

武汉理工大学

毕业设计（论文）周记

学 院	计算机科学与技术学院
专 业	软件工程（卓越）
班 级	ZY1501
姓 名	卫亚峰
指导教师	邹承明

2019 年 ____ 月 ____ 日

毕业设计（论文）周记填写要求

毕业设计（论文）是本科教学计划中的重要教学环节，是深化、拓宽、综合所学知识的教学过程，也是影响人才培养质量的突出环节。通过毕业设计（论文）环节，不仅要培养学生的工程设计能力和科学研究能力，而且要培养学生热爱科学、崇尚真理的思想品质，严肃认真、一丝不苟的工作态度，深入实践、不断进取的工作作风、独立思考善于合作的工作能力。

抓好毕业设计（论文）工作的过程管理，是保证毕业设计（论文）质量的必要手段。毕业设计（论文）周记是用来记录学生在毕业设计（论文）过程中每周的工作进度和工作内容、完成的工作任务、工作收获和体会，以便于工作计划落实到每一周，确保设计（论文）任务按期完成，也有利于不断总结工作经验，养成良好的工作习惯和工作作风，提高工作效率和水平。同时，也有利于指导老师掌握学生工作情况，及时给予指导。

学生在毕业设计（论文）阶段，必须每周写工作周记，用简短的文字记录总结一周的工作。周记的内容包括：①工作内容；②完成情况；③存在问题；④指导老师的指导情况。指导老师必须每月审阅检查一次，并写出评语和指导语。凡未按要求写周记的学生，不得参加毕业设计（论文）答辩。评阅学生周记是每位指导老师的工作职责，必须认真填写。

本周记是进行学生毕业设计（论文）过程检查的重要依据，将与设计说明书（论文）、图纸、评分手册，英文翻译等资料一道存档保存，务必请教师及学生认真填写，妥善保存。

武汉理工大学教务

毕业设计（论文）周记

第一周（2.18~2.24）

1. 工作内容

本周工作，主要是考察用户需求，定义系统所提供的功能。

在现行微服务架构下，开发只针对特定某个微服务。一般的开发流程，大致如下：明确需求，开发，自测，提测，上线。其中，提测指将代码提交 QA 部门进行统一测试，而自测指开发者自己进行测试。经考察，现行针对微服务的自测方案大致如下：为了测试特定微服务，需要 mock 实现出所有下游服务 server，并实现一个当前服务的 client 进行调用，手动提供测试数据，并对比返回值与预期值的差异。

现行自测方案有如下问题：

- 1、RD 手动实现下游服务 server 及当前服务 client，增加开发量。
- 2、测试用例由 RD 手工维护，没有统一的解决管理方案，往往造成测试不够充分的后果。

为解决上述问题，本系统所提供的主要功能如下：

- 1、针对当前服务，根据用户提供的 IDL 文件及当前服务的服务端口，生成并动态加载运行当前服务的 client。
- 2、针对下游服务，根据用户提供的 IDL 文件及特定下游服务的服务端口，生成并动态加载运行特定下游服务的 server。
- 3、针对用户的某一种测试模式，帮助用户进行测试用例的维护管理。

2. 完成情况

完成情况良好。

3. 存在问题

需要跟身边同事进行大量沟通，咨询他们的自测解决方案。。

4. 收获及体会

通过本周的工作，对一般针对特定微服务开发的自测解决方案有了一些了解，虽然每个人都不太相同，但大致逻辑都是一致的。通过这些沟通，也使我认识到了实际生产环境中，自测这一环节，存在效率低下的问题，这也是我本次毕业设计所实现系统需要解决的问题。

毕业设计（论文）周记

第二周（2.25~3.3）

1. 工作内容

本周工作，是在理解用户需求的前提下，进行系统的总体的架构设计。

首先处于系统需尽可能用户友好的考虑，摒弃了初期将系统做成一个命令行工具的打算，而将系统基于 CS 架构进行设计。

前端架构较为简单，使用了 HTML，CSS 技术进行页面绘制与渲染。针对用户交互逻辑，实现技术大致有如下选择：原生 JS，jquery，vuejs，react 等。这些技术之前接触的不多，大多不甚了解。后经考察，考虑了学习成本及开发效率等因素，选择了 vuejs 作为实现用户交互的技术，而使用较为成熟的 axios 作为 http 库。

