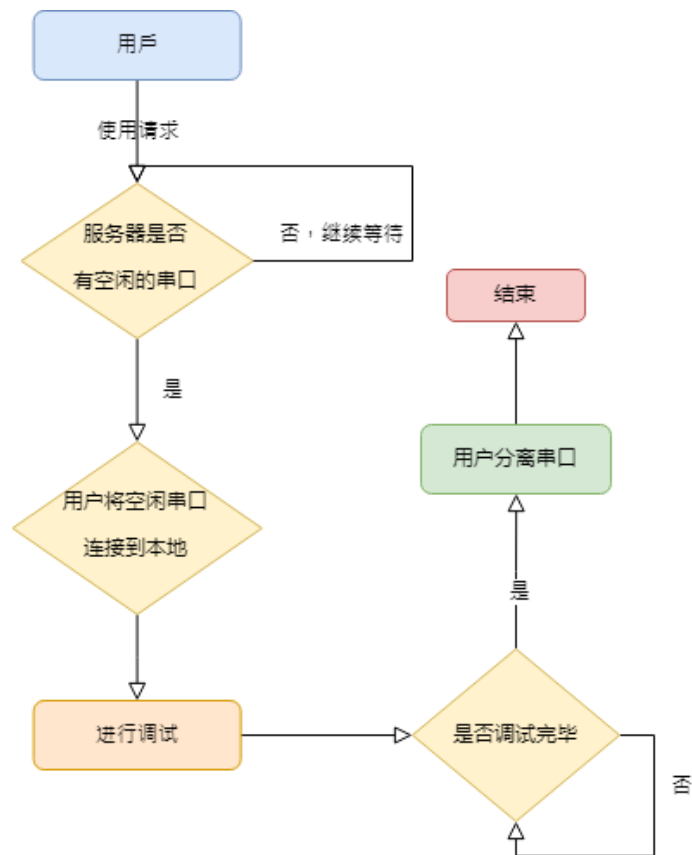


图 3.4 用户调试流程

3. 利用 picocom 等串口调试工具，连接到串口上获取串口信息。
4. 利用 TFTP 服务上传内核文件到开发板上进行调试。
5. 调试结束后，将串口从本地分离。

3.3.1.2 用户需求与问题提出

这样的流程虽然可以正常工作，但在实际使用的时候却存在不少的改进和问题。

比如一个比较影响用户体验的步骤，就是使用 USBIP 协议的话，在串口使用完毕后需要手动把串口从本地分离。如果没有手动分离的话，除非管理员介入，否则会导致其他用户无法使用该串口。在一个普通的平台里这一步通常都是由系统来完成或者会自动在后台完成的。

第二个问题就是 TFTP 服务器。在使用 TFTP 服务的时候常常会因为网络原因从而导致在远程调试时会出现连接超时的情况。比如在本地直接向树莓派使用 TFTP 服务，经常会出现连接超时的情况。但是如果在同一个局域网内的话则可以大大的避免连接超时的情况出现。

另外，在这个调试流程中，用户还需要自行去配置环境，比如需要安装 USBIP 协议所需要用到的一些内核模块、安装 picocom 库以及 TFTP 服务的客户端等等，这些过程都比较费时间而且还有机会因为系统版本而不能使用，使得用户体验变差。

因此，可以大致得出用户端的有如下需求与遇到的问题：

1. 需要改进串口的使用方法，即改进 USBIP 协议的使用，比如协商好每个用户都需要遵守手动分离串口，或者由系统来处理这些操作。
2. 改进 TFTP 服务，使非局域网的 TFTP 传输文件变得可使用以及更可靠。
3. 需要一个已经配置好能够使用的环境，让用户不需要再去自行安装相应的库来配置相关环境。

3.3.2 服务器端需求分析

3.3.2.1 服务器端流程

目前的服务器端主要是以下的流程：

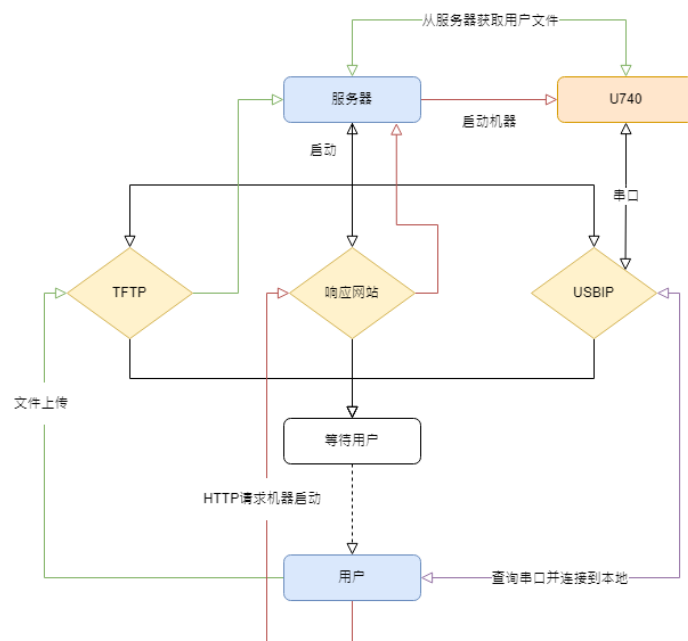


图 3.5 服务器启动流程

1. 服务器启动时同时启动 TFTP 服务、响应网站以及 USBIP 协议服务，将与开发板相连的串口暴露到网络上。
2. 等待用户。
3. 当有用户发出 HTTP 请求时，响应网站根据请求启动相应的开发板。

4. 当有用户使用 TFTP 上传文件时，服务器将文件存放在服务器中，供用户在开发板里读取使用。

3.3.2.2 服务器需求与问题提出

现阶段的流程以及实现比较简单，但服务器已提供了最基本可远程调试的环境。对于服务器的需求，首要任务就是要先解决用户在使用上的问题并进行优化。

第一个是对远程连接串口的方式进行优化。使用 USBIP 协议来暴露串口，继而进行远程连接，完全是依靠用户自主来连接与分离，假使有用户忘记分离而导致串口处于无法访问的状态的话只能由管理员强制分离再重新暴露接口。这无疑对于用户或者是系统管理员来说都是不方便的，因此需要有另一种方法来代替这种需要手动连接与分离的方法。

第二个就是解决 TFTP 连接超时的问题。对于这个问题，首先想到的是让用户可以登陆到服务器上，然后将需要的文件直接放到 TFTP 服务器中，这样便可以解决了连接超时的问题。但显然，这样做的话会因为泄露了服务器的使用权限从而构成安全隐患。因此，需要找到一种可以使用户透过某种方式连接在局域网之内进行调试的方法，这样便可以很好利用 TFTP 服务器。

第三个是用户调试环境的预配置，大多数的用户都会希望能够更好的把时间花在调试而非把时间都浪费在配置环境上，因此服务器需提供一个已配置好的环境，能够让用户登陆并直接使用，而且还应该能够保存每个用户自己的信息。

另外，还需要有一个策略来进行资源的调度。所谓资源指的就是开发板的使用权。目前的做法是，向设备管理员报备预约，然后再由管理员去作出分配。显然这种做法只适用于用户数量以及资源数量都比较少的情况。当用户数量以及比较大的时候，管理员会无法处理。因此需要有一种比较合理的预约方法，应当是比较系统的去分配，比如让用户填写一个预约时间表来预约使用时间然后由系统来分配，在分配时动态查看开发板的使用情况然后再去分配。

3.4 模型逻辑构思

在查阅了相关资料后，大致构思出以下的模型。

3.4.1 用户端模型

对于用户端，有如下的模型构思：

在此模型中，简化了用户使用的难度，其次也改良了用户使用开发板的方法。

1. 用户只需申请由服务器提供的虚拟环境并使用，无需自己进行额外的环境配置。
2. 用户的串口使用由服务器的虚拟环境提供，无需再主动连接串口，只需进行调试即可。
3. 用户只需要连接虚拟环境，无需再担心网络因素而导致无法使用 TFTP 上传文件。
4. 用户使用开发板前需要到服务器上进行预约，并根据预约的时间使用开发板。

3.4.2 服务器端模型

对于服务器端，有如下的模型构思：

1. 服务器中生成统一的虚拟环境供用户使用。
2. 服务器在虚拟环境中已连接开发板的串口，用户无需手动连接。
3. 虚拟环境直接在服务器中运行，使虚拟环境与 TFTP 服务器处于同一局域网，确保 TFTP 能够正常使用。
4. 服务器上运行一个预约系统，供用户按自己需求预约使用开发板，无需管理员介入。
5. 服务器上搭建一个数据库，用来储存用户的相关信息。

3.4.3 构思流程

3.4.3.1 用户端

基于上述模型设计出一个用户使用流程如下：

1. 用户于服务器上申请使用虚拟环境。
2. 用户通过预约系统进行开发板的使用预约。
3. 当到达预约时间，用户登陆到虚拟环境进行调试。
4. 当预约时间截止，用户调试结束。

对比初始的流程，改进后的流程对用户的操作需求明显降低了。用户只需申请虚拟环境以及预约时间即可使用。无需自行连接串口等操作。

3.4.3.2 服务器端

基于上述模型设计出一个服务器工作流程如下：

1. 服务器启动时同时启动 TFTP 服务、响应网站、数据库。
2. 数据库启动后启动预约系统。
3. 等待用户。

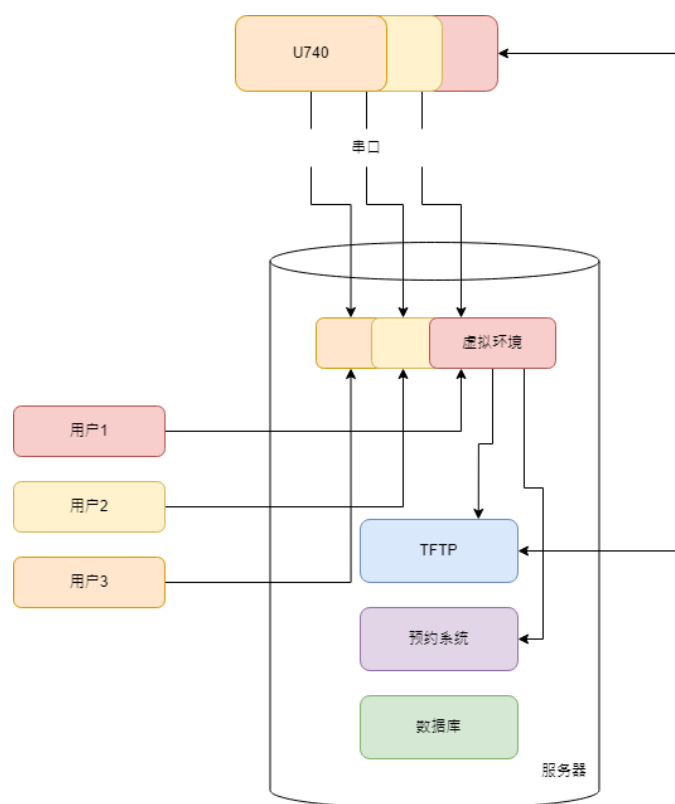


图 3.6 用户-服务器模型

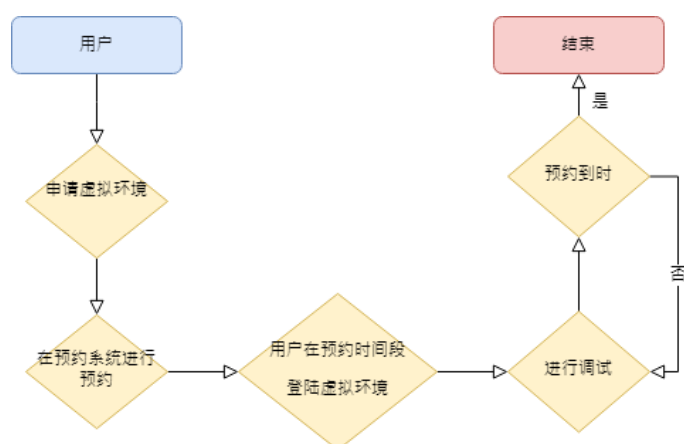


图 3.7 用户流程构思

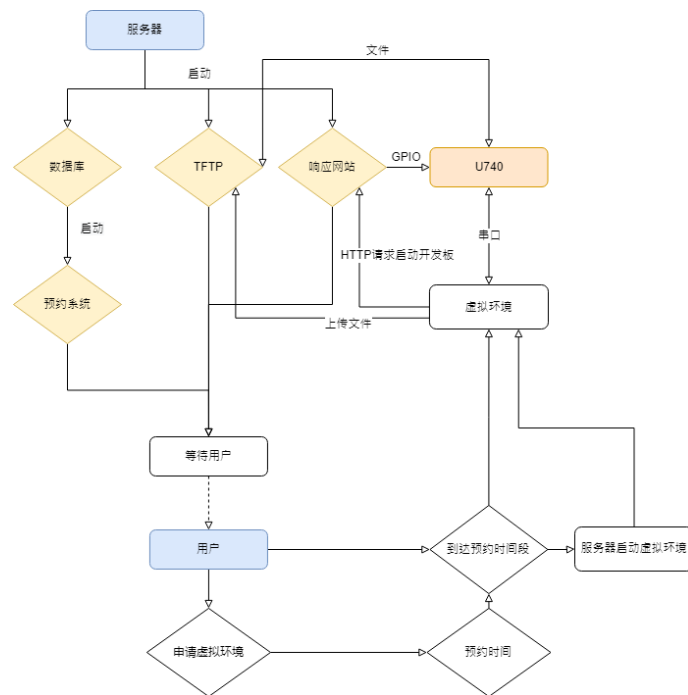


图 3.8 服务器工作流程构思

4. 当有用户申请虚拟环境时，根据用户信息分配虚拟环境。
5. 当有用户利用预约系统进行预约时，将预约信息录入，更新数据库。
6. 当预约时间到时，服务器启动相应的虚拟环境。
7. 当预约时间到时，用户登入虚拟环境。

