**毕 业 设 计（论文）**

**题 目：** 基于docker的京东外挂程序的

设计与实现

**学院：** 计算机与通信学院

**专业：** 网络工程 **班级：** 1801 **学号：** 201803120132

**学生姓名：** 李 婷

**导师姓名：** 黄 鲲

**完成日期：** 2022年6月5日

诚 信 声 明

**本人声明：**

**1、本人所呈交的毕业设计（论文）是在老师指导下进行的研究工作及取得的研究成果；**

**2、据查证，除了文中特别加以标注和致谢的地方外，毕业设计（论文）中不包含其他人已经公开发表过的研究成果，也不包含为获得其他教育机构的学位而使用过的材料；**

**3、我承诺，本人提交的毕业设计（论文）中的所有内容均真实、可信。**

**作者签名： 日期： 年 月 日**

**目 录**

[摘要 I](#_Toc28157)

**[Abstract](#_Toc32694)** [II](#_Toc32694)

[第1章 绪 论 1](#_Toc17280)

[1.1 引言 1](#_Toc29937)

[1.2 课题的目的与意义 1](#_Toc29477)

[1.3 课题的发展现状和前景展望 2](#_Toc26145)

[1.4 课题的主要内容和研究方法 3](#_Toc5329)

[1.4.1 主要内容 3](#_Toc16517)

[1.4.2 研究方法 3](#_Toc27150)

[1.5 论文的工作和安排 4](#_Toc11192)

[第2章 相关应用技术与理论基础 5](#_Toc8023)

[2.1 JavaScript技术 5](#_Toc82)

[2.1.1 JavaScript的概述 5](#_Toc14540)

[2.1.2 JavaScript的特性 6](#_Toc4172)

[2.2 Node.js简介 6](#_Toc3440)

[2.2.1 Node.js概述 6](#_Toc14282)

[2.2.1 Node.js的发展优势 8](#_Toc3419)

[2.3 docker简介 8](#_Toc8445)

[2.3.1 docker的概述 8](#_Toc12939)

[2.3.2 docker的优势 9](#_Toc15409)

[2.4 青龙面板简介 10](#_Toc14653)

[第3章 设计分析 11](#_Toc13647)

[3.1 设计目的分析 11](#_Toc25578)

[3.2 设计技术分析 12](#_Toc14930)

[3.3 设计目标分析 12](#_Toc28771)

[3.4 JavaScript编辑软件分析 12](#_Toc17025)

[3.5 流程分析 13](#_Toc20134)

[3.5.1 整体设计流程 13](#_Toc17521)

[3.5.2 外挂程序任务流程 13](#_Toc28497)

[第4章 外挂程序任务分析 14](#_Toc31002)

[4.1 京东种豆得豆任务 14](#_Toc28403)

[4.1.1 任务组成 14](#_Toc31872)

[4.1.2 程序分析 15](#_Toc19582)

[4.1.3 结果分析 16](#_Toc24772)

[4.2 东东萌宠任务 16](#_Toc3237)

[4.2.1 任务组成 16](#_Toc11926)

[4.2.2 程序分析 17](#_Toc30969)

[4.2.3 结果分析 18](#_Toc26205)

[4.3 京豆变动告知 18](#_Toc24819)

[第5章 环境搭建 19](#_Toc28572)

[5.1 Node.js运行环境的使用 19](#_Toc13394)

[5.2 docker的搭建 19](#_Toc18811)

[5.3 控制面板的搭建 20](#_Toc5194)

[5.3.1 搭建面板 20](#_Toc30648)

[5.3.2 拉取外挂程序 22](#_Toc19729)

[5.3.3 安装依赖管理 22](#_Toc11399)

[第6章 时间设置以及任务拉取 22](#_Toc5202)

[总 结 24](#_Toc7263)

[参 考 文 献 25](#_Toc4128)

[致 谢 27](#_Toc29327)

**基于docker的京东外挂程序的设计与实现**

摘要：随着互联网的发展，网上购物让大家足不出户就能买到全国各地的好东西。网上购物越来越普及，与此同时购物APP也越来越多了，如淘宝、京东等。而对于京东APP，它提供了一些游戏或任务，当用户完成这些可获得购物优惠或其他福利，非常吸引用户，但任务做的越多，就越繁琐，越费用户时间。所以针对此情况，设计出京东的外挂程序，它可以一键使用，自动完成任务，使用简单，同样非常节省用户时间。此程序使用docker容器、JavaScript语言、Node.js开发工具等。通过测试结果发现本程序可以使用户轻松的完成京东程序里的任务、游戏等，减轻了用户花费时间的精力，帮助用户更轻松的做任务。

关键词：JavaScript；docker；外挂程序；Node.js

**Design and Implementation of Jingdong plug-in program based on Docker**

Abstract:With the development of the Internet, online shopping allows everyone to buy good things from all over the country without leaving home. Online shopping is becoming more and more popular, and at the same time, there are more and more shopping apps, such as Taobao and JD.com. As for the Jingdong APP, it provides some games or tasks. When users complete these, they can get shopping discounts or other benefits, which are very attractive to users, but the more tasks are done, the more cumbersome and time-consuming the user is. Therefore, in response to this situation, we designed a plug-in program for JD.com, which can be used with one click, automatically completes tasks, is simple to use, and also saves user time. This program uses docker container, JavaScript language, Node.js development tools, etc. Through the test results, it is found that this program can enable users to easily complete tasks, games, etc. in the Jingdong program, reduce the time and energy of users, and help users to do tasks more easily.

Keywords:JavaScript; Docker; Plug in program; Node.js

# 第1章 绪 论

## 1.1 引言

随着时代的发展，社会的进步，网上购物的这一现象越来越普及，深入到了人们的生活。与此同时购物软件也被广泛使用，如京东购物平台，商品种类繁多，十分受大众喜爱。同时，京东出现了很多任务、游戏、关卡等，当用户完成这些任务，得到对应的奖励、优惠券、京豆等等。于是乎，很多用户就会利用自己时间去签到得京豆、完任务、游戏等。任务做完的结果虽然是让人愉悦的，但是完成过程却是繁琐、费时，有时候甚至隔一天没做，任务就需要重新开始，十分头疼。因此设计关于京东的外挂程序，它可以在某个时间自动去完成这些任务，非常方便，节省时间。外挂程序的使用不受时间、地域和设备的限制，拥有一台电脑，即可完成。它使用户在完成任务时拥有一种轻松、方便的使用感。

在国内的发展中，外挂多是用于游戏或者在一些其他方向被使用，但是这些大都是一种不被允许甚至违法的行为。但此次针对京东设计的外挂程序，是一种针对只为用户个人服务的模式行为，为了京东用户在完成游戏、任务方面更加方便而设计的外挂程序。京东用户有时为完成几个任务获得奖励，短时间可能花费十几分钟，长时间有可能花费几小时完成。因此，这个外挂是简化用户没时间或长时间做任务时的一种工具，可以用更短的时间完成任务。外挂根据用户的想法可随时用，也可随时停，十分便利。

对于外挂程序的设计，主要利用JavaScript脚本语言来设计，在Node.js开发环境中被应用的程序。它的出现可以让原本需时刻看着任务完成，变成了只需要一键使用外挂程序就可以自动完成任务，节省做任务时间、因忘记做任务而恼等等情况，是一种便利京东用户个人使用的方式。

## 1.2 课题的目的与意义

网上购物的普及，购物软件的广泛使用。网上购物的方式实现了无时无刻、任何地方购买商品，十分的受大众喜爱。在这一现象普及的情况下，购物软件出现了丰富多彩的样式、展现。而此次课题是针对京东APP，设计出一些外挂程序实现自动完成任务、游戏的方案。对于京东APP，它界面上的应用丰富多彩，包含了小游戏、小任务等，比如领京豆、玩游戏(种豆得豆、东东萌宠等)、浏览店铺和商品等等，总之，任务样式繁多。当这些任务被完成时，有机会兑换商品优惠券、奖励的情况，在付款商品时可减少付款金额。优惠券、奖励等十分吸引使用者，于是乎，很多用户会去浏览、签到、玩游戏等。但是某些时候做这些事又非常费时、麻烦，因此，针对这些缺点，课题设计了一种基于docker的京东外挂程序，此外挂程序的使用可以方便用户在完成京东任务时快速方便，实现更好的任务体验。这个外挂程序是凭用户想法，需要使用就用它，想自己完成就关闭它，用户自己做。它的设计，让原本需要时刻盯着京东任务，变成了只需一键使用外挂程序，即可自动实现需要完成的任务，轻松、省时、便利。当用户不想自己做任务时候，即可使用外挂程序，空出的时又可以去做一些其他事情，相当于一时多用。 此次设计的外挂程序涵盖在领京豆额外奖励、种豆得豆、东东萌宠这三个方面，然后也涉及到做任务时关注的店铺和收藏商品的过多而自动取关的外挂，以及资产变更情况外挂。京东任务非常多，可根据需要在设计需要的外挂程序。

## 1.3 课题的发展现状和前景展望

对于外挂的使用，已经在很多方面被使用，用在最多的地方可能就是游戏外挂，游戏外挂的发展也有十几年的历史了。对于外挂的使用，如果不正当使用，是属于不合法的行为，但是当它被合法合理使用，又是可以非常便利人们的。

外挂这一词最早是出现在2003年的时候，那时候外挂的出现是一些对电脑技术极具热情的年轻人研究开发出来的一种软件，当时他们并不是通过外挂技术来获取利益，而是简单的想要研究与电脑相关的一些语言程序，例如研究出现比较早的基础语言C语言，到后面外挂程序也开始应用到了Java程序语言、Python程序语言、JavaScript程序语言等等。在这些年轻人的研究下，外挂的应用得到越来越多，尤其是在游戏中，外挂马上火爆了起来，从此外挂这一词也就有了由来，得到的关注也越来多。