关于后端架构，前期设想后端服务作为一个模块，使用较为传统的 MVC 架构。后摒弃这种方案，主要原因有二：1、C++没有合适的 HTTP 库，整个项目用 C++做开发成本较高。2、整个项目逻辑较为复杂，集成到一起，逻辑层次不够明确。

考虑到上述问题，将整个后端系统分为两个模块：1、cgi 模块，使用 beego 框架作为 HTTP 接入层，主要负责处理 HTTP 请求，用户管理，测试用例管理，服务元信息管理。2、kernel 模块，系统内核模块，主要负责 server、client 的生成及动态加载运行工作。两个模块通过 thrift rpc 进行通信。

2. 完成情况

完成情况良好。

3. 存在问题

后端架构前期考虑不周。

4. 收获及体会

经过本周工作，明确了总体的系统架构，并经考察明确了所要使用的一些技术，未后续的学习指定了道路，对后续的研究及开发工作，也有了较为清晰的思路。

毕业设计（论文）周记

第三周（3.4~3.10）

1. 工作内容

本周工作，主要是进行了系统数据库的设计。

在设计一个系统时，在前期进行清晰明确的数据库设计是极为重要的一环。设计数据库时，需要对系统所提供的功能有清晰的认知，并基于这种认知与系统的大致实现逻辑，抽象出系统中的一些公共的抽象概念。然后明确哪些东西是需要持久化的，哪些东西是不需要持久化的。在进行数据库设计时，抽象出了下列需要持久化的概念：

- 1、User，即使用该系统用来对某个微服务进行测试的 RD
- 2、Service,此处不分服务器还是客户端，一个 IDL 即对应了一个 Service，是 thrift 微服务的抽象概念。
- 3、ServicePattern, 服务模式，针对与某次测试，实际上具有一个服务（当前服务）与多个服务（下游服务）的映射关系，这种服务的映射关系，称为服务模式。
- 4、Interface, 接口，每个服务均具有一个或多个接口。
- 5、InterfacePattern, 接口模式，针对特定的服务模式，在发起测试时，实际上，会触发当前服务的一个接口，并调用下游多个服务的一个或多个接口，这种一个接口与多个借口的映射关系，称为借口模式。
- 6、TestCase，测试用例是针对某个接口模式的，测试用例提供上游接口的请求数据及下游接口的响应数据。

2. 完成情况

完成情况良好。

3. 存在问题

按照系统功能，抽象出一些核心概念是一个挺艰难的问题。

4. 收获及体会

本周通过数据库的设计，理解和认识到了系统中的一些核心概念。一方面对系统所实现的功能有了更加深刻的理解，另一方面对系统的实现思路有了更加清晰的认知。

毕业设计（论文）周记

第四周（3.11~3.17）

1. 工作内容

本周的主要工作，是学习 thrift 协议本身，并研究如何动态生成 thrift server 及 client 和加载运行。

Thrift 工具本身分为两部分，一部分为库，一部分为编译器。编译器将 IDL 文件生成一些桩代码，与库文件链接到一起，才能提供服务，以进行 RPC 调用。经我对 thrift 本身的研究，根据用户提供的 IDL 文件生成和动态加载运行 server 或 client 的解决方案大致流程暂如下所述：

- 1、调用脚本，使用 thrift 编译器，动态生成代码。
- 2、针对生成代码，进行文本处理，添加生成客户端或服务端逻辑，并针对客户端请求或服务端响应添加解析制定 json 文件的逻辑。
- 3、调用脚本，利用 g++ 编译器，将生成代码编译成特定动态链接库。
- 4、在程序中动态链接指定的库。
- 5、启动服务器，或触发客户端的请求接口。