对比初始的流程，服务器端作出了非常大的改动。改动主要是在资源管理上。在设计中加入了一套管理系统，包括数据库以及预约系统。数据库主要用来储存用户的信息，包括帐户名称、帐户密码、预约表信息等。而预约系统主要依靠数据库，通过查询以及更新数据库信息来完成自动预约功能。此外，在预约系统里还会加入一个前端与用户交互，用来接收用户信息并保存到数据库。

第 4 章 线上平台实现

4.1 线上平台的选择

经调研后^[5-8]，选择使用 Moodle 作为本次工作的线上平台框架。选用 Moodle 平台的原因有如下几点：

1. Moodle 是一个开源的线上教学平台，具备现成的课程系统，拥有登录界面，用户管理，课程管理等操作，免去自己再搭建数据库、网页并且将彼此连接起来的步骤。
2. Moodle 具有很高的可扩展性，所提供的网页模板亦十分容易上手，可轻易添加自己需要的内容，比如超链接、网页、文件等等，适合本次工作所需要的预约系统。
3. 世界上有不少的大学都选用了 Moodle 作为教学平台，比如哈尔滨工业大学的操作系统课就选用 Moodle 作为教学平台，因此证明 Moodle 在操作系统课的执行上具有可行性。

4.2 服务器端

4.2.1 Docker 虚拟环境

4.2.1.1 Docker 部署

Docker 是一款开源的开发软件，多用于开发应用、交付应用、执行应用等。Docker 允许使用者将基础的配置应用分割出来，用以提高交付速度。经调研后^[9-12]，与本次工作中对虚拟环境需求吻合，本次工作中搭建虚拟环境就是为了减低用户的操作需求，加快用户投入调试环节，故部署 Docker 来为用户分配虚拟环境。

4.2.1.2 Docker 配置与资源调度

Docker 旨在生成一个能够让用户调试的最低配置。在用户申请时即完成以下的安装以及配置如下：

1. 安装 picocom，用来调试串口。
2. 安装 TFTP 客户端，用来向服务器传送文件供开发板使用。
3. 安装 OpenSSH，用来实现 SSH 连线。
4. 复制配置文件，如 SSH 自启动文件、提示语句、script 脚本、SSH 公钥文件

等。

```
1 FROM ubuntu:20.04
2
3 ARG DEBIAN_FRONTEND=noninteractive
4
5 RUN apt-get update \
6     && apt-get install -y iputils-ping \
7     && apt-get install -y net-tools \
8     && apt-get install -y picocom \
9     && apt-get install -y vim \
10    && apt-get install -y tftp-hpa \
11    && ln -fs /usr/share/zoneinfo/Asia/Shanghai /etc/localtime \
12    # for SSH
13    && apt-get install -y openssh-server \
14    && apt-get install -y openssh-client \
15    && apt-get install -y ssh
```

图 4.1 Docker Build 文件示例

使用 Docker 来达到资源调度的效果，实现方法主要如下：

1. 每个用户分配一个 Docker 镜像。在运行时才生成并且搭载串口。
2. 所有 Docker 都使用统一配置，除了 SSH 公钥以及对外端口，保证每个用户都能使用不同的端口。
3. 每个 Docker 只连接一个开发板的串口，确保开发板间互不干扰。
4. Docker 在退出后会自动删除，便于动态运行，不占服务器空间。

在示例图 4.2(a)中，可以看到服务器存有不同的镜像。以其中一个容器“admin”来作说明。首先可以看到镜像名称是“u740:admin”，每个用户都有他专属的镜像，这是由于用户的 SSH Key 文件需要在 Docker build 阶段就以参数的形式传入到 Docker 里，否则如果在 Docker run 阶段才传入的话会因为文件存取权限问题无法使用免密登陆。而容器名称，见图 4.2(b)，亦与帐号名称相同，方便管理员进行辨认。

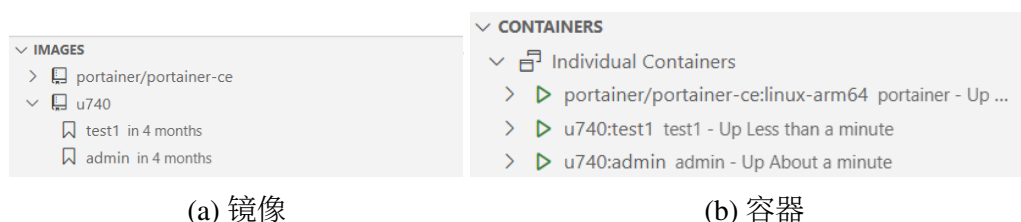


图 4.2 Docker 分配示例

4.2.1.3 用户帐号信息

对于 Moodle 的帐号，参考了不少需要使用集群以及服务器的课程，主要采取由管理员统一分配策略。使用这种方法的话，可以单对单的向用户邮箱发送帐号信息，确认信息正确以及确保用户为课程学生。而不是让用户自行去申请帐号，可以随意登入。另外，对于密码的处理，目前是先使用统一初始密码，在初始登陆时

强制用户更改密码。

Moodle 上的用户信息主要包括以下几项：帐号、密码、邮箱、姓名。

表 4.1 Moodle 用户信息示例

Moodle 用户信息	内容
帐号	admin
密码	passwd
邮箱	admin@test.mail
姓名	admin

目前 Moodle 帐号的申请流程是：

1. 管理员根据用户提供的信息（包括姓名、电话、邮箱），为用户创建 Moodle 帐号，并设置强制更改密码。
2. Moodle 帐号创建后，把帐号和初始密码信息发送到用户的邮箱里。
3. 用户在第一次登陆 Moodle 时，必需更改初始密码。

4.2.2 用户文件保存

在使用虚拟环境时，很多情况下都需要往虚拟环境里暂存一些文件但又不想每次启动都被删除。因此设立一个目录让用户可以在此保存文件，即使环境遭删除或需要重置用户所需的文件仍然会保存在服务器上直到管理员将他们清理。对于是否存在文件复写问题，由于同时间每个用户最多只会有一个 Docker 在运行，因此不会存在文件复写问题。

一个用户保存的示例图。图中的例子，模拟了一个保存文件的过程：左侧是服务器端，右侧是用户登陆到 Docker 后的界面。在一开始用户文件夹里有三个文件分别是“a.txt”，“b.txt”和“test.txt”，然后在 Docker 端新增一个文件“admin.txt”内容为“testing”。此后在服务器端便可以同步文件，可以查看到内容为“testing”的文件“admin.txt”。

4.2.3 用户连线安全保障

对于虚拟环境，用户连线安全亦是一个很重要的一环。目前用户登陆到 Docker 的方法是使用 SSH 连线。SSH 是一种加密的网路传输协定，最常见的用途是于远端登录系统。在普通的密码登陆的基础上再使用 RSA 加密，让用户提供公钥，然后使用公钥登录到 Docker，不使用密码登陆，可以更大程度的保障用户连线的安


```

ubuntu@piroot:/docker_usr/user/admin$ date -R
Mon, 10 Jan 2022 10:03:08 +0800
ubuntu@piroot:/docker_usr/user/admin$ ls
a.txt b.txt test.txt
ubuntu@piroot:/docker_usr/user/admin$ date -R
Mon, 10 Jan 2022 10:03:28 +0800
ubuntu@piroot:/docker_usr/user/admin$ ls
a.txt admin.txt b.txt test.txt
ubuntu@piroot:/docker_usr/user/admin$ cat admin.txt
testing
ubuntu@piroot:/docker_usr/user/admin$

```

```

root@cc51356b0aee:~/admin# date -R
Mon, 10 Jan 2022 10:03:03 +0800
root@cc51356b0aee:~/admin# ls
a.txt b.txt test.txt
root@cc51356b0aee:~/admin# echo "testing" >> admin.txt
root@cc51356b0aee:~/admin# date -R
Mon, 10 Jan 2022 10:03:26 +0800
root@cc51356b0aee:~/admin#

```

图 4.3 用户文件保存示例

全。

具体实现的做法是，用户在 Moodle 注册 FU740 帐号时，同时提供 SSH 公钥信息，然后在 Docker Build 时把公钥导入到 Docker 里面，这样用户在登陆 Docker 的时候便可以免密登入。

```

PS C:\Users\user> ssh -p 20000 root@183.173.18.234
root@cc51356b0aee:~#

```

图 4.4 SSH 免密登陆

4.2.4 用户行为分析记录

作为管理员，如果能够记录用户使用 Docker 调试时的操作，输入输出结果再加以分析，便能获得一个很好的反馈从而更好的提升实验的质量以及平台的改良。因此有了这项实现。在一开始是选用 Ubuntu 原生的命令“history”来实现。history 命令能够记录用户在命令行上的输入，然后在查询时返回用户所有的命令记录。但 history 存在一定的缺点，比如有限的记录次数、无法记录命令行输出等。因此，经过查询相关资料后选用了另外一个原生的命令“script”来实现。script 能够很好的记录用户在终端上的输入输出信息，而且能够以静默模式运行，在用户不察觉的情况下记录用户的所有操作。

```

ubuntu@piroot:/docker_usr/info/test1/logs$ cat session_2.log
log
loglog
Tue May 10 11:00:06 CST 2022 Script started on 2022-05-10 11:00:06+08:00 [TERM="xterm" TTY="/dev/pts/1" COLUMNS="92" LINES="37"]
Tue May 10 11:00:10 CST 2022 root@8045de478fec:~# ls
Tue May 10 11:00:10 CST 2022 test1
Tue May 10 11:00:13 CST 2022 root@8045de478fec:~# exit
Tue May 10 11:00:13 CST 2022 exit
loglogloglog
Tue May 10 11:03:33 CST 2022 Script started on 2022-05-10 11:03:33+08:00 [TERM="xterm" TTY="/dev/pts/1" COLUMNS="80" LINES="24"]
Tue May 10 11:03:39 CST 2022 root@2da87c589650:~# ls /dev/ttyUSB*
Tue May 10 11:03:39 CST 2022 /dev/ttyUSB0
Tue May 10 12:00:16 CST 2022 root@2da87c589650:~# exit
Tue May 10 12:00:16 CST 2022 exit
ubuntu@piroot:/docker_usr/info/test1/logs$

```

```

14 adddate() {
15     while IFS= read -r line; do
16         printf '%s %s\n' "$(date)" "$line";
17     done
18 }
19
20 /usr/bin/script -fq >{ adddate >> /etc/session.log} && exit

```

(a) 配置文件

(b) 记录示例文件

图 4.5 script 示例

4.2.5 机器使用预约系统

本项工作的重点开发内容便是开发机器使用预约系统。机器使用预约系统主要由两个部分组成，分别是预约优先分配算法以及机器状态更新算法。

而预约表的更新是每一个小时更新一次，其更新主要有两项：机器的使用权限更新以及连接到时提醒。由于开发板的资源有限，不足以让每个用户都有专属的开发板，因此必需要动态分配。