而此次针对设计京东的外挂程序，是使用户规范合理的运用外挂程序，当有需求的时候为用户个人服务的外挂程序。外挂字面意思即指在电脑运行中，一个程序通过某种事件触发而得以挂接到另外一个程序的空间里，挂接的目的通常是想改变被挂接程序的运行方式，从而达到轻松做完任务的好处。而对于此次课题的研究，是设计基于docker的京东外挂程序，在设计过程中使用了docker这个技术。容器技术发展得如火如荼，不管是大型互联网企业还是初创公司，都在开发部署中应用容器技术来降低成本，促进产品的快速迭代[1]。但是，在容器的发展中以docker为代表的技术是发展的最好的，docker在一定时间内迅速占领了市场。它在开发或者测试服务设置、部署中可以节省人力资源和机器资源。在这一个点上，docker减少了环境部署中的能量损失，同时在一台机器上运行多docker container。在2013年的时候，dotCloud公司宣布将docker进行开源处理，随着越来越多的工程师发现了它的优点，docker的人气迅速得到了攀升，成为当时最火爆的开源技术之一，它的使用也是逐年的在增加。而对于JavaScript脚本语言，它经过多年的发展，也成为现代技术世界中很受欢迎、增长很快的编程语言之一。而Node.js又可以让JavaScript脱离浏览器使用，因此使JavaScript在设计中可以更好的得到应用，更加便利对外挂程序的编写与设计。

京东外挂程序的合理使用，它是可以提升用户的使用感。它可以随用随停，当用户对它有需求的时候可使用，当用户想自己完成京东任务的时候可以不使用外挂，自己完成，它的操作十分简单。此次课题针对京东应用里面的任务，方便大家省时、简便的完成任务。此外挂程序是针对用户个人服务，合理的使用它便利用户，自动做任务节省用户频繁使用手机的时间。

## 1.4 课题的主要内容和研究方法

### 1.4.1 主要内容

（1）编写外挂程序代码。分析京东游戏需要完成的任务，根据任务需求设计相应脚本程序去实现的功能，达到仅使用外挂程序就可以把任务做完的效果。

（2）测试代码。通过测试外挂程序代码，检测代码是否可行，是否达到设计目标的预期显示效果。再通过查看京东APP，查看任务是否显示都以完成。

（3）进行环境搭建。在设计外挂程序的配置过程中，用Xshell连接虚拟机，通过使用命令来进行环境的配置与搭建。

（4）使用方式。为方便用户使用，增强对外挂程序的使用感，将外挂程序统一导入一个控制面板中，实现外挂程序统一安放。面板通过连接用户京东cookie，根据用户自身需要，设置外挂程序开始使用的日期、时间等，到点即可自动完成设置的任务，无需关注，实现自动完成任务的功能。

### 1.4.2 研究方法

（1）文献研究法

在设计此程序之前，根据分许程序需要的技术和方法，充分利用学校图书馆的资源以及网络上的知识、资料，结合现代信息技术的手段，收集和查阅相关的书籍、期刊和网站等。

（2）实验法

通过已有的知识、资源，在尝试和设计中，设计出能够一键自动去完成京东任务的外挂程序。

## 1.5 论文结构

本次的设计是开发一个基于docker的京东外挂程序。通过对外挂程序的设计，在Node.js环境中应用JavaScript程序语言进行外挂编写；使用安全、可靠的docker容器进行一个环境搭建，然后拉取安装控制面板，方便用户更好使用外挂程序。设计的京东外挂程序它可以自动完成京东任务，从而达到提高用户完成任务速度、根据想法轻松使用轻松目的效果。外挂程序涉及的京东任务主要是针对东东萌宠、种豆得豆这两个京东游戏，还有一些京东辅助外挂程序。

本次的毕业论文主要是围绕为京东APP的东东萌宠、种豆得豆任务等设计一个基于docker的外挂程序。设计内容包括以下部分：

第1章是绪论，这一章节是简述基于docker的京东外挂程序的设计与实现这一课题的引言、目的与意义、发展现状和前景展望、主要内容及研究方法。

第2章是相关应用技术和理论基础，这一章节主要是介绍在设计京东外挂程序过程中使用的技术、工具、理论等等。

第3章是论述设计分析，主要是分析课题设计的目的、技术、目标以及所使用的的软件进行分析，然后还有设计流程，通过这些内容来阐述设计实现。

第4章主要是此课题的核心设计部分，主要是叙述分析京东游戏、任务设计符合的外挂程序。通过分析设计的部分过程讲述代码的用途。

第5章是环境搭建，这一章主要介绍课题的实现需要准备的环境，以及为了方便用户的使用涉及到的工具。通过对这些的分析，进一步分析课题的可行性。

第6章是面板设置，这一章是通过对面板的使用进行剖析和说明，同时讲述它该如何使用、如何操作的流程。

最后部分是为本次毕业设计做的总结。

## 1.6 本章小节

本章主要从课题的目的、意义上先分析研究此技术的原因是什么，再分析外挂是怎样产生的，外挂出现的目的是什么等，通过这些再来叙述应用外挂程序的发展前景。由此便展开对京东外挂程序的分析，主要叙述使用外挂程序的应用，以及对京东用户带来的好处和方便。

# 第2章 相关应用技术与理论基础

## 2.1 JavaScript技术

### 2.1.1 JavaScript的概述

JavaScript是基于原型的一种通用的、面向对象的脚本语言，它的目的是提供一个通用的应用程序代码可以嵌入在不同的系统，在不占用大量网络资源的情况下使用[2]。提到JavaScript，我们不要被它的外表迷惑，因为JavaScript和Java一丁点关系都没有。在本次课题设计中，主要通过JavaScript编程语言去实现外挂程序。JavaScript简称JS，它是一种具有函数优先的轻量级，解释型或即时编译型的[编程语言](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80/9845131" \t "https://baike.baidu.com/item/JavaScript/_blank)。JavaScript是一种在Web程序开发设计中应用十分广泛的脚本语言[3]。JavaScript 基于原型编程、多范式的动态脚本语言，并且支持[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1/2262089" \t "https://baike.baidu.com/item/JavaScript/_blank)、命令式、声明式、[函数](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0/301912" \t "https://baike.baidu.com/item/JavaScript/_blank)式编程范式。JavaScript在1995年由[Netscape](https://baike.baidu.com/item/Netscape/2778944" \t "https://baike.baidu.com/item/JavaScript/_blank)公司的Brendan Eich在网景导航者浏览器上首次设计实现而成，命名为LiveScript。后因为Netscape与[Sun](https://baike.baidu.com/item/Sun/69463" \t "https://baike.baidu.com/item/JavaScript/_blank)合作，Netscape管理层希望它的外观看起来是像Java，由于是Netscape公司基于LiveScript在尝试引入了Java的编程设计方式和理念后，又将LiveScript更名为JavaScript。

根据现在的发展来看，JavaScript可以依赖浏览器执行使用，也可以通过Node.js环境来编译。图2.1是JavaScript在浏览器中的结构组成，JavaScript它主要由ECMAScript、文档对象模型、浏览器对象模型组成，ECMAScript实现JavaScript语言语法描述，文档对象模型(DOM)是用于处理网页内容方法的接口，浏览器对象模型(BOM)实现与浏览器的交互[4]。

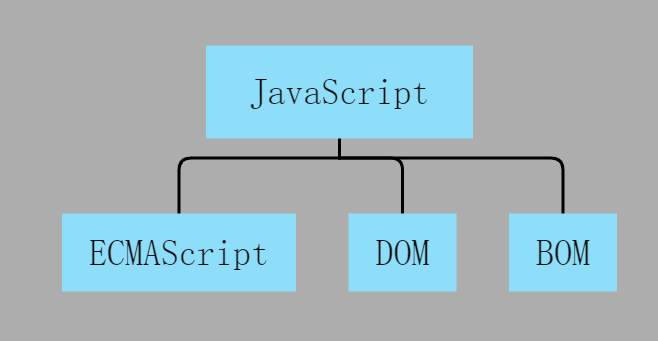


图2.1 JavaScript结构组成

Node.js环境中的JavaScript的组成：ECMAScript核心+全局成员+核心API模块。

（1）全局成员有console、setInterval、setTimeout......

（2）核心API模块：就是Node平台单独提供的一些API，如文件、网络、路径等等，这些API是Node.js平台独有的。

### 2.1.2 JavaScript的特性

通过对JavaScript脚本语言的研究与学习，它具备了多种开发优势的特性，具体如表2.1所示。

**表2.1 JavaScript的特性**

|  |  |
| --- | --- |
| 特性 | 描述 |
| 基于对象 | JavaScript是基于对象的语言，它不仅可以创建对象，也能使用现有的对象，具有一定的灵活性，能够实现一些复杂的功能。 |
| 实时性 | JavaScript事件是可以实现在客户端的处理，页面对处理结果可以作出实时响应。 |
| 动态性 | JavaScript本身包了含很多方法与属性，能够灵活的响应多种文档与页面事件。 |
| 跨平台性 | JavaScript脚本语言不依赖于操作系统，仅需要浏览器的支持。因此一个JavaScript脚本在编写后可以带到任意机器上运行使用。 |
| 脚本语言 | JavaScript是一种解释型的脚本语言，C、C++等语言先[编译](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91" \t "https://baike.baidu.com/item/JavaScript/_blank)后执行，而它是在程序的运行过程中逐行进行解释。 |

JavaScript脚本语言不依赖于操作系统，仅需要浏览器的支持。因此一个JavaScript脚本在编写后可以带到任意机器上使用，前提是机器上的浏览器支持JavaScript脚本语言，它已被大多数的浏览器所支持。但是时至今日，JavaScript的应用领域早已从前端的Web浏览器，也延伸到了后台服务器。所以此次课程设计的JavaScript程序员通过依赖Node.js环境下进行开发、创建的，而后可以脱离浏览器运行JavaScript程序语言，更加方便，操作方式更多。

## 2.2 Node.js简介

### 2.2.1 Node.js概述

上文提到了课程设计使用的程序语言是JavaScript，它是一种最基础的网络脚本语，用于web应用开发等等，通常是被嵌入在HTML中以实现自身的功能。所以当时JavaScript只能在浏览器中运行，但2009年出现的Node.js彻底改变了这一局面，使得JavaScript可以脱离浏览器在Node.js环境中开发、运行，Node.js的功能使JavaScript可以运行运行于服务端！

