北京邮电大学

计算机学院（示范性软件学院）

实验报告

课程名称： 区块链技术原理

项目名称： 以太坊溯源系统实验

项目完成人：

姓名： 孙津硕 学号： 2023140748

姓名： 赵振宇 学号：

姓名： 学号：

姓名： 学号：

指导教师： 杨 谈

日 期： 2023 年 12 月 23 日

# 实验目的

本次实验的主要目的是通过编写部署智能合约及创建基于以太坊的溯源系统，让参与者深入理解以太坊的智能合约编写与部署流程，以及如何基于以太坊私链构建一个实际应用场景——溯源 Dapp。通过本实验，学会并掌握智能合约的编写、编译和测试方法，并能够运用以太坊平台创建并实现一个完整的区块链防伪溯源系统。

# 实验内容

本次实验旨在通过实践探索区块链技术在防伪溯源领域的广泛应用潜力，并结合以太坊智能合约开发一个基于食品追溯的Dapp。实验内容具体涵盖了以下四个核心步骤：

1. 首先，参照实验一的指导搭建并配置以太坊私链环境，确保其成功启动运行，为后续实验提供稳定的区块链基础设施。

2. 在此基础上，设计并编写名为“FoodTransportation”的智能合约，该合约专门用于记录食品从生产至销售各环节的关键信息（如地点、负责人和时间戳等），实现对食品流转过程的透明化追溯。

3. 利用 Remix 这款在线 Solidity 编辑器完成智能合约的编译与测试工作，确保合约逻辑无误并在以太坊虚拟机上能够正确执行。

4. 接着，采用 truffle 框架和 webpack 打包工具构建并部署一个基于以太坊的溯源 Dapp。该 Dapp 集成了 MetaMask 插件，用户可以在 Chrome 浏览器中进行直观、便捷的交互操作。

随着区块链技术的快速发展，其在防伪溯源场景的应用价值日益凸显，受到众多科技巨头的关注与布局。其中，数据溯源的核心在于如何有效地构建数据模型，以确保源头信息的有效获取、存储及后期利用。为此，实验过程中需针对不同业务领域和应用场景抽象出合适的数据模型，并规范各类数据接入标准；同时，将复杂的业务流程细化成各个阶段，针对性地整合不同阶段产生的业务数据；最后，运用数据特征标识技术，实现在整个生命周期内追踪和查询数据的历史版本。

智能合约作为区块链技术的一大革新，其工作原理类似于传统应用程序中的条件触发机制——即if-then语句，当预设条件满足时，智能合约会自动执行相关条款，且由于其运行于去中心化的区块链网络之上，无需任何中介参与，确保了自动化和可信的执行过程。

以太坊作为一个建立在区块链和加密货币理念上的开源、图灵完备的区块链平台，为智能合约提供了安全可靠的运行环境。丰富的API接口使得开发者能轻松地在以太坊平台上编写、编译并部署各种由区块链技术支持的智能合约应用。

综上所述，本次实验集合了四个关键组成部分：搭建以太坊区块链服务的虚拟机服务器、实现防伪溯源功能的FoodSafe智能合约源代码、集成在Chrome浏览器中的MetaMask插件以及用户友好的前端交互界面。整个溯源Dapp项目依托truffle框架进行整体管理，并借助webpack工具进行了高效的前端模块打包处理，从而实现了区块链技术在实际防伪溯源场景中的创新应用。

# 实验环境

本实验需要以下软硬件环境支持：

Linux 或 Windows 系统

Remix（在线 Solidity 编辑器）

Truffle 框架

Webpack 打包工具

以太坊 Geth 客户端

MetaMask 插件

# 附录

（附上实验文档，如：问题分析、设计方案、算法、设计图、程序、仿真结果、运行结果、调试心得等，具体内容根据实验要求来定。源代码请附在这里。源代码排版请特别注意，用5 号字体，行间距为单倍行距。注意节省空间，不要浪费纸张。）