在服务器的数据库里加入了三个表，分别用来储存用户 Docker 信息，当天预约时间表以及当前机器使用状态。

第一个表是用户 Docker 信息，内容格式见表 4.2，记录了用户在注册 FU740 帐号时的 Moodle 用户 ID(usr_id)、用户名称 (usr_name) 以及所分配的端口号 (machine_port)。

表 4.2 用户 Docker 信息格式示例

usr_id	usr_name	machine_port
2	admin	20000
4	test1	20001

第二个表是预约时间表,内容格式见表 4.3,记录了当天用户的预约情况,表项有时间段 (period)、开发板 1 的预约用户 (machine_1)、开发板 2 的预约用户 (machine_2) 以及开发板 3 的预约用户 (machine_3)。当每次有用户进行预约后都会实时同步更新用户的预约情况。目前将一天划分为 12 个时间段，预约时间从早上九点开始，到晚上九点结束预约。使用时间从早上十点开始，到晚上十点结束，每个时间段为一小时。在时间段开始后不能够再进行该时间段的预约。

表 4.3 预约时间表

period	machine_1	machine_2	machine_3
1	test1	admin	NULL
2	NULL	admin	NULL
.	.	.	.
.	.	.	.
11	NULL	NULL	NULL
12	NULL	NULL	NULL

第三个表是当前机器使用状态，内容格式见表 4.4，记录了当前时间段机器的使用情况。表项有时间段 (period)、开发板 1 的用户 (machine_1)、开发板 2 的用户 (machine_2)、开发板 3 的用户 (machine_3)。用户 1 所使用的端口 (port_1)、用户 2 所使用的端口 (port_2)、用户 3 所使用的端口 (port_3)。该表能够方便管理员查询目前时间段正在使用的机器以及其端口信息。

表 4.4 机器使用状态

period	machine_1	machine_2	machine_3	port_1	port_2	port_3
2	NULL	admin	NULL	0	20000	0

当用户在预约界面上进行预约操作后，服务器会执行预约优先分配算法为用户分配机器。当到达定时更新时间段时再执行机器状态更新算法更新机器状态。

4.2.5.1 预约优先分配算法

对于预约优先分配算法，以用户输入的用户名称以及选择的时间段作为输入参数，然后输出预约结果。

具体过程为：首先从用户输入获取待分配的用户名称，然后获取用户所预约的时间段。通过对比数据库查看用户是否存在以及用户是否已进行过预约。

若果用户不存在或在该时段已进行过预约，返回失败。否则进行下一步判断。判断用户在相邻时间段内是否有预约并优先分配机器。即在上一时段如果已有预约，在下一时段优先预约使用相同机器，让用户在相邻的时间段里可以使用相同的机器。因为如果在使用过程中每次都更换机器会十分影响用户体验，故采取这种比较合理的处理方法。对此，首先判断是否预约第一个时间段，如果预约时间段为第一个时间段由于没有更前的预约，故按默认从 1 号机开始分配机器。

若果不在第一时段，则获取上一个时段的用户群，查看目前预约的用户是否在该用户群中。若果不在，则直接分配空闲机器，更新预约表，返回成功。若果在，则记录该用户所使用的机器，并查看当前预约时间段是否已有人预约该机器。若无则直接分配该机器，更新预约表，返回成功；若有，则交换两者的机器预约并更新预约表。返回成功。

预约优先分配算法伪代码可见算法 4.1。

算法 4.1 预约优先分配算法

INPUT: *username, selected period***OUTPUT:** *appointment_result*

```
1: username ← INPUT
2: period ← selected period
3: user_exist ← database check username
4: user_appointment ← username in same period
5: if user_exist is not None and user_appointment is False then
6:   previous_user = None
7:   if period != 1 then
8:     previous_user ← timetable
9:   end if
10:  if username in previous_users then
11:    previous_machine ← previous_users(username)
12:    spare_machine ← a spare machine
13:    reserved_user ← timetable(previous_machine)
14:    if reserved_user is not None then
15:      switch the machine
16:      user ← reserved_machine
17:      reserved_user ← spare_machine
18:      update timetable
19:    else
20:      distribute the previous machine
21:      update timetable
22:    end if
23:  else
24:    distribute a spare machine normally
25:    update timetable
26:  end if
27:  return success
28: else
29:  return fail
30: end if
```

4.2.5.2 机器状态更新算法

对于机器状态更新算法，首先获取当前时间段。对于第一个时间段不用进行判断，如果预约时间表不为空，更新机器状态即可。对于之后的时间段，需要判断 1. 下个时间段是否有预约 2. 是否为同一人预约 3. 是否使用同一机器。首先获取下个时间段的预约时间表，判断是否有人预约。

若下个时间段没有人使用，则检查当前已分配的用户是否仍在使用的。具体检查方法为检查 SSH 连接数，若连线数为零，则表示当前正在启动的 Docker 没有人在使用，停止该 Docker，更新机器使用状态。若连线数不为零，代表仍有人在使用该 Docker。鉴于下个时间段没有人使用，该机器仍分配给该用户使用，不停止该 Docker，直至下个时间段更新有人使用为止。采用这种方法可以最大化机器资源，让有需要的用户可以延长使用时间，直到有新的预约出现。

若下个时间段有人使用，则检查下个时间段预约的用户是否有现在正在使用的用户。若有，则检查这些用户是否使用同一机器。若为同一机器则不停止该 Docker，否则停止该 Docker。虽然上述的预约分配算法保证了相邻时间段同一用户的预约一定会分配到同一机器，但在最大化机器资源的原则下，则允许了用户预约一次能使用若干个时间段的情况。比如以下一种情况：用户 1 预约了时间段 1 后开始使用。在时间段 2 时用户 2 预约了时间段 3，用户 1 也在此时预约了时间段 3。预约情况如下表 4.5。由于预约分配优先化只作用于相邻的时间段，在用户 1 预约时间段 3 时由于机器 1 已被用户 2 分配，且上个时间段 2 亦没有预约记录，故被默认分配空闲机器 2。此时如果在更新机器状态时只检查预约用户是否与正在使用的用户为同一用户，则会发生冲突。由于用户可以在没有预约的情况下一直使用机器，因此加入这个判断解决了这种冲突情况的发生。

表 4.5 模拟预约时间表

period	machine_1	machine_2	machine_3
1	user1	NULL	NULL
2	NULL	NULL	NULL
3	user2	user1	NULL

若无现在正在使用的用户，则停止所有 Docker。然后根据预约时间表启动下个时间段的 Docker，更新机器使用状态。

机器状态更新算法伪代码可见算法 4.2。

算法 4.2 机器状态更新算法

INPUT: *timetable, machine_status***OUTPUT:** *machine_status*

```
1: period ← localtime
2: GET next_usernames
3: if period == 1 then
4:   if next_usernames is not None then
5:     for all user in next_usernames do
6:       START Docker
7:       UPDATE machine_status
8:     end for
9:   end if
10: else
11:   GET current_usernames
12:   if next_usernames is not None then
13:     if current_usernames is not None then
14:       for all current_user in current_usernames do
15:         if current_user not in next_usernames then
16:           STOP Docker
17:         else
18:           GET current_machine
19:           GET next_machine
20:           if current_machine != next_machine then
21:             STOP Docker
22:           end if
23:         end if
24:       STOP Docker
25:     end for
26:   end if
27:   for all next_user in next_usernames do
28:     if next_username not in current_usernames then
29:       START Docker
30:       UPDATE machine_status
31:     end if
32:   end for
33: else
34:   for all current_user in current_usernames do
35:     CHECK SSH connection
36:     if ssh_users_number == 0 then
37:       STOP Docker
38:       UPDATE machine_status
39:     end if
40:   end for
41: end if
42: end if
```

4.2.6 排程服务器后台实现

上述定时的预约表更新主要是依靠 Flask-APScheduler 来实现。Flask APScheduler 是 Flask 的一个排程工作库，可以用来完成一些定时的任务。在本次工作中，设置了从每天的早上 10 点开始，在每整点都会执行一次机器状态更新算法来更新机器状态，在每小时的 55 分，都会向用户发送提示信息，提醒用户只剩下五分钟的 Docker 使用时间。对于此提醒的实现主要是通过服务器进入 Docker 环境里面执行 Ubuntu 原生的广播“wall”指令，用户会收到一组提示语句以及提示音效。提示效果如下图 4.6

```
Broadcast message from root@cc51356b0aee (somewhere) (Mon Jan 10 10:55:01 2022)

:: Docker will shutdown in 5 minutes if you do not make an appointment in the
next period, files will be vanished if you do not save it in your directory
::

root@cc51356b0aee:~/admin#
```

图 4.6 Docker 到时提醒

4.2.7 Moodle 安装

按照 Moodle 官网在 Ubuntu 上的 Step-by-Step 安装，把 Moodle 安装在服务器上，Moodle 网站会随服务器启动而启动。

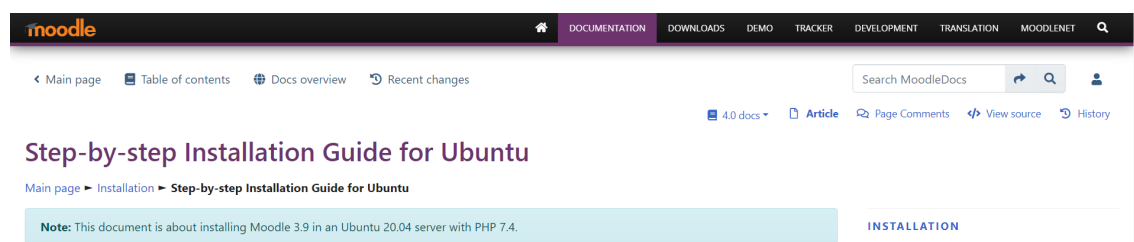


图 4.7 Moodle 安装流程

4.3 用户端

4.3.1 响应网站

利用 Flask 构建了一个简单的网页，配合 Python 的模块控制树莓派的 GPIO 输出。透过用户输入不同的访问路径，然后作出对应的 GPIO 输出。比如对一号机的控制示例路径如图 4.8。后缀/poweron1 代表启动 1 号机、/poweroff1 代表关闭 1 号机。

```
ubuntu@piroot:~$ curl http://183.173.18.234:8182/poweron1
<p>Power on machine-1!</p>ubuntu@piroot:~$
ubuntu@piroot:~$

ubuntu@piroot:~$ curl http://183.173.18.234:8182/poweroff1
<p>Power off machine-1!</p>ubuntu@piroot:~$
ubuntu@piroot:~$
```

图 4.8 HTTP 请求示例

4.3.2 Docker 使用

用户只要在申请界面进行过申请并显示成功信息，且预约界面成功预约，在到达预约时间时便可以利用 SSH 连接到自己的 Docker 进行调试。

4.3.3 调试方式选择

目前是默认使用 Docker 来进行调试，但 Docker 环境没有配置图形操作界面。若果对图形界面有特殊需求的用户，亦可以选择使用 USBIP 协议，把串口连接到本地，然后再把文件从 Docker 上传到服务器这种方法来进行调试。

4.3.4 Moodle 交互界面实现

在用户端配置了三个交互式的界面，分别是申请 FU740 帐号、预约机器使用时间以及查询预约。



图 4.9 主页界面

4.3.4.1 FU740 帐号申请与界面

用户的 Docker 配置申请在该页面内完成，界面可见图 4.10。用户通过输入 Docker 用户名与 SSH 公钥来生成 Docker。

申请中会有提示信息，待申请成功后会提示申请成功讯息。为公平起见，约定每个 Moodle 用户只能申请一个 Docker 环境，当重复申请时会出现错误讯息，即使更改 FU740 帐号名称也不可重复申请。效果可见图 4.11

U740 机器使用预约系统

[个人主页](#) / [课程](#) / [U740 apply](#) / [机器连接权限申请](#) / [申请U740使用帐号](#)

申请U740使用帐号



申请需知

该项目 (U740线上调试环境) 目前尚处于测试阶段, 请输入用户名以及SSH Key后点击申请

用户名:

SSH Key:

申请

输入用户名及SSH公钥申请U740使用帐号

图 4.10 FU740 帐号申请界面

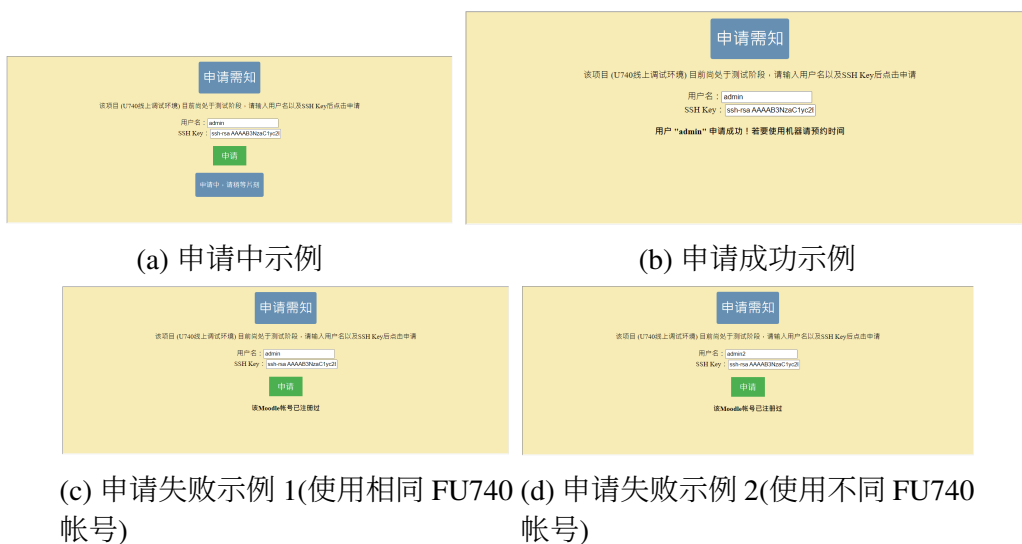


图 4.11 申请帐号示例

4.3.4.2 预约界面

用户的预约服务在此页面内完成，界面可见图 4.12。通过输入用户名并选择时间段来完成机器预约。



图 4.12 预约界面

主要通过 HTML+JavaScript 来实现此网页的功能，比如显示表格点选效果，弹出确认用户信息框等，具体效果见 4.13。



(a) 表格点选效果

183.173.17.47:8188 中的嵌入页面显示

请确认你的用户名为: admin
你所选择的时间段为: 16:00 --17:00
如有错请点击取消



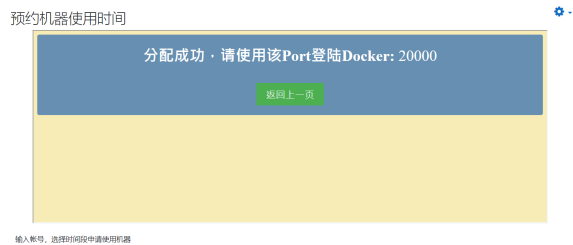
(b) 弹出确认用户信息框

图 4.13 预约讯息示例

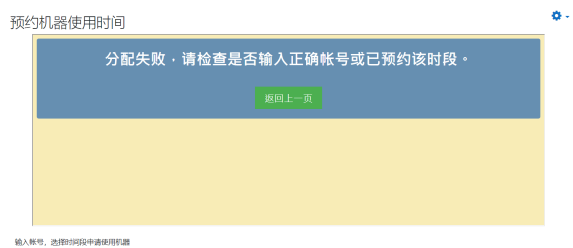
当用户成功预约时会出现预约成功界面，当用户预约失败时会出现错误讯息。为公平起见，约定每个用户在一个时间段内只能够预约使用一台机器，效果见图 4.14。

4.3.4.3 预约查询界面

当用户忘记了预约时间段的时候，可以到此网页输入 FU740 帐号进行查询，界面可见图 4.15。



(a) 预约成功示例



(b) 预约失败示例

图 4.14 预约讯息示例

U740 机器使用预约系统

[个人主页](#) / [课程](#) / [U740 apply](#) / [机器连接权限申请](#) / [查询预约时间及Port](#)



图 4.15 查询界面

查询成功时返回结果为时间段以及端口号；查询失败时返回错误讯息，效果可见图 4.16。

预约时间段及Port查询

输入用户名查询

用户名：

查询

用户 "admin" 预约情况如下：
时间段：17:00 -- 18:00 || Port: 20000
时间段：18:00 -- 19:00 || Port: 20000
时间段：19:00 -- 20:00 || Port: 20000

预约时间段及Port查询

输入用户名查询

用户名：

查询

请检查用户名称 "aaa" 是否正确！

(a) 查询成功

(b) 查询失败

图 4.16 查询示例

第 5 章 线上平台的应用

5.1 Hifive Unmatched 远程开发和 NXOS 移植-工作介绍

在本项工作的支持下，包括远程环境搭建、环境使用的技术支持、疑难排解等等。重庆师范大学的胡自成同学成功利用 HiFive Unmatched 远程开发完成 NXOS 移植。

5.2 应用成果

摘录自胡自成同学在 2022 全国操作系统比赛在线技术的报告。
系统启动 LOG

```
Load address: 0x84000000
Loading: #####
#####
10.9 MiB/s
done
Bytes transferred = 1331400 (1450c8 hex)
## Starting application at 0x80200000 ...
[ 0.000] SBI: OpenSBI v1.0
[ 0.000] SBI Specification Version: 0.3
[ 0.000] [INFO-LOG] Hello, Hifive-Unmached! on core 1
[ 0.000] [INFO-Page] Memory NX_Size: 10000000 Bytes 256 MB
[ 0.000] [INFO-Page] Normal memory: 0000000082000000~0000000089000000
NX_Size:112 MB
[ 0.000] [INFO-Page] User memory: 0000000089000000~0000000090000000
NX_Size:112 MB
[ 0.000] [INFO-Buddy] mem: 0x0000000082000000 size: 0x0000000007000000
[ 0.000] [INFO-Buddy] mem: 0x0000000089000000 size: 0x0000000007000000
[ 0.000] [INFO-Page] OS map early on [0000000080200000~0000000089000000]
[ 0.000] [INFO-Page] set MMU table: 000000008025D000
[ 0.000] [INFO-Page] MMU enabled
[ 0.000] [INFO-Page] Memroy init done.

_ _ _ _ _
| \ \ \ / _ \ _ |
|  \ \ /  / / \ \
| \  /  \ / \ | |
|  \ \ /  \ / \ |
|  \ \ /  \ / \ |

Kernel      : nxos
Version     : 0.1.0
Build      : May 4 2022
Platform    : riscv64-hifive_unmached
Copyright (c) 2018-2022, NXOS Development Team
[ 0.000] [INFO-PageCache] span base addr: 0000000082119000
[ 0.000] [INFO-PageCache] span mark used page: 28
[ 0.000] [INFO-LOG] stage2!
[ 0.000] [INFO-smp-riscv64] boot core is:1
```

图 5.1 U740 上的 NXOS 启动 LOG

NXOS 在 Hifive Unmached 平台上的移植

- FU740 有 5 个核，小核为 0，大核心为 1-4，小核 core 0 不能通过 SBI 启动（所以启动要从 hartid=1 开始）。
- MMU 的 PTE 需要加入 A 和 D 位才行。启用页表（写 satp）之后，程序就没反应了，这是因为 FU740 对页表项（PTE）中的 A、D 位有规定，它不支持硬件修改 A 和 D，并且一定要把 A 和 D 设为 1，否则会 page fault，这和之前做法把 A、D 设为 0 的是不一致的。
- 启动方式可能和标准的启动方式有差异，导致的问题就是 hartid 和设备树的传递问题，后续可以研究有没有更好的方式来引导，目前是用不上设备树的。
- PLIC 的偏移地址也比较奇怪，所以需要注意 PLIC 的地址和某块内存区域的大小，在 MMU 映射虚拟地址的时候需要注意。
- 内核的链接地址是 0x80200000，U-Boot 会跳转进入到该内存地址并执行

第 6 章 结论

6.1 工作总结

在本项工作中，主要研究对 FU740 的控制以及建立一个线上实验平台。

通过复现前人工作，一步一步透过自己的方法在新的树莓派上搭建自己的服务器。在这过程中作出了改进方案，如增加机器使用额、脚本获取动态 IP、增加交换机。在执行改进方案之后，开始了平台的设计构思，设计出服务器的框架，包括以 Moodle 作基础，搭建数据库，使用 Docker 作资源调度，预约机器分配策略。

最终对 U740 的控制达到了以下五项：GPIO 电源控制、HDMI 的输出、TFTP 文件传输、OpenOCD 调试 U-Boot 以及 JTAG 串口读取；线上平台提供交互界面并支持申请预约功能；服务器端实现预约优先分配算法以及机器状态更新算法。整合后成功实现了一个基于 FU740，利用 Moodle 作前端的线上交互平台。

在本工作中，已成功帮助了一名同学完成线上的开发。以在开发阶段的工作来看，算是取得一个十分不错的成绩。

6.2 工作展望

对于未来，可以将本工作结合其他操作系统实验课的评测机，实现线上调试和评测一体化。

插图索引

图 2.1	远程开机流程示意图	3
图 2.2	HiFive Unmatched 引导流程示意图	4
图 2.3	复现整体架构示意图	6
图 3.1	系统组成示意图	7
图 3.2	系统模块化功能总结图	8
图 3.3	改进后整体架构示意图	9
图 3.4	用户调试流程	10
图 3.5	服务器启动流程	11
图 3.6	用户-服务器模型	14
图 3.7	用户流程构思	14
图 3.8	服务器工作流程构思	15
图 4.1	Docker Build 文件示例	17
图 4.2	Docker 分配示例	17
图 4.3	用户文件保存示例	19
图 4.4	SSH 免密登陆	19
图 4.5	script 示例	19
图 4.6	Docker 到时提醒	25
图 4.7	Moodle 安装流程	25
图 4.8	HTTP 请求示例	26
图 4.9	主页界面	26
图 4.10	FU740 帐号申请界面	27
图 4.11	申请帐号示例	27
图 4.12	预约界面	28
图 4.13	预约讯息示例	28
图 4.14	预约讯息示例	29
图 4.15	查询界面	29
图 4.16	查询示例	30
图 5.1	U740 上的 NXOS 启动 LOG	31

表格索引

表 4.1	Moodle 用户信息示例	18
表 4.2	用户 Docker 信息格式示例	20
表 4.3	预约时间表	20
表 4.4	机器使用状态	21
表 4.5	模拟预约时间表	23

参考文献

- [1] Adzharuddin N A, Ling L H, et al. Learning management system (lms) among university students: Does it work[J]. International Journal of e-Education, e-Business, e-Management and e-Learning, 2013, 3(3): 248-252.
- [2] Al-Khanjari Z A, Al-Roshdi Y M. Extending the functionality of lms to support computer science education using plug-in tools[J]. Journal of Emerging Technologies in Web Intelligence, 2014, 6(2): 220-225.
- [3] Simanullang N, Rajagukguk J. Learning management system (lms) based on moodle to improve students learning activity[C]//Journal of Physics: Conference Series: volume 1462. IOP Publishing, 2020: 012067.
- [4] Hylén J. Open educational resources: Opportunities and challenges[M]. OECD, 2021.
- [5] Jin S. Design of an online learning platform with moodle[C/OL]//2012 7th International Conference on Computer Science & Education (ICCSE). 2012: 1710-1714. DOI: 10.1109/ICCSE.2012.6295395.
- [6] Al-Ajlan A, Zedan H. Why moodle[C/OL]//2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems. 2008: 58-64. DOI: 10.1109/FTDCS.2008.22.
- [7] Rice W, William H. Moodle[M]. Packt publishing Birmingham, 2006.
- [8] Zhigang S, Xiaohong S, Ning Z, et al. Moodle plugins for highly efficient programming courses [Z]. 2001.
- [9] Douce C, Livingstone D, Orwell J. Automatic test-based assessment of programming: A review [J]. Journal on Educational Resources in Computing (JERIC), 2005, 5(3): 4-es.
- [10] Anderson C. Docker [software engineering][J/OL]. IEEE Software, 2015, 32(3): 102-c3. DOI: 10.1109/MS.2015.62.
- [11] Rad B B, Bhatti H J, Ahmadi M. An introduction to docker and analysis of its performance[J]. International Journal of Computer Science and Network Security (IJCSNS), 2017, 17(3): 228.
- [12] Špaček F, Sohlich R, Dulík T. Docker as platform for assignments evaluation[J/OL]. Procedia Engineering, 2015, 100: 1665-1671. <https://www.sciencedirect.com/science/article/pii/S1877705815005688>. DOI: <https://doi.org/10.1016/j.proeng.2015.01.541>.