Node.js是一个新兴的开源技术，能够让Javascript运行在浏览器之外——服务端的开发平台。Node.js主要是将Google V8引擎封装起来，作为服务器运行平台，用来执行JavasScript编写的后端脚本程序[5]。在Node.js出现之前，JavaScript只能运行在浏览器中，作为网页脚本使用，为网页添加一些特效，或者和服务器进行通信。但是有 Node.js以后，JavaScript就可以脱离浏览器，像其它的编程语言一样直接在计算机上使用，想做什么就做什么，不再受浏览器的限制。近年来，尤其是互联网进入到以前后端分离为主导的开发模式后，Node.js在服务端开发领域的重要性也越来越得到重视。

在用户体验方面，Node.js的实质是运行在Chrome V8引擎上的JavaScript[6]。Node.js就像浏览器一样，可以解析执行JavaScript脚本，但它是让JavaScript运行在操作系统环境下。Node.js不是一门新的编程语言，也不是一个JavaScript框架，它是一套JavaScript运行时候的环境，用来支持JavaScript代码的执行[7]。用编程术语来讲，Node.js是一个JavaScript运行时，所谓运行时，就是程序在运行期间需要依赖的一系列组件或者工具；把这些工具和组件打包在一起提供给程序员，程序员就能运行自己编写的代码了。Node.js由Ryan Dahl开发，发布于2009年5月，Ryan Dahl 是一名专注于实观高性晚Web服务器的优化专家，他将Chrome浏览器的V8引擎单独移植出来，在此基础之上，为其上层的JavaScript提供友好的API，供开发人员使用，并且完全开源免费。

Node.js是一个基于Chrome的JavaScript运行时建立的一个平台；是一个事件驱动I/O服务端的JavaScript环境，基于Google的V8引擎，V8引擎执行JavaScript的速度非常快，性能非常好，是一个可以快速构建网络服务及应用的平台，它是用JavaScript语言构建的服务平台。从近几年的发展来看，Node.js是一种热门的服务器端脚本语言，它允许在后端(脱离浏览器环境)中运行JavaScript代码。要实现在后台运行JavaScript代码，代码需要先被解释，然后正确执行。Node.js原理正是如此，使用Google的V8引擎去解释和执行JavaScript代码。在这里Node.js的主要思想：在面向跨分布式设备运行的数据密集型的实时程序时，使用非阻塞、事件驱动的I/O来保证轻量和高效，是一个满足特定需求的环境、平台。

图2.2为Node.js的内部结构图，我们可以看到，自下向上主要可以分为三层：最底层是Node.js依赖的各种库，如V8引擎库、libuv库；中间是各种Binding代码等等，它相当于两层之间的粘合剂；最上层是应用代码，可以使用Node.js的各种API等等。

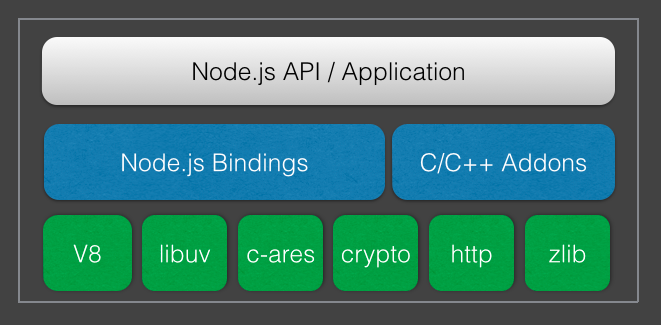


图2.2 Node.js内部结构图

### 2.2.1 Node.js的发展优势

（1）高并发性支持

Node.js采用异步式I/O与事件驱动的架构设计来代替多线程，使用单线程模型，占用服务器资源少，避免了频繁的上下文切换，使得node.js能够处理大量的高并发链接。

（2）跨平台且兼容性高

是一个开源的跨平台，支持Windows，Mac，Linux等操作系统，为用户提供了极大地便利，不再受操作环境的影响。并且，Node.js不运行在浏览器中，所以也就没有JavaScript的浏览器兼容性问题。

（3）实现高性能服务器

严格地说，Node.js是一个用于开发各种web服务器的开发工具。在Node.js服务器中，运行的是高性能V8 JavaScript脚本语言，该语言是一种可以运行在服务器端的脚本语言。

（4）开发周期短

Node.js具有开发周期短的优势，并且开发的成本相对来说也比较低。

1. 学习成本低

Node.js学习起来比较容易，熟练JavaScript程序语言后，很容易上手Node.js的使用。

## 2.3 docker简介

### 2.3.1 docker的概述

docker是基于Go语言实现的轻量级虚拟化技术，为应用提供快速、灵活的部署方式。Docker是一种开源的软件容器平台，开发人员可将其应用程序和依赖项打包到可移植镜像中，这对于应用程序的部署和扩展都很方便[8]。它可以发布到任何可以安装docker引擎的服务器上，如Linux系统、Windows系统、macOS系统，实现了跨平台应用。它使得开发人员比以往更容易将软件封装成“一次构建，随时随地运行”，从而推动了该技术的推广[9]。docker项目是由Solomon Hykes所带领的团队发起，在docker公司的前身dotCloud内部启动孕育，代码托管于Github。在2013年3月，docker正式发布开源版本，并在GitHub中形成一股热潮，在此之后，docker便步入了发展的快车道。2013年11月，REHL6.5正式版发布，它集成了对docker的支持，从此拉开了业界各大厂商竞相支持docker的序幕。

镜像（image）、容器（Container）、镜像仓库（docker Repository）是docker中比较重要的三个基本概念。镜像就相当于模板，是创建docker容器的基础；容器是从镜像创建的应用运行实例，它可以启动、开始、停止、删除，而这些容器又都是彼此相互隔离、互不可见的；镜像仓库则是用来管理镜像。docker是一个可以用于开发，交付和运行应用程序的开放平台。docker能够将应用程序与基础架构分开，从而可以快速，一致的交付的环境平台。借助docker可以与管理应用程序相同的方式来管理基础架构。docker容器是将主机资源分配出一部分，进行隔离，提供一个系统环境。容器更加轻量化，它能够共享操作系统，快速复制出一个相同的运行环境，能够大大减少资源开销和时间开销。docker就是完全使用沙箱机制，相互之间不会有任何接口，可以方便快速地在其它操作系统中部署应用[10]。它几乎没有性能的开销，可以很容易地在机器和数据中心中运行。最重要的是，它不依赖于任何语言、框架包括系统。所以docker在高效的利用系统资源、快速的启动时间、一致的运行环境、持续交付和部署、轻松的迁移、轻松的维护和扩展等方面，这些足以表现出它对课题设计提供的优势。

### 2.3.2 docker的优势

（1）一次构建，多次交付

它类似于集装箱的“一次装箱、多次运输”，docker镜像可以做到“一次构建，多次交付”。当涉及到应用程序多副本部署或者应用程序迁移时，更能体现docker的价值。

（2）一致的运行环境

开发过程中常见的一个问题就有环境一致问题，由于开发环境、测试环境、生产环境的不一致，导致有些bug在开发的过程中并不会被发现，最后使用的时候导致一些错误。而docker的镜像提供了除内核外完整的运行时环境，确保了环境的一致性，从而不会在出现“这段代码在我机器上有问题”这类情况。

（3）更轻松的迁移

由于docker确保了执行环境的一致性，使得应用的迁移就变得更加的容易。docker可以在很多平台上运行，无论是物理机、虚拟机、公有云、私有云，甚至是笔记本，其运行结果都是一致的。因此用户可以很轻易的将一个平台上运行的应用，迁移到另一个平台上，而不用担心可能因为运行环境的变化导致应用无法正常运行的情况。

（4）更好的安全隔离性

docker赋予了应用隔离性，不仅仅局限于彼此隔离，还独立于底层的基础设施。docker默认提供最强的隔离，因此当应用出现问题的时候，也只是单个容器的问题，并不会波及到整台机器。

## 2.4 青龙面板简介

为了方便用户更简单的使用外挂程序，此次设计内容也包含了采用青龙控制面板。它通过上文提到的docker来进行安装、搭建。它可以集中存放外挂程序；它自带多种环境；它可以定时设置任务开启时间，十分的便利。

青龙面板可以将代码统一拉取，存储在一起，实现统一的管理。它是一个定时任务的面板，作用是可以定时脚本开启任务时间。它自带Node.js、Python等环境，是一个可以独立的使用环境。青龙面板还具有的优势是在于它可以支持多种脚本语言（如JavaScript、Python、shell等），用户可以多方面使用；它支持在线管理脚本、环境变量、配置文件；它支持在线查看任务日志；它支持秒级任务设置等等。

青龙面板的搭建是使用docker来拉取青龙的镜像镜像安装的。只要在可以运行docker的电脑上，都可以安装此控制面板，电脑端进行虚拟机的安装，然后搭建docker环境，最后安装青龙面板及一键拉取外挂程序。因此针对此次的课题设计，主要是用户在个人电脑端使用此面板，通过面板用户可以集中管理自己需要的外挂程序，最后进行京东任务的自动完成。

## 2.5 本章小节

本章主要是论述了本课题设计中应用的技术、工具、理论基础，分别对JavaScript程序语言、Node.js开发环境、docker容器技术和青龙控制面板进行详细的介绍。如介绍了JavaScript、Node.js、docker

# 第3章 设计分析

## 3.1 设计目的分析

随着购物软件的普及，很多购物软件也都出现了称作“薅羊毛”的现象，比如获得隐藏优惠券、现金红包等等，还有通过完成任务得到金币、完成游戏得到奖励等，获得这些可能让用户购物付款的时候出现更优惠的价钱。虽然完成这些可能只省了几毛或者几块，但是总而言之，还是让大家真实的“薅到了羊毛”，经过研究可以知道越来越多人愿意享受到这种“薅羊毛”的快乐。正所谓少一分是一分，省钱从“薅羊毛”开始。针对京东，完成它里面的游戏或任务，都是可以达到“薅羊毛”的快乐。故设计关于京东这款购物软件的外挂程序，可以方便用户“薅羊毛”，在某些情况下来不及或者没时间做任务、游戏的时候，外挂程序可以一键自动完成任务、游戏，帮助用户得到奖励。外挂程序的使用一定程度上节省了用户长时间花费在游戏任务，它使用户节省了很多在时间方面或者忘记了去做任务的问题。这个外挂程序可以当用户工作、吃饭、逛街、休息等时候，只需要启动此程序应用，就可以自动、按时完成用户指定的任务。

