以太坊学习：第五天

目录

EVM

- 以太坊虚拟机EVM是智能合约的运行环境
- 作为区块验证协议的一部分，参与网络的每个节点都会运行EVM。他们会检查正在验证的块中列出的交易，并且运行由EVM中的交易触发的代
- EVM不仅是沙盒封装的，而且是完全隔离的，也就是说在EVM中运行的代码时无法访问网络、文件系统和其他进程的，甚至智能合约之间的访
的
- 合约以字节码的格式（EVM bytecode）存在于区块链
- 合约通常以高级语言（solidity）编写，通过EVM编译器为字节码，最终通过客户端上载部署到区块链网络中

EVM和账户

- 以太坊中由两类账户：外部账户和合约账户，它们公用EVM中同一个地址空间
- 无论账户是否存储代码，这两类账户对EVM来收处理方式是完全一样的
- 每个账户在EVM中都有一个键值对形式的永久化存储。其中ｋｅｙ和value的长度都是256位，称之为存储空间storage

EVM和交易

- 交易可以看作是从一个账户发送到另一个账户的消息，它可以包含二进制数据（payload）和以太币
- 如果目标账户含有代码，此代码会在EVM中执行，并以payload作为入参，这就是合约的调用
- 如果目标账户是零账户（账户地址为0），此交易就将创建一个新合约，这个用来创建合约的交易的payload会被转换为EVM字节码并执行，执
为合约代码永久存储

EVM和ｇａｓ

- 合约被交易触发调用时，指令会在全网的每个节点上执行：这需要消耗算力成本；每一个指令的执行都有特定的消耗，ｇａｓ就是用来量化表
耗
- 一经创建，每笔交易都按照一定数量的ｇａｓ预付一笔费用，目的是限制执行交易所需要的工作量和为交易支付手续费
- EVM执行交易时，ｇａｓ将按特定规则逐渐耗尽
- ｇａｓ　ｐｒｉｃｅ时交易发送者设置的一个值，作为发送者预付手续费的单价。如果交易执行后还有剩余，ｇａｓ会原路返回
- 无论执行到什么位置，一旦ｇａｓ被耗尽，将会触发一个ｏｕｔ－ｇａｓ异常。当前调用（call　frame）所作的所有状态修改都将被回滚

EVM数据存储

storage

- 每个账户都有一块初九话的存储空间，成为storage，这是一个将256位字映射到256位字的ｋｅｙ－ｖａｌｕｅ存储区，可以理解为合约的数据
- 永久存储在区块链中，由于会永久保存合约状态变量，所以读写的ｇａｓ开销也最大

Memery

- 每一次消息调用，合约会临时获取一块干净的内存空间
- 生命周期仅为整个方法执行期间，函数调用

Stack

- EVM不是基于存储器的，时基于栈的，因此所有的计算都在一个被称为栈的区域执行
- 存放部分区域值类型变量，几乎免费使用的内存，但有数量限制

EVM指令集

- 所有的指令都是针对“256位的字（word）”这个基本的数据类型来进行操作
- 具备常用的算数、位、逻辑和比较操作，也可以做到有条件和无条件跳转
- 合约可以访问当前区块的相关属性，比如：它的块高度和时间戳

消息调用（Message Call）

- 合约可以通过消息调用的方式来调用其他合约或者发送以太币到非合约账户
- 合约可以决定在其内部的消息调用中，对于剩余的ｇａｓ，应发送和保留多少
- 如果在内部信息调用时发生了ｏｕｔ－ｏｆ－ｇａｓ异常（或其他任何异常），这将由一个被压入栈顶的错误值所指明；此时只有与该内部消息送的ｇａｓ挥别消耗掉

委托调用（Ｄｅｌｅｇａｔｅｃｌｌ）

- 一种特殊类型的消息调用
- 目标地址的代码将在发起调用的合约的上下文中执行，并且ｍｓｇ．ｓｅｎｄｅｒ和ｍｓｇ．ｖａｌｕｅ不变
- 可以由此实现“库”（library）：可复用的代码库可以放在一个合约的存储上，通过委托调用引入相应代码

合约的创建和自毁

- 通过一个特殊的消息调用create calls，合约可以创建其他合约（不是简单的调用零地址），该其他合约不是存储在区块链上的，而是存储在内
- 合约代码从区块链上移除的唯一方式是合约在合约地址上的执行自毁操作ｓｅｌｆｄｅｓｔｒｕｃｔ；合约账户上剩余的以太币会发送给指定的其存储和代码从状态中被移除