致 谢

衷心感谢导师向勇副研究员对本人的悉心指导。陪同我一起度过了这短暂的六个月。在这六个月当中，向勇老师不时的与我讨论并指点我在研究上正确的方向，使我在这几个月中获益匪浅。而且，向勇老师总能在我不解时为我指点迷津。也因为有向勇老师的批评和指正才能够完成这次的毕业设计与论文。

此外，也特别感谢曹隽诚同学。在我调试 FU740 的过程中，孜孜不倦地帮助我。解答我所有的疑问，使我能够顺利的完成我的设计项目。

最后，仅以此文祝贺我自己在如此严峻的疫情下能够保持正面的心态去完成这次的毕业设计。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： 唐皓冰 日 期： 2022/06/15

外文资料的书面翻译

AL-KHANJARI, Zuhoor A.; AL-ROSHDI, Yusra M. Extending the functionality of LMS to support computer science education using plug-in tools. *Journal of Emerging Technologies in Web Intelligence*, 2014, 6.2: 220-225.

Extending the Functionality of LMS to Support Computer Science Education Using Plug-in Tools

Zuhoor A. Al-Khanjari and Yusra M. Al-Roshdi Department of Computer Science, College of Science, Sultan Qaboos University Muscat, Oman

Email: zuhoor@squ.edu.om and m095653@student.squ.edu.om

摘要——电子学习是一个环境用以支持教育领域，并且也是一个提供特定领域知识的地方。计算机科学教育 (CSE) 是一个使用电子学习来在各个方面教育学生并分发知识的领域。一些学习管理系统 (LMS) 支持许多软件工具来管理 CSE 的理论材料。然而，在大多数 LMS 中，计算机科学课程所需的实用软件工具被忽略或根本不考虑。为了克服 LMS 在 CSE 实践方面的局限性，本文展示了如何准备工具并使它们可以通过 LMS 随时使用。为此，本文的作者在讨论如何扩展 LMS 的一般功能和 Moodle 作为特例方面向前迈进了一步。此扩展涉及将所需的软件工具作为组件添加到 Moodle 的架构结构中。此外，本文还介绍了一些支持和实现这些扩展的文献。

索引词——电子学习、计算机科学教育、Moodle、软件工具、VPL

I. 介绍

电子学习利用计算机软件支持学生在学习过程中理解新概念 [1]。它为教师提供了新的资源，可以利用 LMS [2] 中的不同工具来他们的课程。此外，它可以在何时、何地以及如何分配教育资源方面提供很大的灵活性 [3]。如今，许多 LMS 将其源代码用于在各种显示的开源许可证下进行修改和定制以满足用户的需求 [4]。在本研究论文中，将使用 Moodle，因为它是开源电子学习软件，并在阿曼苏丹国的苏丹卡布斯大学 (SQU) 中使用。

许多研究人员正试图在教育领域，特别是在计算机科学教育领域普遍优化和扩展 LMS 的功能。CSE 需要额外努力为电子学习环境中的学生准备实践课程。因此，LMS 中应提供所需的软件工具，以用作传统方法的支持或补充。这个领域需要更多的图形应用工具，可以通过共享知识和资源来鼓励学习。为此，可以在 LMS

系统中构建和插入所需的软件工具，作为新的扩展组件 [5, 6, 7, 8]。

本文根据该领域的一些文献解释了如何改进和扩展 LMS（例如 Moodle）的功能以支持 CSE。此外，它还提供了一种为 CSE 开发软件工具并将其集成到 Moodle 中的方法。本文还提供了一种解决方案，以克服在线 CSE 实践课程所需的软件工具的局限性。第二部分，介绍了将工具集成到 LMS 中的一些相关工作。第三节介绍了学习管理系统 (LMS) 的一般概述和 Moodle 作为特例。第四部分，显示了计算机科学教育中软件工具的要求以及 LMS 中这些工具的局限性。第五部分提出了针对 LMS 中 CSE 实用软件工具限制问题的建议解决方案。第六节，提供结论和未来的工作。

II. 文献综述

许多大学和教育机构需要改进他们的学习管理系统，以包含来自教育标准的实践并提供灵活的集成环境 [5]。有多种方法可以通过软件工具实施和支持 CSE，并将它们集成到 LMS 中。这个想法是由 Doberkat 及其同事在 2005 年提出的 [1]。他们引入了“电子学习软件工程”的概念，这是支持软件工程教育的新型电子学习概念技术和工具。它通过提出名为“MuSoft”的电子学习门户，用于支持软件工程课程电子学习系统中可用的实用工具。在这个门户中，他们使用了视频技术和动画，并为统一建模语言 (UML) 开发了工具。此外，他们在此门户中集成了其他工具。

2008 年，Corbera 及其同事 [9] 描述了一种新的 Moodle 模块 CTPractical，该模块被开发用于支持基础计算机组织课程的实践内容。该模块在计算机课程中进行了广泛的活动和评估。此外，它还为学生的工作提供反馈，并作为材料存储库。

2009 年，Rößling 和 Vellaramkalayil [10] 提出了将超文本作为算法动画的 Moodle 模块集成到学习材料中的方法。同年，Rößling 和 Kothe [6] 改进了以前的工具以集成到 Moodle 中。这些工具是使用 AJAX 技术准备的，用作活动插件和 Moodle 中的资源工具。此外，Prieto-Blazquez 及其同事 [11] 为计算机工程和软件工程中的编程虚拟实验室提出了一种集成结构。他们讨论了有关在教学和实际工作中使用该实验室的问题。

2010 年，Riabov 和 Higgs [12] 设计了一个实验室，通过使用最先进的免费许可软件工具进行在线虚拟实验室来教授各种计算机科学课程。这些实验室旨在帮助学生探索计算机系统分析和设计、编程、网络、模拟和建模、图像处理、多媒体、网络开发和数据库等现代复杂技术。所有在线课程都使用基于实验室的项目来支持学生的学习。同年，Bochicchio 和 Longo [13] 提出了 Moodle 和 MicroNet 之间的

集成方法。MicroNet 是一个支持网络的协作电子可以由 2 到 20 名用户组成的小组远程操作的显微镜。他们通过重建和重新设计 MicroNet 作为新插件添加了它，以适应 Moodle 的风格和技术。

2011 年，Huan 等 [3] 提出了 Programming Compiler 等在线 CS 课程的教学实践。关于此编译器的实时视频讨论了如何编写、编译和更正 C++ 代码。Huan 和同事提到了常用的课程内容交付格式，例如 Microsoft PowerPoint (.ppt)、Microsoft Word (.docx) 和 Adobe Portable File Format (.pdf)。这些研究人员还介绍了一些课程工具，其中包括列出 Blackboard 中所有可用资源的内置菜单。此外，他们还推出了一个很好的课件，比如 myitlab (<http://www.myitlab.com>)。同年，Fest [14] 提出将 Java 小程序开发的动态几何软件 (DGS) 的体系结构集成，然后作为组件集成到交互式学习活动中。

2012 年，Lavrishcheva 及其同事 [7] 研究了如何用软件工具支持软件工程。他们开发了一个网站，其中包含不同类型的工具，例如编程语言编译器、集成开发环境 (IDE)，例如 (Eclipse、MS.Net、.. 等) 等。他们表示，该网站可以支持软件工程的学习过程，并使工具可供公众使用。

2013 年，Palumbo 和 Verga [5] 提出了将 Moodle 与 Office 套件集成的集成方法，以提供 OpenOffice.org 多功能和灵活的功能，例如：通过电子表格或图形应用程序的计算、绘图和图表。他们选择 Moodle 和 OpenOffice.org 来集成它们，因为它们的代码风格简洁有效。他们使用 Java 和 AJAX 和 XML 来创建一个嵌入在新 Moodle 模块网页中的 Java 小程序。通过使用这些工具，学生可以将他们的工作保存在 Moodle 服务器上的一个或多个临时文件中。此外，他们可以从本地系统导出或导入他们的工作，编辑和查看他们的文件并提交他们的作业。

此外，Ramos 及其同事 [8] 提出了基于 Corbera 及其同事 [9] 所做工作的 Matlab 评估架构。他们解释了如何扩展 Moodle 的功能以方便评估 Matlab 的编程语言。他们从 CTPractical [9] 引入了一个新的扩展。他们在 Moodle 中重建了一个新模块作为其组件，该模块使用活动模块和控制块来支持新模块功能。基本上，学生可以通过 (.m) 或 (.zip) 文件提交他们的工作，并在测试人员对其进行评估后从他们的实验室工作中找到信息。因此，自动验证过程可以提供关于规范、截止日期、反馈和分配结果的信息。

III. 学习管理系统概述 (LMS)

学习管理系统 (LMS) 是管理虚拟学习环境 (VLE) 的软件系统，供学习者和教师使用以支持在线教育。他们提供了一套工具来支持在线课程的创建、维护和交付

[4]。许多 LMS 通常被实现为遵循客户端/服务器方法 [15] 的 Web 应用程序。LMS 主要有两大类：1) 开源项目，例如 Moodle、Atutor、Sakai 等；2) 商业或财产解决方案平台，例如：Blackboard、WebCT、Desire2learn、SharePoint 等。开源 LMS 通常是建立在可扩展的框架上，允许开发人员调整和修改 LMS 以满足他们的特定需求。

A. MOODLE

Moodle 作为开源软件，是目前最成功的 LMS。它是一个学习管理系统，旨在建立模块化面向对象的动态学习环境。它是通用公共许可证 (GNU) 下的开源系统。开源的特性将方便其根据客户的需求进行开发和维护。此外，这提供了一个研究领域来扩展 CSE [16] 的软件工具。

Moodle 是一个模块化系统，它将内容组织为参考课程的单元和包含课程材料的活动和资源的部分 [5]。Moodle 中有数百个通用插件，它们是由许多开发人员开发的，用于改进 Moodle 的功能。这些工具包括：论坛、聊天、测验工具、作业工具、维基等等。尽管如此，Moodle 的管理员可以安装各种附加插件来扩展 Moodle 安装的活动集。在技术方面，Moodle 是用 PHP 开发的：超文本预处理器编程语言，与大多数 LMS 一样，它使用常见的三层客户端/服务器架构 [16]。

IV. 计算机科学教育中的软件工具

CSE 与一系列与计算和技术知识及其应用教育有关的科学学科有关。计算机科学教育包括以下两个主要部分的课程教学：1) 理论部分：涉及课程所需的理论内容和知识。为了组织理论内容，许多 LMS 提供了一组软件工具，例如：

办公套件（例如演示文稿、电子书、工作表、PDF 文档、Word 等），
多媒体（视频、音频）、
维基和其他。

2) 实践部分：关注学习者通过使用计算机应用程序在实验室的实践课程中获得的知识和实践。许多计算机科学课程需要不同的软件工具，这些工具可以分为以下几类，以便将它们与合适的电子学习系统配置相集成 [17]：独立的软件工具，例如模拟器，组装工等

大型软件工具，例如 Ada L 语言系统，Mathematica、SPSS、编程语言编译器、工程包和电子图书馆。

特殊软件工具，例如虚拟实验室，科学演示、多媒体软件包（如 Matlab 和 Photoshop）、电影、UML CASE 工具 在不同系统中工作的软件工具环境，例如 UNIX 软件工具，Windows 中的 MS-Office。电子学习应用程序中这些软件工具的

有限可用性导致电子学习与传统学习的巨大差距。这些限制可以通过开发和制作这些工具作为 LMS 的一部分来解决，因此学生可以随时随地使用它们。该解决方案可以提供一种使电子学习更加强大和有用的方法，尤其是在计算机科学教育中。

V. 电子学习应用程序中扩展软件工具的建议解决方案

为了克服有关实践课程的这些限制，可以通过将 Moodle 与所需的软件工具集成来增强它。集成可以通过将软件工具构建为 Moodle 中的一个组件来完成。据 Rößling 和 Kothe [6] 指出，电子学习环境中缺少 CSE 需要的功能，包括结合编程练习，这些练习允许学生练习他们的编程知识。图 1 展示了在 Moodle LMS 中开发和集成软件工具作为插件所需的以下阶段。