## 3.2 设计技术分析

此程序采用的JavaScript语言是一种计算机程序设计语言，因常被用于脚本开发也被称为脚本语言。JavaScript伴随着互联网的发展一起发展，互联网周边技术的快速发展，刺激和推动了它的发展，它的使用也越来越广泛。它可以实现从嵌入式系统到桌面应用程序、互联网应用等等几乎所有的工作，使用范围广，而且它相对于其他程序语言，属于比较简单的语言，易于学习和执行。JavaScript可以快速定义和开发对象，开发过程多样化，同时快速灵活。它还具有跨平台性，不依赖于操作系统，仅需要浏览器的支持或者在具有Node.js的开发环境中使用，因此一个JavaScript脚本在编写后可以在任意机器上使用。对于Node.js，它是目前十分流行的一种网络编程技术，是一个几乎可用于任何项目的流行工具，它自带http服务器的支持，不需借助外网即可方便的建立Web应用；是一个开源跨平台的服务器端JavaScript运行环境。Node.js搭配JavaScript程序语言使用，Node.js可以使Javascript更加强大，为它添加许多强大的功能，极大地拓展了JavaScript的用途和编程能力。Node.js还可以为JavaScript提供操 作文件、创建HTTP服务、 创建TCP/UDP服务等的接口。此外挂程序是基于docker这个环境平台，docker可以把数据、脚本快捷打包放入一个可移植的容器中，然后安装到有需要的Windows、Linux系统的平台上运行。通过docker拉取一个镜像，然后可以用来存放外挂程序，方便用户操作。

上述这三个技术都已经发展到了一定阶段，具有相对成熟的技术支持，具有的优势也是非常受大众喜爱的技术。课题通过使用这些技术，更好的对课题进行分析与设计。

## 3.3 设计任务分析

课程是设计一个基于docker的京东外挂程序，给京东用户带来不一样去完成任务方式的体验感。京东购物软件中任务、游戏繁多，因此设计内容主要是针对京东的东东萌宠、种豆得豆任务，然后还涉及领京豆额外奖励、京豆变动告知情况、取关京东店铺和收藏商品。通过为这些游戏、任务设计外挂程序，实现自动完成任务的目的。外挂程序是采用JavaScript脚本语言进行程序的设计、编写，从而帮助用户在这东东萌宠、种豆得豆等游戏中实现一键运行外挂程序自动完成任务的效果。外挂程序的设计让原本需要时刻盯着京东里的任务，变成了只需要一键使用外挂程序，就可以自动实现任务，轻松省时，节省出来的时间还可以做一些其他需要完成的任务，相当于一时多用。

## 3.4 Visual Studio Code应用软件分析

通过使用Node.js可以实现JavaScript脱离浏览器运行，为了更方便展示外挂程序，安装Node.js环境，使外挂程序得到更好的使用。本次外挂程序代码设计主要是使用Visual Studio Code，然后搭建Node.js环境来进行JavaScript脚本的调试、运行。

Visual Studio Code（通常简称VS Code）是一款于2015年4月由微软免费开源的现代化轻量级代码编辑器，目前非常流行的一款代码编辑器。它也是一个可跨平台的源代码编辑器，可以运行于Mac OS、Windows、Linux之上。由于其支持插件扩展，通过丰富的插件用户可以获得更多高级的功能，所以通常作为IDE使用。一经推出，便深受广大程序开发人员的喜爱[11]。使用VS Code比较方便简单，上手也比较容易。

Visual Studio Code具有丰富的插件安装功能和可扩展的生态系统，该软件的应用优势有：

（1）它具有多个语言相关类插件，支持多种语言编写，如Python、Java、JavaScript、HTML、C++等等。

（2）它是一款流行的免费开源的代码编辑器，安装、设置简单，界面美观，自带丰富多样的主题。

（3）它支持各种文件格式，使用代码提示类插件的智能提示功能、强大的调试功能。

（4）它具有对[JavaScript](https://baike.baidu.com/item/JavaScript/321142" \t "https://baike.baidu.com/item/Visual%20Studio%20Code/_blank)和[Node.js](https://baike.baidu.com/item/Node.js/7567977" \t "https://baike.baidu.com/item/Visual%20Studio%20Code/_blank)的内置支持。

## 3.5 流程分析

### 3.5.1 整体设计流程

根据课题内容进行设计分析，其设计流程主要分为三个大方向：首先是程序设计，此次外挂是通过JavaScript语言设计的，要想课题能够实现，要先根据京东任务分析设计方案，分析通过JavaScript怎样才可以实现外挂程序；其次是测试和查看，根据设计出的外挂程序，测试运行程序，查看它是否有达到基本目标，是否通过它完成了任务；最后是进行环境搭建，外挂程序是课题的中心设计，通过它才实现这个设计，但为了更好、更方便的应用外挂程序，搭建了一个docke环境，在docker中引用青龙面板镜像，通过青龙面板作为控制面板来应用外挂程序，更大程度上方便用户对外挂程序的使用。整体设计流程如下图3.1所示。

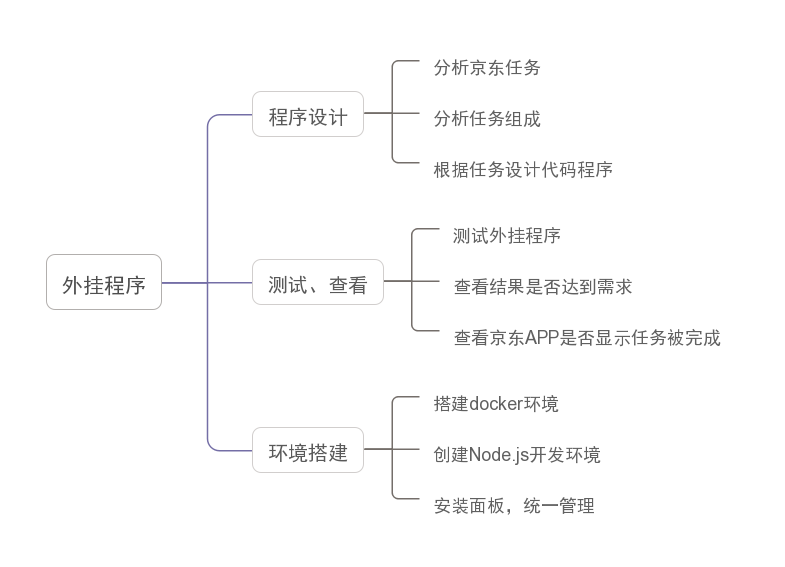


图3.1 整体设计流程图

### 3.5.2 外挂程序任务流程

根据京东APP里的游戏、任务选择要设计的哪方面的外挂程序，再根据游戏任务界面分析它需要设计的内容涵盖方面。此次内容主要针对的是京东APP的小游戏东东萌宠和种豆得豆设计外挂程序，然后也涉及到领取京豆额外奖励、取关店铺和收藏商品、京豆变动告知相关的外挂程序。根据具体的京东任务设计合适的外挂程序，通过京东APP与青龙面板进行连接，在面板中通过使用外挂程序，从而达到自动完成任务，帮助用户更速度的完成任务要求的目的。流程图如图3.2所示。

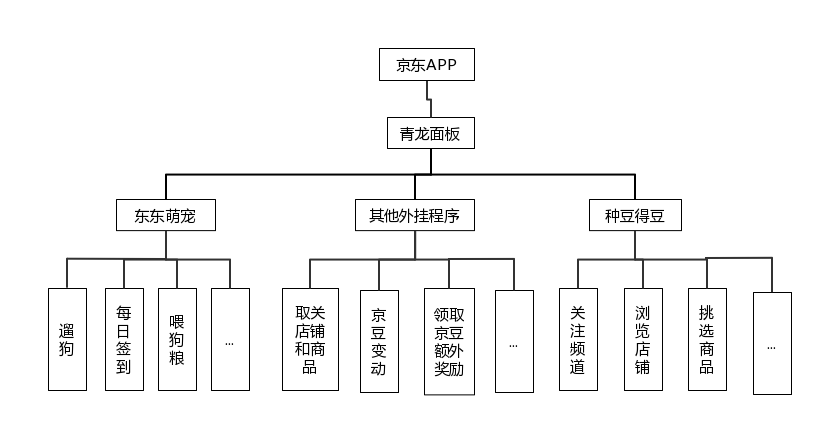


图3.2 外挂程序任务流程图

## 3.6 本章小节

本章从此次设计方向进行分析，对设计目的、设计技术、设计任务进行分析，还分析了Visual Studio Code代码编辑器以及整个流程分析。在流程分析中，主要叙述如何展开课题的设计以及此时外挂程序涉及到京东的哪些游戏、任务等。

# 第4章 外挂程序分析

此次针对京东设计的外挂程序，其内部游戏、任务繁多，不一一例举，针对几个外挂程序应用进行分析，主要是京东种豆得豆、东东萌宠、京豆变更告知及其他需要的程序功能。设计要求主要通过代码去实现功能，通过代码运行达到完成任务的目的。设计此程序实现了一键自动完成这些任务，节省时间一个一个去点击完成，达到快速完成任务的效果。

外挂程序能够为用户个人服务开始的第一步，是需要获取用户的京东cookie。外挂程序通过与cookie的绑定，即与用户的京东APP进行连接，外挂程序便与京东APP的游戏、任务绑定，然后就可以开始执行外挂程序达到完成任务的目的。

## 4.1 京东种豆得豆任务

### 4.1.1 任务组成

进入京东APP首页，点击“种豆得豆”选项进入任务中。种豆得豆是一款瓜分京豆互动游戏，用户通过做任务收集营养液培养小小豆，周期结束后即可瓜分京豆，然后京豆可兑换现金或其他奖励。其中种豆得豆任务中有逛逛会场、母婴家清、关注频道、挑选商品等等，任务随机产生。部分任务显示图4.1所示。