## 问题分析

（说明你对这个问题的理解，包括，这个程序要解决什么问题、功能、性能、健壮性等）

### 目标

本实验的核心目标是利用区块链技术，尤其是以太坊智能合约的特性，构建一个去中心化、公开透明且不可篡改的食品溯源系统。通过“FoodTransportation”智能合约记录食品从生产、加工、运输到销售等各个环节的关键信息，确保食品供应链的全程可追溯性，从而提升食品安全监管效率，增强消费者对食品来源的信任度。

### 功能

#### 功能一：添加新位置信息。

智能合约提供了一个名为 addNewLocation 的接口函数，允许参与者上传新的流转环节信息，包括地理位置、负责人、时间戳等关键数据，并将这些信息永久存储在区块链上。

#### 功能二：查询位置信息。

当用户按“查找”按钮之后，程序显示输入框，等待用户输入。用户输入文字的同时，对用户的通信录按照字母或者汉语拼音进行检索。

为了方便用户追踪食品流向，设计了getLocationNum、getFoodName 和 getLocation 等查询函数。其中，getLocationNum 返回食品经过的所有流转位置总数；getFoodName 获取食品的基础信息如名称；getLocation 则用于检索指定位置的详细流转记录。

### 性能

#### 以太坊网络

考虑到以太坊网络的运行机制，智能合约的实际执行速度会受到网络状态（如区块生成速度、交易确认时间）的影响。在本地私链环境下进行开发和测试时，由于无需考虑网络延迟及共识过程，智能合约的操作具有较高的实时响应能力。

### 健壮性

#### 异常输入处理

为确保系统的稳定性和安全性，智能合约在设计中充分考虑了异常输入处理。例如，在调用 addNewLocation 函数时，对输入参数进行了严格的类型检查和边界控制，要求 locationNum 必须为有效整数，并且不能超过当前实际的位置记录数量。此外，针对可能出现的其他错误情况，智能合约还包含了相应的错误处理逻辑，以防止因非法操作导致的数据混乱或系统崩溃。

## 设计方案

**（在此处，用文字描述整体的设计，包括，分成多少个模块，多少个文件，多少个函数，每个模块、函数、文件的大致功能； 此外，还需要包含一个程序整体的流程图，反映各个模块的执行次序和关系；但是避免绘制细节的流程）**本实验中的 Dapp 设计采用了前后端分离架构。前端负责用户交互界面的设计与实现，使用现代Web开发技术构建直观易用的UI界面，通过集成MetaMask插件完成与后端智能合约的交互。后端部分主要由FoodTransportation智能合约构成，作为数据存储和业务逻辑处理的核心模块，其所有操作均基于以太坊区块链进行。

## 重要算法

（在此处，用流程图描绘重要的算法，例如排序、查找等。一般的程序代码不必在此说明）

### 算法一：xxxxxx

### 算法二：xxxxxx

## 实验结果

**（说明实验完成情况，有需要文字回答的题目，以及实验结果截图，请写在这里）**

**（在此处，详细描述运行的结果。包括：不同输入对应的不同结果，异常情况的执行结果）**

实验过程中，成功搭建并配置了以太坊私链环境，运用 Remix 编写了 FoodTransportation 智能合约并完成了编译和单元测试，确保其逻辑正确无误。随后，借助 Truffle 框架和 Webpack 打包工具部署了溯源 Dapp，并成功实现了与 MetaMask 的集成，使得用户能够在浏览器中便捷地与智能合约进行交互。

最终，通过在Dapp上模拟添加和查询食品流转信息的操作，验证了智能合约的有效性和整个防伪溯源系统的可行性，充分达到了本次实验深入理解智能合约编写、部署以及基于以太坊构建实际应用场景的教学目的。

## 调试心得

（总结你在调试程序时的收获，对于某一类或者某几类警告、错误的处理方法。这些心得，能够对学习C语言程序设计的新手具有一些指导作用）