第一阶段：应该观察和分析现有的 LMS（例如 Moodle），以探索其所有功能和架构。

第二阶段：提出适合 Moodle 系统的扩展架构，并探索和/或开发所需的软件工具。

第三阶段：Moodle 系统应重新设计和定制，以准备接受在其上构建新件。

第四阶段：所需的软件工具应该是使用与 Moodle 系统相同的编程语言和编码风格实现，以方便新组件和系统核心以及系统不同组件之间的通信。

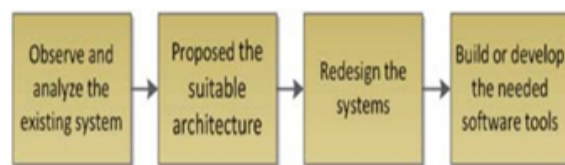


图 1 插件软件工具的开发阶段

A. 插件集成方法的设计与分析图 2 显示了插件工具集成方法。该图显示了用户与 Moodle 组件之间集成的整体架构。对于这种集成，作者建议使用以下三种方法将所需的软件工具插入 Moodle：

1) 将新插件工具作为新功能添加到现有插件工具中，例如：课程和活动插件。因此，这些工具将在活动资源中作为单独的工具提供，教师可以将它们添加到课程材料中。

2) 开发新的插件并创建与应用程序核心和系统中的其他组件进行通信的 API。因此，它将作为新模块出现在 Moodle 中。

3) 在系统核心中构建一个完整的模块，以便在软件工具和一般课程材料中采用更多功能。根据 [18]，在电子学习平台中使用的工具有多种集成方法。这种集成的一种方法是通过包含应用程序编程接口 (API) 来根据以构建电子学习平台的相

同语言编写的代码来实现电子学习平台的功能。图 2 显示了这种方法。

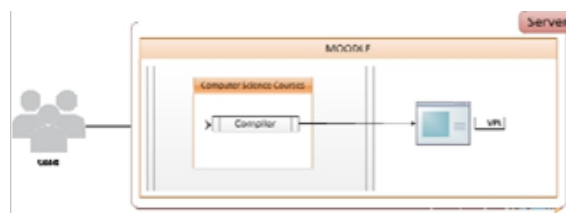


图 2 插件工具集成方法

B. 插件工具实验本文建议在 Moodle 中实现所需的工具作为新组件。不同的研究人员指出了这种方法的重要性 [1, 5, 6, 7]。要为 Moodle 准备 CSE 所需的工具 (例如编译器), 开发人员必须使用与 Moodle 相同的编程语言和技术 (即 PHP)。在这种情况下, 开发人员可以开发一个编译器并将其作为一个活动插件, 他们可以在需要时将其添加到 CS 课程中。图 3 展示了课程开发人员如何准备新工具并将其添加到 Moodle 环境中。它还展示了用户 (即学生) 如何在相同的 Moodle 环境中访问和使用该工具。由于 Moodle 是开源 LMS, 因此 Moodle 中有数百个开发人员创建的插件。例如, Virtual Programming Lab (VPL) [19] 是与计算机科学教育相关的插件之一。它是一个编译器, 允许学生编译不同的编程语言并通过它提交作业。这个插件软件工具是活动模块的一部分, 它是用与 Moodle 相同的编程语言构建的。

开发所需工具的方法是使用 Moodle 的编码风格来开发新插件或子插件。首先, 使用 PHP 编程语言开发具有所需功能的新模块, 并测试新模块是否运行良好。此开发需要访问和自定义一组文件, 这些文件通过其 API 与 Moodle 核心系统进行通信, 以提供所需的功能。首先, 在安装或升级文件 `version.php` 中, 应修改系统版本以检测系统的新版本。其次, 应该引入新模块并将其添加到系统的 `admin_tree` 和导航列表中, 以便课程设计者可以将新的软件工具添加到他们的课程中。然后, 必须为新模块标识事件处理程序, 以便在文件 `event.php` 中的核心和插件以及插件本身之间进行通信。此外, 如果需要在数据库表上存储信息, 则可以编辑数据库表。但是, 不建议在不更改 `install.xml` 和 `install.php` 文件的情况下手动进行修改。所有这些定制工作可能因 Moodle 版本而异。在准备好新模块后以及在 Moodle 安装或升级期间, 新模块将在插件列表中处于活动状态。因此, 课程设计者可以将其添加到他们的课程中, 用户可以轻松访问其功能, 如图 3 和图 4 所示。

C.

VI. 总结和未来的工作

该研究旨在扩展电子学习平台工具开发背后的主要思想, 以便在学术生活中发挥作用。这项研究工作的贡献是研究为 CSE 开发的工具的相关工作。另一个贡

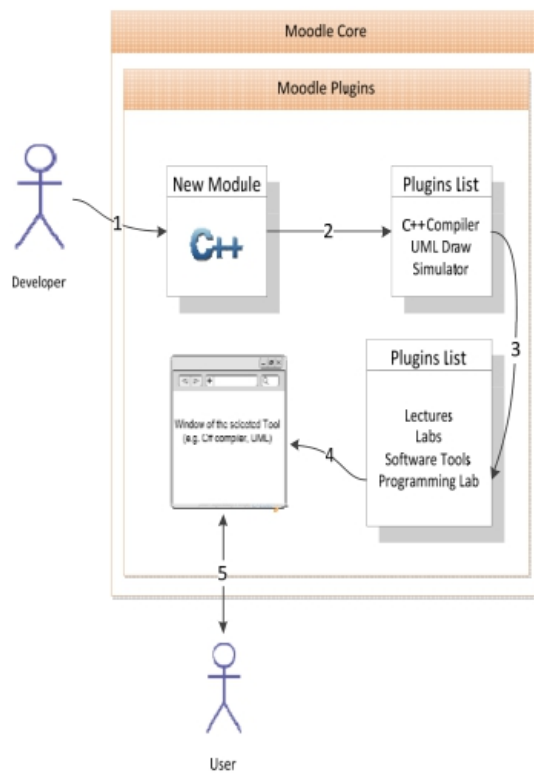


图 3 新模块开发流程

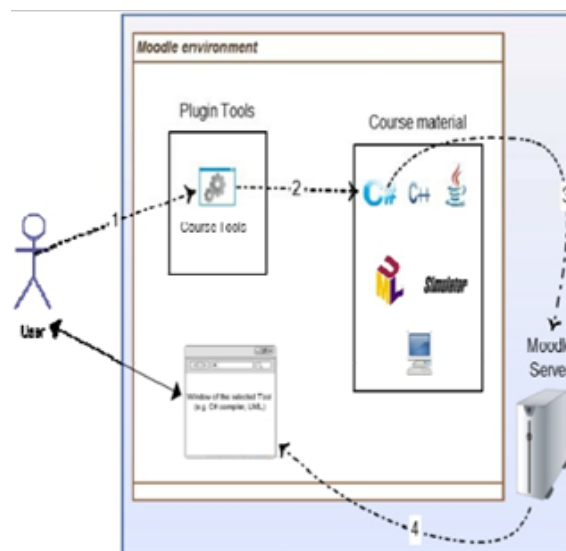


图 4 用户与新组件的交互

献是探索 LMS 的功能, 以支持对计算机科学领域有帮助的软件工具。本文表明, 许多学习管理系统在计算机科学软件工具的可用性方面存在局限性。这些软件工具可以帮助 CSE 的学习者通过实践课程获得知识。有许多 LMS 被归类为开源, 使开发人员能够根据用户的需求对其进行定制。目前作者更喜欢使用 Moodle 有两个原因: 1) Moodle 的属性是开源的, 2) Moodle 被苏丹卡布斯大学使用。在文献中, 一些研究介绍了扩展 CSE 软件工具的一些尝试。在这个方向上, 本文建议在 Moodle 中构建和集成所需的软件工具作为一个新组件。

本文提出的工作有助于学生随时随地在线访问 CSE 的软件工具 [5]。这将有助于提高学生对 CSE 不同实践概念的了解。但是, 提议的解决方案有几个限制, 包括:

插入软件工具可能会在 Moodle 上产生瓶颈, 因为学生将保持登录状态, 在与 Moodle 相同的环境中进行工作。

新添加的组件会消耗 Moodle 的资源, 需要巨大的容量和性能设施 [3]。

新添加的组件可能会降低 Moodle 的性能, 因为它会被运行软件工具的学生使用。

当他们将工具集成到 Moodle 中时, 这种方法可能会增加开发工作量。因此, 它需要被重新设计和重新编码所需的软件工具。本文的未来工作是寻找替代技术或方法来准备软件工具并将它们链接到 Moodle。解决方案之一可能是使用面向服务的方法, 通过该方法, 学生可以使用代表所需软件工具的服务。该解决方案涉及根据所需的服务 (即软件工具) 将学生从 Moodle 虚拟发送到 WWW 中的指定位置。

致谢

作者要感谢苏丹卡布斯大学、理学院和计算机科学系。这项工作部分得到了苏丹卡布斯大学的内部资助。

参考文献

- [1] E.-E., Doberkat, G., Engels, J. H., Hausmann, M., Lohmann, J., Pleumann & J., Schröder (2005). Software Engineering and eLearning: The MuSoft Project. eLeed, Vol. 2005. [online] <<http://eLeed.campussource.de/archive/2/201>> [Retrieved on: 10/2/2014].
- [2] K., Palanivel, & S., Kuppuswami (2011). “Service- Oriented Reference Architecture for Personalized E-learning Systems (SORAPES)” . International Journal of Computer Applications, Volume 24, No.5, pp. 35-44. [3] X., Huan, R., Shehane & A., Ali (2011). “Teaching computer science courses in distance learning” . Journal of Instructional Ped-

agogies, Vol. 6, pp.47-60.

[4] D., Dagger, A., O'Connor, S., Lawless, E., Walsh, & Wade, V. P. (2007). "Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services" . Internet Computing, IEEE, Vol. 11, No. 3, pp. 28-35.

[5] M., Palumbo & F., Verga (2013). "Creation of an integrated environment to supply e-learning platforms with Office Automation features" . Interactive learning Environment. Taylor & Francis LTD, pp. 12, 2013, ISSN: 1049-4820.

[6] G., Rößling & A., Kothe (2009). "Extending Moodle to Better Support Computing Education" . Proceeding of the 14th annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE' 09), pp. 146-150, ACM New York, NY, USA.

[7] E., Lavrishcheva, A., Ostrovski & I., Radetshiy (2012). "Approach to E-Learning Fundamental Aspects of software Engineering" . Institute of Software Systems NAS of Ukraine, ICTERI '12, pp.1-10.

[8] J., Ramos, M. A., Trenas, E., Gutiérrez & S., Romero (2013). "E-assessment of Matlab assignments in Moodle: Application to an introductory programming course for engineers" . Computer Applications in Engineering Education, Vol. 21, No. 4, pp. 728-736.

[9] F., Corbera, E., Gutiérrez, J., Ramos, S., Romero & M. A., Trenas (2008). "Development of a new MOODLE module for a basic course on computer architecture" . ACM SIGCSE Bulletin - ITiCSE '08 Homepage Vol. 40, No. 3, pp. 349-349, ACM New York, NY, USA.

[10] G., Rößling & T., Vellaramkalayil (2009). "First steps towards a visualization-based computer science hypertextbook as a Moodle module" . Electronic Notes in Theoretical Computer Science, Vol. 224, pp. 47-56.

[11] J., Prieto-Blazquez, J., Herrera-Joancomartí & A.-E., Guerrero-Roldán (2009). "A Virtual Laboratory Structure for Developing Programming Labs" . International Journal of Emerging Technologies in Learning (ijet) . Vol. 4, No. 1, pp. 47-52.

[12] V. V., Riabov & B. J., Higgs (2010). "Software Tools and Virtual Labs in Online Computer-Science Classes" . Learning Management System Technologies and Software Solutions for Online Teaching: Tools and Applications, edited by Yefim Kats, IGI Global, 2010, pp. 332-350.