图4.1 种豆得豆部分任务展示图

“种豆得豆”任务入口在京东APP首页，第一次使用外挂程序，需要用户先进入任务界面激活，此后即可以直接进行外挂程序的操作达到一键自动运行。

### 4.1.2 程序分析

针对“种豆得豆”的任务构成，通过代码实现这些任务。如下为其中的三个任务的设计部分代码：

（1）关注频道

此任务是通过关注一个频道可随机获得一瓶营养液，总共可以获得三瓶，主要的设计思路为：

首先根据任务要求需要定义出一个变量，未关注的频道数等于总共的数量减去已经关注数量的，即let unFinishedChannelNum = item.totalNum - item.gainedNum。

定义三个数组，即let goodChannelListARR = [], normalChannelListARR = [], channelList = []。

通过for循环进行频道关注的实现：

for (let i of goodChannelList) {

          if (i.taskState === '2') {

            goodChannelListARR.push(i); }

        }

for (let j of normalChannelList) {

          if (j.taskState === '2') {

            normalChannelListARR.push(j); }

        }

（2）浏览店铺

此任务是通过关注店铺随机获得一瓶营养液，总共可以获得四瓶，其关键设计代码如下所示：

首先需要定义出店铺清单数组，即let moreShopListARR = []。

通过for循环进行浏览店铺，在for循环中定义j为店铺清单，通过if条件语句进行判断，判定所给条件是否满足，从而实现浏览店铺的设计，即：

for (let j of shopList) {

if (j.taskState === '2') {

shopListARR.push(j); }

}

在通过在for循环中使用if语句判断条件是否满足，即：

for (let shop of shopList) {

if (shopRes && shopRes.code === '0') {

if(shopRes.data&&shopRes.data.nutrState&&shopRes.data.nutrState === '1'){

unFinishedShopNum --; }

}

}

（3）挑选商品

此任务通过关注商品随机可得一瓶营养液，总共可以得到六瓶。主要设计代码如下所示：

在一个for循环中定义商品清单的商品显示，在定义商品库存量单位编码skuId产品任务编码productTaskId，再根据对应到京东的商品，定义到一个总的变量body中，如下所示：

for (let product of productList) {

          const { skuId, productTaskId } = product;

          const body = {

            "monitor\_refer": "plant\_productNutrientsTask",

            "productTaskId": productTaskId,

            "skuId": skuId

          }

再定义一个结果productRes，用它表示完成商品关注的情况，在根据if判断语句，判断当结果为关注成功时候，对应的可需要关注商品数量就减1。

 const productRes = await requestGet('productNutrientsTask', body);

          if (productRes && productRes.code === '0') {

              unFinishedProductNum --; }

          }

if判断条件依次进行下去，定义的未完成关注的商品数量为unFinishedProductNum，当它<=0时，即关注的商品数量达到6个，就自动跳出if语句，挑选商品任务就完成。

 if (unFinishedProductNum <= 0) {

            console.log(`${item.taskName}任务已做完\n`)

            break;

          }

“种豆得豆”的目的是通过做任务然后收取营养液，最后培养小小豆，当周期结束后即可瓜分京豆，获得京豆奖励。所以当任务被完成时，需要收取营养液，故设计代码为营养液产生的时候自动领取的程序。通过在for循环中定义变量，当符合要求时，等待任务完成就可以自动收取营养液，部分代码展示：

for (let bubbleInfo of plantBeanRound.bubbleInfos) {

console.log(`收取-${bubbleInfo.name}-的营养液`)

   await cultureBean(plantBeanRound.roundId, bubbleInfo.nutrientsType)

   console.log(`收取营养液成功`)

}

### 4.1.3 结果分析

通过外挂程序运行后的结果分析，设计基本达到预想的目标。采用JavaScript代码去设计符合每个任务的程序，通过代码程序实现完成任务的目的，让用户不需要一个一个浏览店铺、关注商品或完成其他任务。若需要观察任务完成结果，用户可以通过进入京东APP的种豆得豆界面查看任务是否完成，也可以通过查看青龙面板运行任务后的日志，部分日志展示图见图4.2。



图4.2 种豆得豆任务进度图

## 4.2 东东萌宠任务

### 4.2.1 任务组成

在使用此外挂程序之前，用户如果第一次玩此游戏做任务，需要先激活游戏。进入步骤为，点击京东APP“我的”界面，点击进入“东东萌宠”游戏，根据提示要求进行规划。用户根据想法选择需要领养的狗狗和可兑换的奖品，京东APP的“东东萌宠”游戏便也正式开启。游戏中根据选择不同的奖品对应袋不同的勋章，任务完成的难道要求也就不相同。根据游戏界面中，可以看到可以通过做哪些任务获得对应的奖励。

当东东萌宠被激活之后，外挂程序就可以开始发挥它的作用了。东东萌宠界面的任务主要有每日签到、喂狗粮、遛狗、浏览推荐的商品或指定会场等等来赚取狗粮，任务随机产生。用户通过做任务获得狗粮，对宠物进行投喂狗粮，积攒好感度升级勋章即可获得相应金额的无门槛红包或其他优惠奖励。京东APP中东东萌宠部分任务见图4.3。



图4.3 东东萌宠部分任务展示图

### 4.2.2 程序分析

根据“东东萌宠”游戏的任务构成，外挂程序的部分任务设计内容如下所示：

（1）遛狗

每日遛狗任务次数上限为10次。根据东东萌宠中遛狗任务形式，设计一个函数async function petSport()，在函数中通过do循环进行设计。

从第一次遛狗开始，以此类推遛狗次数每次增加1，当遛狗完成，便可领取遛狗奖励，设计为：

do {

    let response = await request(arguments.callee.name.toString())

    console.log(`第${times}次遛狗完成`);

    resultCode = response.resultCode;

    if (resultCode == 0) {

      let sportRevardResult = await request('getSportReward');

      console.log(`领取遛狗奖励`);

      }

    times++;

  } while (resultCode == 0 && code == 0)

  if (times >= 1)

}

因遛狗次数最多为10次，所以当遛狗次数达到11次后需要跳出循环，设计思路为当resultCode == 1013时候循环结束：

else if (resultCode == 1013) {

      let sportRevardResult = await request('getSportReward', {"version":1});

      console.log(`领取遛狗奖励领取`);

      if (sportRevardResult.resultCode == 0) resultCode = 0

    }

（2）投喂狗粮

每日投喂狗粮上限为100g，每10g为一次投喂，最多投喂10次，故可根据已有的狗粮÷10取整数部分即为可以投喂的次数，设计代码为：

let finishedTimes = $.taskInfo.feedReachInit.hadFeedAmount / 10;

需要投喂狗粮的次数设计代码为：let needFeedTimes = 10 - finishedTimes;

根据投喂方式，当每次投喂10g后，又可以进行下一次投喂，当投喂的狗粮结果编码为3003时，即狗粮剩余不足，投喂结束，剩余的狗粮保留下来，再用于第二天的投喂使用。设计程序为：

 do {

    console.log(`还需要投喂${needFeedTimes}次`);

    const response = await request('feedPets');

    if (response.resultCode == 0 && response.code == 0) {

      needFeedTimes--; }

    if (response.resultCode == 3003 && response.code == 0) {

      console.log('剩余狗粮不足, 投喂结束');

      needFeedTimes = 0; }

    tryTimes--;

  } while (needFeedTimes > 0 && tryTimes > 0)

  console.log('投喂任务结束...\n');

}

1. 每日签到

任务之一有通过进行一次签到来获取对应的狗粮。故可以通过创一个函数，在函数async function signInitFun()中通过if条件判断语句，如果当回应的数值码response.code === '0'和结果数值码response.resultCode === '0'同时都成立的时候，就可以进行每日签到，签到成功即可领取对应的狗粮奖励：

async function signInitFun() {

  console.log('准备每日签到');

  const response = await request("getSignReward");

  if (response.code === '0' && response.resultCode === '0') {

    console.log(`【每日签到成功】奖励${response.result.signReward}g狗粮\n`);

  }

}

“东东萌宠”游戏通过完成任务获得狗粮，然后通过投喂小狗，获得好感度，最后得到奖励。所以当好感度产生时，需要收取对应的好感度。通过创建一个async function energyCollect() 函数，通过if语句进行判断，当response.resultCode === '0'时，进行收取：

async function energyCollect() {

console.log('开始收取任务奖励好感度');

let function\_id = arguments.callee.name.toString();

const response = await request(function\_id);

if (response.resultCode === '0') {

message+=`【已获得勋章】${response.result.medalNum}块，还需收集${response.result.needCollectMedalNum}块即可兑换奖品“${$.petInfo.goodsInfo.goodsName}”\n`;

  }

}

### 4.2.3 结果分析

根据此程序的运行结果显示，基本达到预期所需要的设计效果。在输出结果中它会显示每个任务的完成情况，以及东东萌宠游戏的总进度，直接明了的给用户展示任务完成度。在第一次外挂程序使用执行后，用户可以手动去查看京东APP游戏任务，对比外挂是否实现功能，任务是否全部完成。下图4.2为此外挂程序的部分完成结果图：”

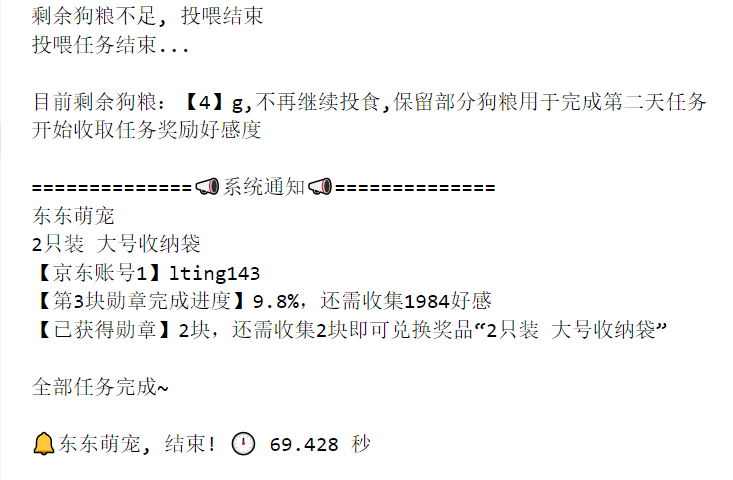


图4.2 东东萌宠完成情况图

## 4.3 取关店铺和商品

由于完成东东萌宠、种豆得豆等游戏任务的时候，会随机关注一些店铺和收藏商品，大多都是用户不需要的，因为这是通过做任务随机关注的。如果关注数量过多，用户一个一个点击取消关注十分麻烦、费时。因此设计这个可以取关店铺和收藏商品的外挂程序，释放用户的双手，节省用户的时间。用户只需要运行外挂程序，就可以自动取关京东里关注的店铺和收藏的商品。其设计过程为：