2. 完成情况

完成情况良好。

3. 存在问题

根据 IDL 文件获取服务器或客户端的逻辑较为复杂。

4. 收获及体会

通过本周的工作，让我对系统中最核心的模块，kernel 模块的实现逻辑有了更加清晰的实现逻辑，这部分逻辑也是最复杂的，同时对我来讲，技术上，也是最具有挑战的部分。

指导老师意见

(主要记载学生表现情况、毕业设计(论文)进展、任务完成情况、存在问题)

毕业设计（论文）周记

第五周（3.18~3.24）

1. 工作内容

本周工作，主要进行了 kernel 模块的接口设计工作，及 kernel 模块部分的开发工作。

cgi 模块与 kernel 模块之间，使用 thrift 进行 rpc,若要实现 kernel 模块，必须首先定义两模块之间的 rpc 接口。经过我的研究学习，kernel 模块暴露出的如接口供 cgi 模块调用：

- 1、GetServiceList, 获取服务列表，可以获取正在运行的客户端或服务列表。
- 2、AddService, 添加服务，可以为服务器或者客户端
- 3、FillData, 为某个服务的某个接口的请求或响应填充参数，即写入相应的文件。
- 4、RequestTrigger, 请求触发，出发某个客户端的某个请求。

在明确定义了上述接口后，本周还进行了部分的 kernel 模块的开发工作。主要是搭建了 kernel 模块的总体框架，添加了常规的读取配置文件逻辑，日志打印逻辑，启动 thrift server 逻辑等。对于 kernel 服务的 handler 暂没有开发。

除此之外，为了本系统开发测试方便，我也写了个 kernel-client 模块，可以直接调用 kernel 模块对外暴露的接口。

2. 完成情况

完成情况良好

3. 存在问题

本周工作难点主要是接口定义比较麻烦。

4. 收获及体会

通过本周的工作，搭建了 kernel 模块的实现框架，并留出 kernel 服务的 handler 函数供后续开发。也实现了一个 kernel-client 工具，使得后续开发测试都会比较方便。

毕业设计（论文）周记

第六周（3.25~3.31）

1. 工作内容

本周工作，主要是完成了 kernel 模块的开发，核心是 kernel 服务 handler 的实现，更具体的讲，是实现了 kernel 模块的四个暴露给 cgi 模块的接口。

kernel 模块本身会维护两个 vector，一个代表已经运行的 server，一个代表已经生成的 client。因此，GetServiceList 是一个较为容易实现的接口。

当用户触发某个客户端的请求时，client 接口中的请求与下游服务 server 接口的响应会分别解析对应的 json 文件，以获取对应的请求或响应。对于的 json 文件路径是 ``pwd`/${client}/${service_name}/${interface_name}.json` 与 ``pwd`/${server}/${service_name}/${interface_name}.json`。因此，FillData 接口，只需要把其接收到的 json 数据写入相应规则下的文件中就可以了。

当触发一个请求时，我们仅需要找到对应 client 的特定接口，触发即可。

在本周的开发工作中，AddServer 是最有复杂的工作，尤其是当添加一个服务器时。这个接口实现逻辑的大致流程如下：1、使用脚本调用 thrift 编译器生成代码；2、文本处理；3、使用脚本调用 g++编译器变异成动态链接库；4、动态加载相应的动态链接库进当前进程。

2. 完成情况

尚有 20% 左右没有完成。

3. 存在问题

AddServer 接口逻辑十分复杂，有一定的技术难度，在实现时，遇到了很多问题。

4. 收获及体会

通过本周的工作，大致完成了 kernel 模块的开发工作。在开发过程中，遇到了很多问题。有些是技术问题，有些事系统逻辑的问题。在解决这些问题的时候，自己也学到了很多的东西。例如对 C++有更好的了解，尤其是对动态链接库与动态链接机制有了一些理解，对 thrift 协议的实现原理也有了比以前更加深刻的理解。