- [13] M. A., Bochicchio & A., Longo (2010). Extending LMS with collaborative remote lab features. Paper presented at the Advanced Learning Technologies (ICALT), 2010 IEEE 41st ASEE/IEEE Frontiers in Education Conference GOLC1, October 12 - 15, 2011, Rapid City, SD.
- [14] A., Fest (2011). “Adding intelligent assessment: a Java framework for integrating dynamic mathematical software components into interactive learning activities” . Journal: ZDM, Vol. 43, No. 3, pp. 413-423.
- [15] G., Vossen & P., Westerkamp (2006, 5-7 July 2006). “Towards the Next Generation of E-Learning Standards: SCORM for Service-Oriented Environments” . The 6th International Conference on Advanced Learning Technologies. pp.184-186, July 05-07, 2006.
- [16] C., Alario-Hoyos, M. L., Bote-Lorenzo, E., Gómez- Sánchez, J. I., Asensio-Pérez, (2012). “GLUE!: An Architecture for the Integration of External Tools in Virtual learning Environments” . Computers & Education, Vol. 60, No. 1, pp.122-137.
- [17] Z., Al-Khanjari, N., Kutti, & M., Hatam (2006). “An Extended E-Learning Architecture: Integrating Software Tools Within The E-Learning Portal” , International Arab Journal for Information Technology (IAJIT), Vol. 3, No. 1, January, 2006.
- [18] F., Huertas & A., Navarro (2013). “Integration Mechanisms in e-learning Platforms” . International Journal of Computer Information Systems and Industrial Management Applications. Vol. 5, pp. 714-721.
- [19] J. C., Rodriguez-del-Pino (2014). “New Virtual Programming Lab (VPL) module” , [Online] <http://www.moodlenews.com/2011/3rd-release-update-to-moodles-virtual-programming-lab-vpl-module/> [Retrieved on: 2/3/2014].

ZHIGANG, Sun, et al. Moodle plugins for highly efficient programming courses. 2001.

Moodle Plugins for Highly Efficient Programming Courses

Sun Zhigang, Su Xiaohong School of Computer Science and Technology, Harbin Institute of Technology, China, sun,sxh@hit.edu.cn Zhu Ning, Cheng Yanyu Undergraduate College, Harbin Institute of Technology, China, zhun,chyy@hit.edu.cn

摘要

本文介绍了三个 Moodle 插件, Online Judge Assignment、Moss Plagiarism 和 Github Assignment, 它们是为 CS1、数据结构、算法、编译器和操作系统等涉及编程的课程设计的。Online Judge 扩展了文件分配的高级上传, 以针对预定义的测试用例即时编译和运行提交。根据通过测试用例的数量和权重, 它可以自动对提交的提交进行评分。没有达到最高等级的学生可以一次又一次地改进和上传他们的程序以获得更高的成绩。它支持 40 多种编程语言。Moss 是一个抄袭插件。它比较相同/相关作业中的所有提交, 列出最相似的配对, 并让教师确认谁作弊。它的底层引擎旨在解析源代码, 但该插件扩展了这种能力以支持 pdf、doc、docx 和 odt 文档。Github 作为一个赋值类型的插件, 可以同步来自 github.com 中 git 仓库的提交, 并将它们存储为提交。因此, 教师可以详细跟踪学生项目并直接在 Moodle 中对其进行评分。此外, 对于基于团队的项目, 每个组使用一个共享 repo, 插件可以统计每个组的工作量百分比。实践证明, 这些插件可以显着减少教师的劳动, 提高学生的学习积极性, 让编程课程更有效率。所有这些都是开源和免费的。欢迎您从 <http://git.io/hmdl> 下载它们。

关键词 Moodle、在线评委、Moss、抄袭、Github、团队项目

介绍

Moodle 是一个很棒的通用平台, 具有许多构建在线课程的内置工具。它适用于任何人、任何机构和任何学科的几乎所有类型的课程。但是不同的课程通常有不同的要求。在计算机科学或软件工程系, 大部分课程都与编程有关, 老师们希望 Moodle 能够提供以下功能:

首先, 在线评委提交成绩作业。在线评审系统可以编译和执行提交, 并使用预先构建的数据对其进行测试以判断正确性。这种方法在很多编程竞赛中得到广泛应用, 如 ACM 国际大学生编程竞赛 (ACM-ICPC, 2012) 和 TopCoder (Inc., 2012)。还有许多线上评测网站 (维基百科, 2012 年) 提供了大量的问题集。它们是如此有吸引力, 以至于许多学生主动在它们身上练习编程技巧。自然地, 线上评测应

用于课程并已被证明是成功的 (Kurnia et al., 2001) (Cheang, Kurnia, Lim, & Oon, 2003 年)。

其次, 检测作业中的抄袭行为。我们的大部分课程都有超过 100 名学生。有的教师甚至需要在一门课上指导 400 多名学生。教师不可能手动发现抄袭。Turnitin (2010 年 8 月) 已与 Moodle 很好地集成, 但价格昂贵且不适合源代码。有多种为代码设计的工具 (Hage, Rademaker 和 Vugt, 2010 年), Moss (Aiken, 2012 年) (Schleimer, Wilkerson 和 Aiken, 2003 年) 是最受欢迎的一种。

第三, 跟踪学生项目的进程。有些课程要求学生在 4-16 周。通常设置 1-3 个检查点, 学生进行展示, 教师进行评价。有两个问题很难解决: 1 教师不知道检查点之间发生了什么。学生会有拖延症吗? 2 老师不知道学生讲的分工是否属实。小组中是否有人只做很少甚至什么都不做?

Github.com 使跟踪项目的整个历史成为可能。但是 github 不是为课程管理而设计的。导师、学生、小组和成绩最好在 Moodle 中进行管理, Moodle 应该将学生的活动与 github 同步, 并以更方便的界面呈现给教师。

本文介绍了 Moodle 的 Online Judge Assignment、Moss Plagiarism 和 Github Assignment 插件。我们从 2006 年开始在 C/C++ 编程语言 (CS1) 课程中使用它们, 从 2008 年开始在操作系统课程中使用它们。也有许多来自不同国家的用户。反馈表明, 它们可以为教与学带来极大的便利、公平和效率。

在线评测 (Online Judge)

Moodle Online Judge 包含两个插件:

1 本地插件实现了一个名为 Judgelib 的在线评测库。它包含三个模块: 判断引擎、判断守护进程和库 API。2 分配类型插件提供 UI 并调用 Judgelib 来判断提交。

图 1 是 Moodle Online Judge 的架构。客户端 (我们只实现了一种客户端——在线评测分配类型) 通过 API 向 judgelib 提交评委任务。每个任务都有源代码、测试数据、资源限制以及应该使用哪个评测引擎。任务被写入数据库并标记为 ONLINEJUDGE_STATUS_PENDING。评测系统守护进程始终驻留在内存中, 并定期检查数据库中是否有待处理的任务。当发现一个待处理任务时, 将其标记为 ONLINEJUDGE_STATUS_JUDGING 并发送到指定的判断引擎。判断引擎需要一些时间来判断并返回结果。然后将任务的状态设置为结果。最后触发一个 online-judge_task_judged 事件, 让客户端知道结果。评测系统守护进程很棘手。它可以在并行模式下工作。因此同时运行多个判断守护进程是安全的。

作业类型扩展自 Moodle 官方高级上传文件作业类型, 继承了大部分标准功能,

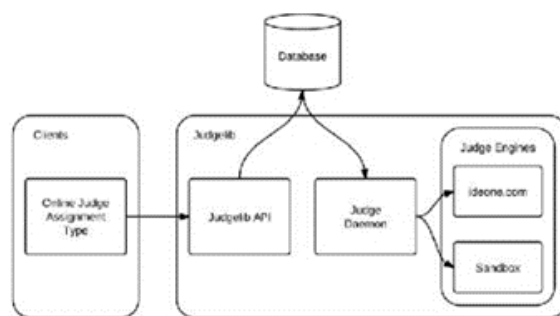


图 5 在线评测架构

如上传文件、限时、手动评分、反馈等。在其设置页面，教师可以选择编程语言，设置内存和 CPU 时间限制。它的设置块可以指导教师重新判断所有提交或配置测试用例。每个作业可能有几个测试用例，包括输入数据、输出数据、等级和错误反馈。提交的材料针对每个案例分别进行评判。如果一个案例通过，则其成绩累加到最终成绩。否则，反馈将作为提示显示给学生（见图 2）。



图 6 提交的在线评测结果

评测引擎被设计为子插件。因此，很容易集成更多的引擎。现在有两个评测引擎：sandbox 和 ideone。Sandbox 使用 libsandbox(Liu, 2012) 作为在 Linux 上执行 C/C++ 提交的受限环境。它可以在运行时监视目标程序的行为。当程序的执行超过特定的时间或内存限制，或者调用了不允许的系统调用时，sandbox 会立即终止它。Ideone 引擎通过调用 ideone.com 的 Web 服务支持 40 多种编程语言（例如 Java、Perl、Python、C#）。它比 sandbox 更安全，但延迟更长。此外，ideone 只允许一个免费帐户每月提交 2000 次。但他们正在考虑为 Moodle Online Judge 的用户提供大量配额。

Moss 查重

Moss 的官方网站 (Aiken, 2012) 说：

Moss（用于衡量软件相似性）是一种用于确定程序相似性的自动系统。迄今为止，Moss 的主要应用是检测编程课程中的抄袭。自 1994 年开发以来，Moss 在这个角色上一直非常有效。moss 背后的算法是对其他作弊检测算法（至少，对我

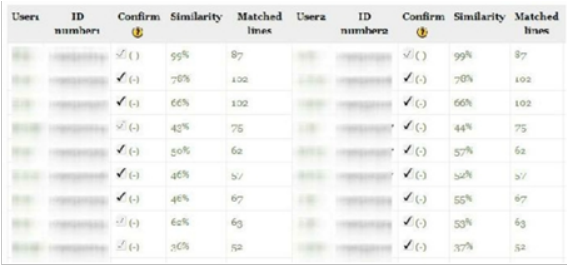
们已知的那些)的重大改进。

它支持数十种程序语言: C、C++、Java、C#、Python、Visual Basic、JavaScript、FORTRAN、ML、Haskell、Lisp、Scheme、Pascal、Modula2、Ada、Perl、TCL、Matlab、VHDL、Verilog、Spice、a8086 汇编、MIPS 汇编和 HCL2。

Moss 作为服务与用 Perl 编写的客户端程序一起工作。该客户端可以 (a) 上传文件; (b) 指定上传文件的编程语言; (c) 指定一组文件作为骨架代码, 不会被视为抄袭; (d) 设置给定段落被忽略之前可能出现的最大次数。成功后, 它将打印一个 URL 并退出。在任意网页浏览器中访问该网址都可以得到抄袭检测结果页面, 该页面并排显示可能的作弊代码并突出显示相似的代码段。

在应用 Moodle 之前, Moss 已经在我们的课程中使用了 2 年。因此, 集成它们的要求很简单。Moss 插件的第一个版本是作为一个块实现的。Moodle 2 引入了 Plagiarism API (“Plagiarism API”, 2012) 之后, 迁移升级到了 Moodle 2。新版本不仅继承了 Moodle Plagiarism API 和 Moss 的所有能力, 还引入了四大显著特性。

第一个功能是确认。Moss 的结果仅供参考。教师必须手动检查结果以确认谁作弊。Moss 插件会在每次提交时添加确认状态。教师可以查找潜在抄袭的排序列表 (图 3), 然后单击查看并排比较。如果他们确定一定有人作弊, 可以按下确认按钮, 相关学生将通过 Moodle 消息系统得到通知。他们还可以轻松地在同一页面中对已确认的学生进行惩罚。当然, 如果他们改变主意, 还有不确认按钮 (例如学生给出合理的解释。在图 3 中, 第一对 99% 相似度的人未确认, 因为他们中的一个偶尔上传了另一个人的文件。他展示了自己的文件, 并且可以解释其中的每一个细节, 以证明它是属于他的。)



Users	ID numbers	Confirm	Similarity	Matched lines	Users	ID numbers	Confirm	Similarity	Matched lines
...	...	✓ (-)	99%	87	✓ (-)	99%	87
...	...	✓ (-)	76%	100	✓ (-)	76%	100
...	...	✓ (-)	66%	100	✓ (-)	66%	100
...	...	✓ (-)	42%	75	✓ (-)	44%	75
...	...	✓ (-)	26%	62	✓ (-)	27%	62
...	...	✓ (-)	46%	57	✓ (-)	52%	57
...	...	✓ (-)	48%	67	✓ (-)	55%	67
...	...	✓ (-)	64%	63	✓ (-)	53%	63
...	...	✓ (-)	26%	52	✓ (-)	37%	52

图 7 Moss 查重插件结果页面

第二个功能是支持 pdf、doc、docx、rtf 和 odt 文档。尽管在主页中从未提及, 但 Moss 的客户端支持纯文本。所以 Moss 查重插件从常见的文档格式中解析出纯文本, 让 Moss 完成剩下的工作。

第三个特点是历史文件。Moss 引擎从不长时间存储文件。老师们通常会在新

学期开始时重新设置课程，以清理所有未注册学生的提交。Moss 插件可以保留这些文件，并将在以后的测量中包含它们。此外，不同课程中具有相同 moss 标签（由教师设置）的作业也可以一起测量。