首先分析怎样去获取京东APP收藏商品，通过定义一个function getFollowGoods()函数作为对象属性来使用，在函数中创建一个headers接口对接，获取京东APP相关API的信息。API它是应用编程接口，在计算机程序的领域，它的作用就是可以让多个应用程序之间进行通讯，以便交换彼此的数据，作为其中的媒介使用。如在headers中获取host、cookie等，如下：

      headers: {

        "Host": "wq.jd.com",

        "Accept": "\*/\*",

      "Connection": "keep-alive",

        "Cookie": cookie,

'Referer': '收藏商品选项网址'

...... }

此外挂程序设计的关键部分除了与京东API接口获得连接，还需要获取京东关注的店铺和收藏商品对应的界面网址。同样在浏览器中进入https://m.jd.com中获取对应网址，进入关注店铺和商品选项中，即可以看到浏览器左上角显示的网址。通过这类网址，即可获取到京东中对应的商品、店铺信息。

同理也同样通过此调用获取关注店铺的情况，定义一个function getFollowShops()函数进行相关信息获取。

通过一个if判断语句，商品信息当符合条件的时候，就可以对商品进行的取消关注的操作：

if (res.iRet === 0 && res.errMsg === 'success') {

console.log(`取消关注商品---${item.commTitle.substring(0, 20).concat('...')}---成功`)

console.log(`已成功取消关注【商品】：${$.unsubscribeGoodsCount}个\n`) }

同理对关注的店铺进行判断，当符合时候，取消已关注的店铺：

if (res.iRet === '0') {

          console.log(`取消已关注店铺---${item.shopName}----成功`)

          $.unsubscribeShopsCount++;

          console.log(`已成功取消关注【店铺】：${$.unsubscribeShopsCount}个\n`)

        }

其中此外挂程序的部分执行结果如下图4.3所示：

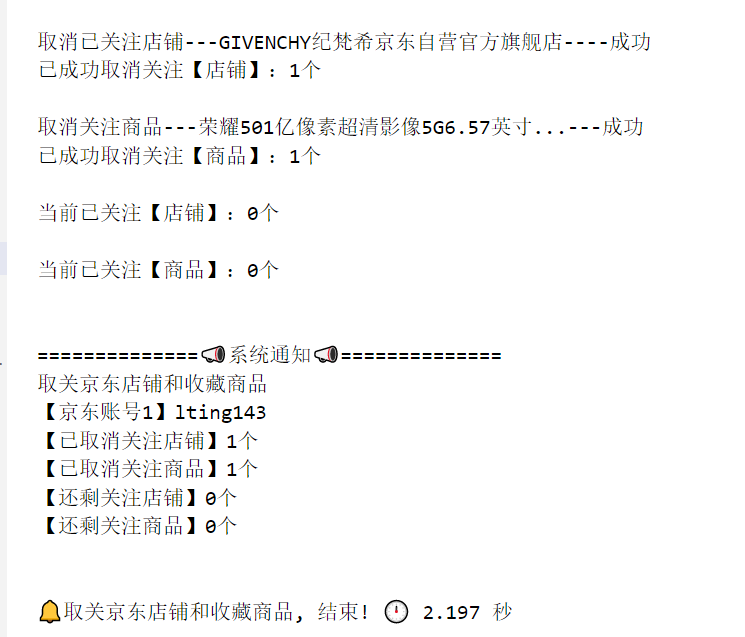


图4.3 取关店铺和商品运行图

## 4.4 京豆变动告知

设计的此外挂程序不属于京东的某个任务或游戏，它仅仅是用来显示京豆的变动情况，简单明了的告知用户京豆变动情况。通过它用户可以轻松的知道今天做的任务、游戏获得京豆情况，它显示了今日、昨日京豆支出或收入的变动状态，同样还展现了一些京豆在什么时候即将过期的情况。

此外挂程序为京豆变动告知，就需要清楚的给用户展示京豆情况。通过创建一个async的消息显示showMsg函数，在函数中列举出今日、昨日京豆收支等情况，通过这个设计让用户一下知道今日通过做任务获得京豆情况，即输出的信息显示有：

  allMessage += 今日收入：${$.todayIncomeBean}京豆 \n

今日支出：${$.todayexpenseBean}京豆 \n

昨日收入：${$.incomeBean}京豆 \n

昨日支出：${$.expenseBean}京豆 \n

当前京豆：${$.beanCount}(今日将过期${$.expirejingdou})京豆

如下图4.3为京豆变动告知执行结果显示：



图4.3 京豆变动情况图

## 4.5 其他设计分析

### 4.5.1 执行时间输出

在上述的每个外挂程序中，在输出结果界面有个统一的时间显示状态，即外挂程序脚本执行时的时间显示。

首先在设计时间脚本时，用到了时间戳这一个概念。时间戳是指从1970年1月1日的0点开始的。而0点的时间戳对应的是1970年1月1日的8点，因为使用北京时间，它是处在东八区，那么实际上时间戳是从我们(东八区)1970年1月1日的8点开始的，也就是格林尼治在那天是0点。全世界某一时刻的时间戳都是同一个值，只是在转换为我们看到的时间格式时加上了或减去了时区的值。因此使用getTimezoneOffset()方法返回当本地区的时区偏移量，时区偏差就是指格林尼治标准时间(GMT)相对于本地时间的分差。所以根据这个设计代码为：

console.log(`\n==============共${CookieJDs.length}个京东账号Cookie=========\n`)

console.log(`==================脚本执行- 北京时间(UTC+8)：${new Date(new Date().getTime() + new Date().getTimezoneOffset()\*60\*1000 + 8\*60\*60\*1000).toLocaleString()}===================\n`)

执行展示图为：

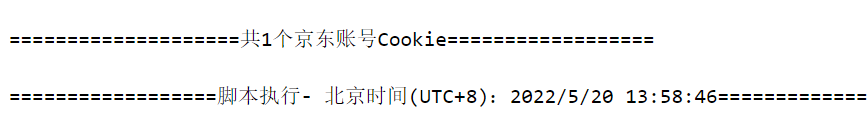


图4.4 执行展示图

### 4.5.2 账号名称输出

通过获取京东Cookie，外挂程序与用户京东的信息连接起来，故在外挂程序执行输出的结果中，设计一部分用来显示京东用户的使用名称，可用来区分京东账户。

首先定义一个变量$，赋值给new创建出的Env实例：

const $ = new Env('XXXX');

定义UserName为京豆用户账号名称，$.UserName通过匹配cookie中的pt\_pin获取用户账号名称：

 $.UserName = decodeURIComponent(cookie.match(/pt\_pin=([^; ]+)(?=;?)/) && cookie.match(/pt\_pin=([^; ]+)(?=;?)/)[1])

最后通过console.log输出京豆账号显示信息：

console.log(`\n\*\*\*\*\*开始【京东账号${$.index}】${$.nickName || $.UserName}\*\*\*\*\*\n`);

### 4.5.3 User-Agent的应用

User-Agent的中文名为用户代理，简称UA，它是一个特殊的[字符串](https://so.csdn.net/so/search?q=%E5%AD%97%E7%AC%A6%E4%B8%B2&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_44477431/article/details/_blank)头，也是Http协议中的一部分，属于头域的组成部分。换一种较为简单的说法，User-Agent是一种使得服务器能够识别客户使用的、向访问网站提供你所使用的浏览器类型、操作系统及版本、CPU 类型、浏览器渲染引擎、浏览器语言、浏览器插件等信息的标识。故通过User-Agent的应用，使外挂程序获取用户下载京豆APP使用的手机型号等等信息，从而实现为用户代理的作用。在一个JavaScript库，用于使用每日更新的数据生成随机匹配到合适的用户代理。故在此次外挂程序的设计中，通过定义获取到多个代理，如下图为一部分：

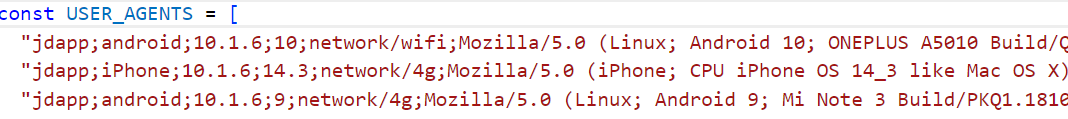


图4.5 User-Agent部分展示

## 4.6 本章小节

本章主要分析外挂程序的设计及实现，通过分析种豆得豆、东东萌宠等游戏、任务去设计外挂程序要实现的功能；通过游戏任务的组成，设计外挂程序需要实现的功能和目的。外挂程序主要是通过JavaScript程序语言进行设计，然后通过执行程序实现任务自动被完成的目的。

# 第5章 环境搭建

## 5.1 Node.js运行环境的使用

Node.js的使用让JavaScript脚本语言脱离浏览器也可以运行，可以直接在服务端进行操作，也可以直接在系统上进行操作，极大地拓展了JavaScript的用途。

Node.js出现之后，不管是服务器上，还是我们自己的的电脑上，只要本地存在了 Node.js的环境，就可以脱离浏览器使用、运行JavaScript代码了。只需要在本地计算机通过官方途径下载并安装Node.js。安装完成后，需要点击win+R，进入命令提示符（cmd）界面，然后输⼊node -v和npm -v命令去检验Node.js是否安装成功。Node.js搭配Visual Studio Code一起使用，可以更轻松的对JavaScript程序语言进行编写、执行、调试，操作起来更加方便。