毕业设计（论文）周记

第七周（4.1~4.7）

1. 工作内容

本周工作主要是学习了 beego 框架的使用。

本系统的 cgi 模块，即接入层，主要提供的功能是处理 HTTP 请求，用户管理，测试用例管理，服务元信息管理等。为了提高开发效率，我使用了 go 语言，并使用了 beego 作为 HTTP 框架。

beego 是一个快速开发 Go 应用的 HTTP 框架，它可以用来快速开发 API、Web 及后端服务等各种应用，是一个 RESTful 的框架。它是基于八大独立的模块构建的，是一个高度解耦的框架。

beego 本身，是基于 MVC 架构的。当接受到一个 HTTP 请求时，基本的处理逻辑如下：controllers 层处理业务逻辑，调用 models 层进行持久化操作，在调用 views 层进行页面展示，这是一个标准的 MVC 架构设计。beego 亦具有完善的 ORM 库。

beego 框架具有一些很通用和独立的模块，使用起来很方便，类似于：config 模块，用于读取配置文件；logs 模块，用于打印日志；session 模块，用户维护管理用户 session 等等，都是一些很实用的模块。

beego 作为一个完善的基于 Go 的 HTTP 框架，内容是很丰富的，本周只是基于基本的 HTTP 处理流程与各个模块所提供的功能对它进行了简单的了解。并非不愿深入研究，而是等到后续使用再做深入学习也不迟，效果和效率都会更好。

2. 完成情况

完成情况良好。

3. 存在问题

无。

4. 收获及体会

本周主要学习了 beego 框架的基本使用方法。beego 是一个基于 Go 语言的 HTTP 框架。虽然之前使用过一段时间的 Go，但时间有些久远了，对于这门语言有些生疏了。也是借此机会再好好学习一下 Go 这门语言。beego 本身是一门比较完善且对程序员较为友好的 HTTP 框架，了解一些它的使用规则之后，使用起来并没有多么大的难度。

毕业设计（论文）周记

第八周（4.8 ~ 4.14）

1. 工作内容

本周主要学习了一些前端相关的技术，主要是 Bootstrap 技术与 Vuejs 库。

由于我之前对前端相关的技术不是很了解，所以特定化了一些时间学习前端相关的技术。就我本周的学习，对这些技术有了一些简单的了解。大致上讲，Bootstrap 可以理解为一个样式表库，针对 HTML 的各种组建，利用 CSS 实现了各种样式，使得不用 RD 手撸 CSS 代码，就可以使 HTML 代码变得好看起来。而 VueJS 是一个基于 JS 的库。其主要的功能是为静态页面添加动态效果。

Bootstrap 是一组用于网站和网络应用程序开发的开源前端（所谓“前端”，指的是展现给最终用户的界面。与之对应的“后端”是在服务器上面运行的代码）框架，包括 HTML、CSS 及 JavaScript 的框架，提供字体排印、窗体、按钮、导航及其他各种组件及 Javascript 扩展，旨在使动态网页和 Web 应用的开发更加容易。简单来讲，Bootstrap 对一系列 HTML 组件的基本样式进行了定义，并且为文本、表格和表单元设计了一套独特的、现代化的样式。

Vuejs 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。Vue.js 的核心是一个允许采用简洁的模板语法来声明式地将数据渲染进 DOM 的系统，这使得页面展示跟逻辑分离。

2. 完成情况

完成情况良好

3. 存在问题

无。

4. 收获及体会

就我本人的技术栈而言，对前端相关的技术一直了解不多，也没有什么机会接触。因此在本周的工作中，还是接触到很多新的技术的，对自己也挺有挑战。但在了解过一些前端技术后，前端本身本身的技术难度我觉得还是不是很大的，可能是由于我只是为了达到目标而没有深究内在原理的缘故。

指 导 老 师 意 见

(主要记载学生表现情况、毕业设计 (论文) 进展、任务完成情况、存在问题)

毕业设计（论文）周记

第九周（4.15 ~ 4.21）

1. 工作内容

本周工作主要是前端页面设计及前端代码编写。

前端代码编写的主要难度在于对于 html, Bootstrap, Vuejs 技术的不熟悉。对于基础的页面展示，无非是利用 html, bootstrap 技术将页面按应该的样子绘制出来，虽然对于相关技术的不熟悉造成了很多问题，但逻辑上并没什么复杂的问题，往往在查阅一些技术文档后，就可以解决了。