第四个特点是多配置。由于文件分配的高级上传允许多文件提交，因此该插件能够为不同的文件使用不同的配置。例如，一个作业需要三个提交文件。一个是报告文件，另一个是 Java 源代码文件。教师可以设置 *.doc、*.pdf 和 *.docx 为纯文本，*.java 为 Java 源代码。结果页面可以显示两个配置的结果一起或单独显示。

与 turnitin(“Plagiarism Prevention Turnitin,” 2012) 和 crot (Butakov, 2012) 插件上传文件后立即显示结果不同，Moss 仅在指定时间（默认为作业的截止日期）进行检测，因为 Moss 要求所有文件必须一起提交。另一个弱点是 Moss 不进行网络搜索。仅测量提交给 Moodle 的文件。但这并不是什么大问题，因为一个以上的学生抄袭同一个来源是很常见的。他们无法躲避 Moss。

Github 作业

Github.com 不仅是开发人员的绝佳网站，也是教师和学生的有用工具。一些导师要求他们的学生在 github 上托管毕业项目。以便主管可以观看项目以了解最新更新并评论提交以提供建议。对于主管来说，一切都比以前更容易，更值得信赖。对于学生来说，公共项目可以向他们未来的雇主展示令人印象深刻的展示。因此，github 也应该对课程项目有所帮助。

但是，github 不是为课程设计的。Github 作业类型插件填补了课程和 github 之间的空白。它允许教师在不离开 Moodle 的情况下跟踪项目。特点是：

1 收集学生的 github 用户名和存储库 URL。2 如果作业处于分组模式，则组成员共享同一个存储库。否则，每个人都有他/她自己的存储库。3 同步所有存储库的所有提交历史并缓存日志。4 统计小组成员的工作量百分比。5 按可选排序顺序（名称、上次更新日期）列出所有存储库。6 指向教师和学生可能关心的 github 页面的直接链接。7 继承了作业类型的标准功能，例如评分。图 4 显示了一个名为 taskmanager-for-linux-chengjisihan 的项目。两名小组成员为此做出了贡献。

这个插件有一个简单的架构。除了用户界面，它还提供了一个命令行同步脚本，它调用 git 命令和 Github API (Duplessis, 2012) 来获取数据。Git 工作目录缓存在 Moodle 临时目录中，日志存储在数据库中。CLI 脚本可以在任何时间段通过 crontab 运行。

现在，该插件仅支持公共存储库。Github 有一个教育计划，该计划为教师提供免费帐户，以根据课程需要创建尽可能多的私人存储库。由于可能有些老师或学

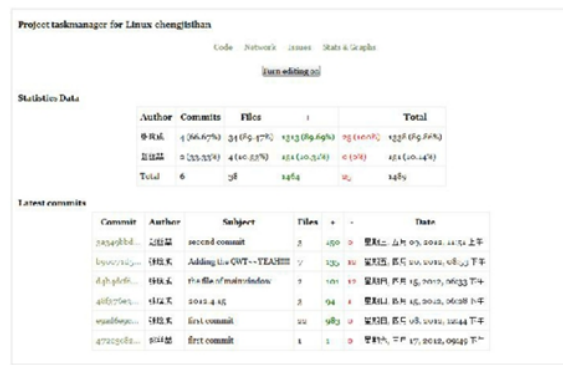


图 8 github 作业插件的项目状态页面

生希望代码是私有的，因此插件将在下一个版本中支持私有存储库。案例分析

Online Judge、Moss Plagiarism 和 Github Assignment 插件为我们的课程做出了重大贡献。它们可以一起使用或单独使用。一些老师表示，他们改变了编程课程的教与学方式。而且，大多数学生也很乐意接受他们。

在线评测

2006 年，Online Judge 插件首次应用于计算机科学和软件工程学生的 C 编程课程。现在它已广泛应用于其他专业和课程以及其他机构和国家。一些用户甚至贡献了波兰语和葡萄牙语翻译。

我校每学期有 2500 多名学生选修 C 程序设计语言课程，授课教师不超过 10 人。有的教师要同时指导 400 多名学生。但学校只支持每位教师 1-4 名助教。在没有 Online Judge 的日子里，作业少了，只能给一部分评分，因为老师负担不起。但是最近几年，一个拿到通过的学生要编写 50 多个程序是很正常的。并且所有的程序都得到了很好的评分。尽管教师必须花时间设计作业和测试用例，但作业可以轻松地在教师之间共享并在将来重复使用。因此，老师们比以往任何时候都更快乐。甚至有人在网上添加了“无限”的在线评测作业，并宣布只有完成最多作业的学生才能获得 100 分等等。

大多数学生也喜欢征服在线评测作业的过程。他们将其视为在线冒险游戏。如果成绩不满意，他们有机会重试。随着他们的代码变得更好，他们获得了更多的技能、经验和更高的成绩。最后，最高等级将利用他们并满足他们的可实现性。在课程论坛中，我们可以轻松找到有关他们享受的帖子。有人甚至需要更多的任务。去年，每位学生平均提交一份作业的时间为 6.97 次。最坚持的学生为一项作业提交了 816 次。

一位教授从 2010 年春季开始使用 Online Judge。在此之前的四个学期，她的课程平均成绩为 75.3、67.2、67.8 和 70.3。在使用 Online Judge 后的接下来的学期中，

平均成绩分别为 89.6、86.6 和 83.7。她说她是在 2010 年春天才降低难度的。之后，难度和数量都增加了很多。数字显示教学质量显着提高。

Moss 抄袭

抄袭是几乎所有课程的大问题。自从纸质作业消失以来，复印一直是零成本。抄袭的来源包括同学、高年级学生和网络。有研究生估计，只有不到 10% 的本科生在毕业前从来没有抄袭过。

在我们大学，Moss Plagiarism 插件主要用于 C/C++ 编程和操作系统课程。所有 C 提交都是纯原始源代码，因此默认设置运行良好。操作系统的作业要求学生修改 Linux 内核并撰写报告。因此，在提交中，有包含 Linux 代码库和学生补丁的测试报告和源代码文件。所以每个分配都有两个配置。(a) *.c 文件被测量为 C 语言并使用 Linux 代码库作为基本文件，因此它们在结果中被忽略。(b) *.pdf、*.doc 和 *.docx 文件以纯文本形式测量。

Moss 给出多达 250 个潜在的抄袭对，无论检测到相似的百分比和多少行。其中只有一部分可以是真正的抄袭。所以确认步骤是最敏感的。教师必须仔细检查结果列表顶部的配对。Moss 的默认排序顺序足够正确，因此通常可以安全地忽略教师认为不是作弊的第一对之后的所有对。据报道，只有极少数学生能够成功挑战确认。

很多同学都说，Moss 给了他们很大的压力。由于只有少数课程使用此插件，因此这些课程的老师在课程的前半场经常会抱怨。然而，中场休息后，当学生觉得自己在压力下确实学到了一些东西时，他们开始热爱和支持老师。2010 年秋季学期，在 OS 课程结束时，有一个调查问“您是否支持在以下课程中使用 Moss Plagiarism?”有 91.7% 的学生选择了“是”。

Github 作业

Github Assignment 插件仍在开发中，于 2012 年 3 月在课程软件设计与开发实践 III 中首次使用。SDDP 是一系列课程。他们要求 200 多名本科生在 16 周内以 2-5 个成员规模的小组为单位设计和开发一个软件项目。一项调查显示，在 SDDP I&II 中，只有 19.27% 的群体可以一直关注项目，而 49.4% 的群体在最后的 1-4 周才开始了他们的工作。此外，许多学生抱怨有些小组成员没有为项目做出贡献，但他们可以得到与其他人相同的成绩。SDDP III 正试图以 Moodle 的方式改变一些东西。

最初的学生和导师（教师或研究生）被添加到 Moodle 组中。然后在分群模式下创建了一个名为“每周报告”的论坛和一个 github 作业。论坛最高分设置为 20，

作业设置为 80。

学生必须每周在论坛上发帖，报告他们在过去一周中做了什么以及他们无法解决的问题。导师应阅读帖子，参与讨论，并根据作品质量对帖子进行评分。因此，学生不断地做某事；导师知道进度和问题很快得到解决。

在 Github 作业中，学生必须在第一周设置 github 存储库和帐户，然后在接下来的几周内提交并推送代码到 github。Git 日志已同步并显示在作业中。导师可以阅读日志，点击 github 链接，享受 github 功能，如内嵌评论和统计图。课程导师关注所有群体。因此，他通常可以查看按最后提交日期排序的存储库列表，以了解学生去了哪里。

在最后一周，学生们进行了答辩和演示。导师在 github 作业中直接聆听，提问并评分。他还检查了 git 日志和统计信息，以了解究竟是谁做了什么。

对于大多数学生来说，这是他们第一次接触 Git 和 Github。除了一些指向 ProGit 和 Github 文档的链接外，没有关于它们的培训。学生必须自学。在本课程结束时，一项调查询问“git 怎么样？”36.14% 的学生选择“它有帮助”，27.71% 的学生选择“麻烦多于解决方案”，36.14% 的学生选择“我不在乎”。所以应该有一些培训来解决下次使用中的麻烦。另一个调查问题是“哪个麻烦最常让你说 F 字？”前两个麻烦是“Cooperation problem”和“Push rejected”。因此，培训应更加关注这些问题。

结论

所有这些插件都以 GPL 许可证发布，可以在 <http://git.io/hmdl> 上免费下载。自第一个版本以来，我们收到了来自世界各地许多教师、管理员和研究人员的许多反馈、建议和补丁。他们和我们的经验证明，这些插件可以使涉及编程的课程更加高效和有效。

Moodle 2.3 更改了分配模块。Online Judge 和 Github 将迁移到它。并且还有很多改进正在计划中。我们希望不断升级它们并使它们免费。

参考文献

ACM-ICPC. (2012). The ACM-ICPC International Collegiate Programming Contest. Retrieved from <http://icpc.baylor.edu/>

Aiken, A. (2012). Moss: A System for Detecting Software Plagiarism. Retrieved from <http://theory.stanford.edu/aiken/moss/>

August, U. (2010). Turnitin Moodle Direct Integration Instructor User Manual. Integration The Vlsi Journal.

iParadigms, LLC. Butakov, S. (2012). Crot Plagiarism Checker. Retrieved from

http://moodle.org/plugins/view.php?plugin=plagiarism_crot

Cheang, B., Kurnia, A., Lim, A., & Oon, W.-C. (2003). On automated grading of programming assignments in an academic institution. *Computers & Education*, 41(2), 121-131. Duplessis, T. (2012). PHP GitHub API. Retrieved May 12, 2012, from <https://github.com/ornicar/php-github-api>

Hage, J., Rademaker, P., & Vugt, N. e V. (2010). A comparison of plagiarism detection tools. *Sciences-New York*. Retrieved December 16, 2010, from <http://www.cs.uu.nl/research/techreps/repo/CS-2010/2010-015.pdf> Inc., T.

(2012). TopCoder. Retrieved July 4, 2012, from <http://www.topcoder.com/> Kurnia, A., Lim, A., & Cheang, B. (2001). Online Judge. *Computers & Education*, 36(4), 299-315. Retrieved from <http://www.sciencedirect.com/science/article/B6VCJ-42SPV9S-2/2/cc5dbbe6b0537f1043d2d08086c6238c>

Liu, Y. (2012). Sandbox Libraries. Retrieved from <http://sourceforge.net/projects/libsandbox/> Plagiarism API. (2012). Retrieved from http://docs.moodle.org/dev/Plagiarism_API Plagiarism Prevention Turnitin. (2012). Retrieved May 14, 2012, from http://docs.moodle.org/22/en/Plagiarism_Prevention_Turnitin

Schleimer, S., Wilkerson, D. S., & Aiken, A. (2003). Winnowing: local algorithms for document fingerprinting. *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (pp. 76-85). San Diego, California: ACM Press.

Wikipedia. (2012). Online judge. Wikipedia. Retrieved July 4, 2012, from http://en.wikipedia.org/wiki/Online_judge

致谢

我们的工作得到了哈尔滨工业大学计算机科学与技术学院和软件学院的资助。Github Assignment 的大部分代码是 Fu Jianyu 写的。Shi Xing 和 Liu Qiqing 为在线裁判做出了巨大贡献。Li Zhijun, Che Wanxiang, Guo Ping, Zhao Wei, Zhang Wei and Li Hanjing 的经验对本文有所贡献。