## 5.2 docker的搭建

对于docker环境的搭建、安装可以通过多个渠道，如在服务器、路由器、电脑中都可以搭建docker环境。而此次课程设计是研究使用电脑进行docker环境的搭建，然后再拉取面板镜像。首先需要使用本地电脑下载VMware Workstation Pro，它是[VMware](https://zh.wikipedia.org/wiki/VMware" \o "VMware)公司推出的一款功能强大的桌面虚拟计算软件，具有[Windows](https://zh.wikipedia.org/wiki/Windows" \o "Windows)、[Linux](https://zh.wikipedia.org/wiki/Linux" \o "Linux)版本。提供用户可在单一的桌面上同时运行不同的操作系统。即软件可建立多个[虚拟机](https://zh.wikipedia.org/wiki/%E8%99%9A%E6%8B%9F%E6%9C%BA" \o "虚拟机)，同时运行多个不同[操作系统](https://zh.wikipedia.org/wiki/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F" \o "操作系统)。每个虚拟机是相互独立，拥有各自的[操作系统](https://zh.wikipedia.org/wiki/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F" \o "操作系统)；并且各虚拟机以及宿主机相互独立，互不影响，可独立进行操作及运行[应用程序](https://zh.wikipedia.org/wiki/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F" \o "应用程序)。因此使用VMware Workstation Pro创建虚拟机，在进行docker的搭建，具体步骤如下：

（1）创建一台centos7系列版本的Linux操作系统的虚拟机。手动进行IP地址的配置，配置IP地址为192.168.101.103，此时配置的IP地址即为通过浏览器访问进入青龙控制面板的地址。

（2）通过Xshell连接到虚拟机的centos系统。Xshell是一个强大的安全终端模拟[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053" \t "https://baike.baidu.com/item/Xshell/_blank)，通过它连接系统可以更方便的搭建docker环境，对docker进行操作。

（3）命令配置docker，如下所示：

1. 升级Centos系统

sudo yum -y check-update

1. 安装docker

curl -fsSL <https://get.docker.com/> | sh

C、docker命令配置

sudo systemctl start docker ，此命令是启动docker。

sudo systemctl status docker ，此命令是查看docker的状态。

sudo systemctl enable docker ，此命令是配置docker开机自启。

通过docker -v命令可以查看docker是否安装完成以及它的版本信息，如图5.1：

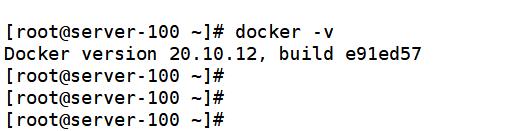


图5.1 docker版本信息

## 5.3 控制面板的搭建

为了更加方便用户对外挂程序的使用，此次课题中使用了青龙控制面板。青龙面板是一个辅助工具，可以帮助用户把外挂程序集中管理，同时也是一个定时任务的控制面板。它可以根据用户需求自己设置定时运行脚本任务，当到达定时的时间时，任务会自动运行。

### 5.3.1 拉取镜像

docker搭建安装完成后，就可以开始进行青龙面板的搭建。第一步是需要先拉取docker镜像，即使用命令在docker中进行面板镜像的拉取、安装。

docker pull whyour/qinglong:latest

whyour/qinglong为拉取的镜像名称，latest表示的是在docker容器中拉取的青龙面板镜像的版本系列，即拉取最新版的面板。

### 5.3.2 部署镜像

镜像拉取成功之后，还需要部署镜像，对青龙面板进一步的安装。通过使用命令进行所需文件的创建和面板的拉取安装。

docker run -dit \

-v $PWD/ql/config:/ql/config \

-v $PWD/ql/log:/ql/log \

-v $PWD/ql/db:/ql/db \

-v $PWD/ql/repo:/ql/repo \

-v $PWD/ql/raw:/ql/raw \

-v $PWD/ql/scripts:/ql/scripts \

-v $PWD/ql/jbot:/ql/jbot \

-p 5700:5700 \

--name qinglong \

--hostname qinglong \

--restart unless-stopped \

whyour/qinglong:latest

在系统根目录下创建ql目录，并在ql目录下创建config，db，log，repo目录，其中PWD表示显示工作目录路径，选择安装的文件所存放的位置，如ql/raw：/ql/raw为raw文件所在位置。容器的名称和hostname都命名为qinglong，文件名称命名为ql，指定端口映射为5700，latest为安装的青龙面板的最新版本号。用户也可以根据自己想法进行某些数据的更改，如更改端口号为5600；更改latest为其他版本，如版本号为2.11.0、2.10.3等等。

### 5.3.3 登录面板

镜像部署完成后，需要确认是否创建成功，通过docker images命令进行查看，如果有对应的镜像、版本等显示，即面板创建成功。

青龙面板拉取创建后，用户通过使用浏览器去访问http:// + IP地址 + 端口号，如http://192.168.101.100:5700的形式，IP地址即为创建的虚拟机的IP地址。登录后便进入青龙控制面板安装界面，如下图5.2所示：

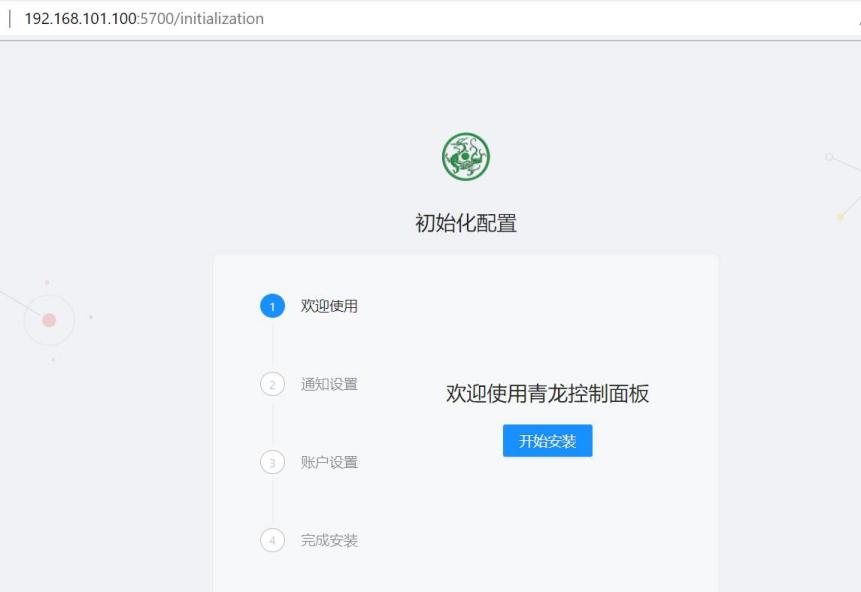


图5.2 面板初始化界面

进入面板安装界面后，点击开始安装，再依次按照步骤进行面板的安装，其中账号设置为用户自己设置，如设置用户名：admin，登录密码123456，这个账号就是后面登录进入青龙面板的账号。第四步完成安装后，便可以开始登录青龙面板，进入主界面中。进入主界面中，可以看到界面的左边从上往下依次有定时任务、环境变量、配置文件、脚本管理、依赖管理、对比工具等等功能。如图5.3所示：



图5.3 面板主界面图

## 5.4 本章小节

本章主要分析进行此外挂程序设计需要准备的环境以及方便用户使用搭建的面板环境。在环境需求方面，首先需要通过Node.js开发环境，它可以让JavaScript程序语言脱离浏览器被使用，扩大了JavaScript的用途。再通过利用docker环境，搭建一个控制面板，实现在面板上对外挂程序的控制、使用。

# 第6章 面板设置

## 6.1 拉取外挂程序

为了更加方便用户对外挂程序的使用，通过使用青龙控制面板来对接京东APP，将外挂程序集中存放在面板中，通过在面板上操作完成外挂程序的运行，最后完成京东任务。

登录进入面板中后，将已有的外挂程序拉取到面板里，点击左边的定时任务，点击右上角新建任务，然后根据要求进行任务创建。将外挂程序存放在Github中，然后通过新建任务，可以将外挂程序全部拉取到青龙面板中。Github于2008年4月10日正式上线，它是一个面向[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90/20720669" \t "https://baike.baidu.com/item/Github/_blank)及私有[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053" \t "https://baike.baidu.com/item/Github/_blank)项目的托管平台。作为开源代码库以及版本控制系统，Github拥有超过900多万开发者用户，在GitHub，用户可以十分轻易地找到海量的开源代码。将外挂程序通过Visual Studio Code连接到我的Github中，通过git命令进行连接。将外挂程序对接到到Gitnub中后，就可以在青龙面板中对外挂程序进行拉取。拉取界面如图6.1所示：



图6.1 定时任务拉取过程

其中上图中的名称是用户自己随机填写的名称，拉取命令为ql repo + Github中存放外挂程序存储库的链接，例如存储库为：，此存储库即为用户使用的外挂程序，用户统一拉取此链接。定时规则即设置每天什么时间段对拉取存储库任务进行更新，可定时0 0 0 \* \* \*，即每天的零点会启动这个任务，如图6.1所示。填写完成后，点击确定，然后在启动执行任务，可看到界面中的状态一栏显示运行中，即在开始拉取京东外挂程序，可通过点击操作栏中的日志查看拉取外挂程序的进度，如下图6.2为日志中的部分截图：



图6.2 拉取外挂程序日志图

当拉取的过程成功后，关于京东应用设计的外挂程序就被集中存放在青龙面板上了，用户可以根据自身需求在面板上对外挂程序进行相关操作。图6.3为拉取外挂程序成功后的部分显示图，如下所示：

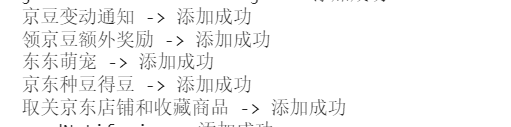


图6.3 外挂程序添加显示图

## 6.2 时间设置

上一步骤为外挂程序的拉取，当拉取成功后用户可以根据自己的需要或者任务的需要自定义外挂程序的执行时间。只要在保持电脑可以长时间待机的状态，不管用户在何地、在何时、再做何事，当到达设置的时间后，青龙面板中的外挂程序就会自启动、自动执行，直至游戏、任务被完成。

通过在面板中调节外挂程序自动执行的时间，可以很大程度上解决用户隔一段时间完成任务、或者因忘记时间做任务的情况。用户根据游戏、任务产生的时间，可自行设置符合的执行时间。到点外挂程序便可执行，实现自动做任务，方便又快速的做任务。对于任务定时的规则，其时间的定时规则格式为：\* \* \* \* \* \* (五/六位cron时间格式)。如下表6.1为时间格式表示方式。

表6.1 时间格式表示方式

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \*（0-59） | \*（0-59） | \*（0-23） | \*（1-31） | \*（1-12） | \*（0-7） |
| 秒（可选） | 分 | 小时 | 日 | 月 | 星期 |

表格中的第1个位置表示秒，第2个位置表示分，第3个位置表示时，第4个位置表示每月的哪一日，第5个位置表示哪个月，第6个位置表示是每周的周几。数字之间是需要用空格隔开；不限制第几个位置的用“\*”号替代即可；定期执行的时间用“？”替代；同一个时间位有多个选项用“，”连接；同一个时间位在一个区间的用“-”连接即可。

一些定时示例如下所示：

0 0 1 \* \* ？ ——表示在每天的1点钟触发运行

0 0 1 \* \* \* ——表示在1点钟触发运行

0 0 10,12,14 \* \* ？ ——表示在每天的10点、12点、14点都会触发运行

0 0-5 14 \* \* \* ——表示在下午14:00到14:05期间每隔1分钟触发运行  
 0 30 9 ? \* \* ——表示在每天的上午9:30触发运行

......