对于前端代码，技术上的难点主要在于对于 Vuejs、axios 技术的应用上，即是前端代码的逻辑主要聚集在与后端服务的交互逻辑上。就与后端交互的角度来看，前端交互逻辑主要分为如下几个方向：

- 1、用户管理
- 2、服务管理
- 3、测试用例管理

在系统的前端部分中，除了基础的页面展示，与后端的交互逻辑之外，对于数据的展示也消耗了我很多工作量，像是每个服务的 IDL 文件的展示，用户输入测试用例的展示，测试的结果的展示，都比较麻烦。

2. 完成情况

完成情况良好。

3. 存在问题

在本周的工作中，遇到了很多琐碎的问题。由于我之前接触的不多，对很多基础的东西了解的不够，被类似于浏览器的缓存机制之类的琐碎的问题困扰了很长时间。

4. 收获及体会

本周主要致力于前端代码的开发，我向来对前端相关的技术能避则避，加上之前对前端知识实在知之甚少，因此也造成了现在的窘境。总的来说，本周踩了很多坑，挺痛苦，但当然也因此学习到了很多基础的前端知识，也学习了一些之前不感兴趣的技术，丰富了自己的技术栈。福祸相依，未尝不是一种收获。

毕业设计（论文）周记

第十周（4.22 ~ 4.28）

1. 工作内容

本周的主要工作是利用 beego 框架进行 cgi 模块业务逻辑的开发。

当 cgi 模块处理一个 HTTP 请求时，从流程上划分，分为如下几个步骤：1、接入层，将请求参数封装成一个 request 对象；2、路由层，根据请求参数重的请求路径，将不同的请求分发给不同的控制器进行处理；3、控制层，根据特定请求、特定方法、特定参数处理特定的业务逻辑，可能会利用持久化层进行持久化和反持久化；4、模式层，一方面，封装了持久化层，另一方面，实现了一些跟特定请求没有强绑定关系的逻辑操作。

cgi 模块实现的功能从逻辑上，可以划分为如下几个模块：

- 1、用户管理，进行基础的用户管理
- 2、服务管理，进行 thrift 服务的管理
- 3、测试模式的管理，测试模式指的是某个当前服务的接口与一个或多个下游接口的映射关系。
- 4、测试用例的管理，即测试数据的管理。

2. 完成情况

完成情况良好。

3. 存在问题

代码实现中遇到的主要问题就是逻辑不够清晰，代码的结构略为混乱，并且代码格式不规范，下一阶段在开发的同时可以进行一定程度的重构。并且在后续的开发中需要注意代码逻辑和代码规范的问题。

4. 收获及体会

通过代码的编写，最大的体会就是之前的详细设计工作如果做好的确能够节省编码的时间，提高编码的效率。因为设计好了，代码就有了框架，开发的时候就能够胸有成竹，具有全面联系的整体观，遇到 bug 也能够快速定位并解决。

毕业设计（论文）周记

（主要记载毕业设计(论文)期间的工作内容、完成情况、存在问题、收获及体会）

毕业设计（论文）周记

（主要记载毕业设计(论文)期间的工作内容、完成情况、存在问题、收获及体会）

指 导 老 师 意 见

(主要记载学生表现情况、毕业设计 (论文) 进展、任务完成情况、存在问题)

毕业设计（论文）周记

（主要记载毕业设计(论文)期间的工作内容、完成情况、存在问题、收获及体会）

毕业设计（论文）周记

（主要记载毕业设计(论文)期间的工作内容、完成情况、存在问题、收获及体会）

毕业设计（论文）周记

（主要记载毕业设计(论文)期间的工作内容、完成情况、存在问题、收获及体会）

毕业设计（论文）周记

（主要记载毕业设计(论文)期间的工作内容、完成情况、存在问题、收获及体会）

指 导 老 师 意 见

(主要记载学生表现情况、毕业设计 (论文) 进展、任务完成情况、存在问题)