## 6.3 安装依赖管理

青龙面板安装好后，当执行一些外挂程序时会提示找不到依赖，外挂程序执行失败。所以可以看到在青龙面板主界面中，它的左边界面有一个依赖管理功能，此功能可用来解决因依赖导致外挂程序执行失败的问题。所以若使用青龙面板来管理、执行外挂程序，需要安装与外挂程序相关的依赖，管理运行程序，程序才可以正常的使用、执行。

根据设计出的JavaScript外挂程序代码，它需要在Node.js环境中安装相关依赖，所以根据需求所安装的依赖名称及其版本有：anvas 2.8.0，png-js 1.0.0，crypto-js 4.0.0，download 8.0.0，got 11.5.1，http-server 0.12.3，jsdom 17.0.0，ws 7.4.3，tunnel 0.0.6，qrcode-terminal 0.12.0，request 2.88.2，tough-cookie 4.0.0等等。

点击青龙面板界面中依赖管理选项，选择将依赖安装在Node.js环境中，根据外挂程序分析安装所需依赖，在面板中进行安装。部分安装完成的结果如下图所示：

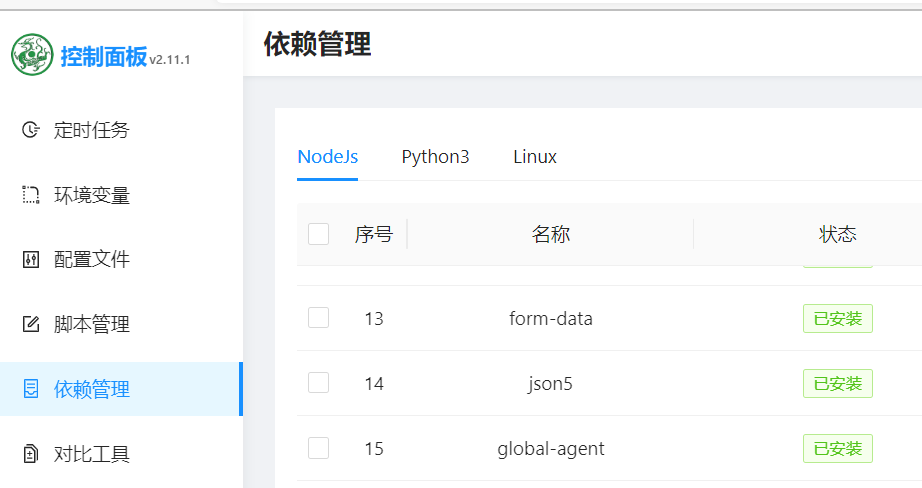


图6.4 依赖安装完成部分展示图

## 6.4 环境变量的配置

在使用青龙面板执行外挂程序的时候，需要将面板和用户京东APP连接起来，才能使外挂程序实现它的功能。因此需要通过在面板中的环境变量这一功能中进行cookie的配置。环境[变量](https://baike.baidu.com/item/%E5%8F%98%E9%87%8F" \t "https://baike.baidu.com/item/%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F/_blank)是在操作系统中一个具有特定名字的对象，它包含了一个或者多个应用[程序](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F" \t "https://baike.baidu.com/item/%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F/_blank)所将使用到的信息。通过获取京东APP的环境变量，可以让外挂程序与任务对接，从而做到用户的专属功能。

进行环境变量配置的方式为：通过在浏览器中访问https://m.jd.com，进入京东手机端网页版界面，其中m即指Mobile 。通过获取pt\_key和pt\_pin的信息，然后在青龙面板中进行JD\_COOKIE的配置。环境变量配置完成后，青龙面板即与用户的京东APP上的信息连接成功。

## 6.5 其他设置

当外挂程序的脚本需要被修改、编辑、删除的时候，可以直接在青龙控制面板中进行操作。通过点击界面中脚本管理的选项，这个功能可以直接在面板中对脚本进行编辑等，即当发现程序需要修改、或者不需要使用此外挂程序，点击选中此外挂即可进行相关处理。

当需要查看外挂程序执行即展示图，可以通过任务日志这个选项查看今天或者之前的外挂程序执行结果、日志。

进入青龙面板的系统设置选项中，还有许多其他的功能，例如可以进行安全设置、应用设置、通知设置、登陆日志和其他设置的更改。

## 6.6 本章小结

本章主要叙述青龙控制面板的设置，叙述它的使用以及操作方法。青龙面板是一个可定时使用的控制面板，它可设置自动执行时间，在确保有执行的条件下时，当到达设置的时间，外挂程序就会在青龙面板中自动的执行，开始完成京东的游戏、任务。外挂程序执行结束，可以通过日志查看外挂程序是否成功。利用它的这些优势让外挂程序更符合用户使用。

# 总 结

此次课题是设计一种基于docker的外挂程序，它可以提升京东用户对完成种豆得豆、东东萌宠游戏、任务的使用感。京东外挂程序是专门为用户完成游戏任务的一种服务，此外挂程序的合理使用可以提升用户

对于京东任务、游戏的完成感。当用户不想花费较多时间去完成游戏、任务的时候，用户可以手动点击面板中的外挂程序实现一键自动完成任务的目的，轻松简单操作；当用户想自己体验任务、游戏完成的过程，可以关闭外挂程序，自己花费时间去做各种各样的任务、游戏。基于docker的京东外挂程序可以被用户随用随停，根据用户的需要使用。

京东外挂程序主要通过JavaScript脚本语言进行编辑分析的，通过Node.js技术使外挂程序脱离浏览器可以直接在本地应用上使用，再通过docker容器搭建一个环境，在面板中集中管理，自由设置使用外挂程序的时间，方便京东用户根据自己的想法随时使用。

本次课题内容涉及技术优势在于：

1. 本外挂程序用JavaScript脚本编写，它不需要编译，在执行过程更轻量，运行速度更快。它也是一种安全性高的脚本语言，还可以跨平台使用。
2. 本课题使用了Node.js技术，它可以让JavaScript脚本语言脱离浏览器也能被使用，更加便利了用户对外挂程序的使用。
3. 本外挂程序通过使用docker，程序可以发布到任何只要安装docker引擎的服务器上，可以实现跨平台应用。因此外挂程序可以在多个途径上都被使用，增加了它可使用的范围。

虽然目前设计出的外挂程序通过测试，其运行表现稳定，实现了它们的价值。但由于时间问题，设计出的京东外挂程序的脚本较少。只设计了京东一小部分任务、游戏的程序，设计的外挂程序还未达到任务的所有功能都去实现。此次还有的一个不足之处就是：docker和青龙面板是通过在本地计算机创建虚拟机搭建的，所以在运行外挂程序时需要开机，才可以更好的满足根据用户设计的时间去运行外挂程序完成任务。

# 参 考 文 献

李娜.Docker容器技术的发展及应用研究[J].数字技术与应用，2018，36(11)：95-96.

张博.基于JavaScript的城市发展项目管理系统的设计与实现[D].吉林大学，2016.

王曙燕，郑佳妮，孙家泽.基于页面对象的Web应用测试用例生成方法[J].计算机应用，2020，40(1)：212-217.

[王真星，石林祥，沈武敏.嵌入式Web服务器应用的关键技术研究[J].计算机工程与应用，2009，45(28)：79-82.](http://vpn.hnie.edu.cn/vpn/895/https/kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&filename=JSGG200928023&v=MDI1NTZxQnRHRnJDVVI3aWZZK2RuRnl6blZidkJMejdNYWJHNEh0ak9wNDlIWjRRS0RIODR2UjRUNmo1NE8zenE=&uid=WEEvREcwSlJHSldSdmVqMVc3NWZQbGtscnYrQkpkS2p3cUhoNThEbXdyND0=$9A4hF_YAuvQ5obgVAqNKPCYcEjKensW4IQMovwHtwkF4VYPoHbKxJw!!" \o "王真星,石林祥,沈武敏.嵌入式Web服务器应用的关键技术研究[J].计算机工程与应用,2009,45(28):79-82." \t "http://vpn.hnie.edu.cn/vpn/895/https/kns.cnki.net/KXReader/_blank)

余任杰，王风硕.基于Node.js的网上图书销售系统设计[J].电脑知识与技术，2021，17(34)：83-84+92.

王金龙，宋斌，丁锐.Node.js：一种新的Web应用构建技术[J].现代电子技术，2015，38(06)：70-73.

Paul Krill.Next step after Node.js:Framework for universal JavaScript apps[J].InfoWorld.com，2016.

王明芬，陈锐旺.基于Docker的Web系统架构设计[J].福建师大福清分校学报，2020，(05)：8-14.

Serdar Yegulalp. What is Docker? The spark for the container revolution[J]. InfoWorld.com，2021.

刘国庆.基于Docker的云集成开发环境的研究与实现[D].电子科技大学，2017.

韩骏.Visual Studio Code权威指南[M].北京：电子工业出版社，2020.

# 致 谢