

# EVM 的两种状态

Machine State 与 World State · 小学生也能懂的版本

---

## 为什么要分「两种状态」？

### 一句话给小学生听

EVM 有「正在算的状态」，也有「已经存好的状态」。

### 先给你大方向

在 EVM 里，状态分成两大类：

- **Machine State**: 算题目时临时用的
- **World State**: 算完后真的写进区块链的

### 小学生版比喻

- Machine State = 草稿纸
- World State = 成绩单

## 什么是 Machine State ?

### 一句话

Machine State = EVM 正在执行一笔交易时，用的临时状态。

### Machine State 的特点

- 只存在于「这一次交易执行期间」
- 交易结束就会消失
- 不会被写进区块链

## 小学生版

你在写数学作业时：

- 草稿写满一页
- 最后只交答案
- 草稿不会存档

草稿 = Machine State

## Machine State 里面有什么？

### 四个一定要记

Machine State 主要包含这四样东西：

### Machine State 的组成

- **Program Counter (PC)**: 现在执行到哪一行指令
- **Stack**: 叠盘子一样放数字（最多 1024 层）
- **Memory**: 交易期间的临时记忆
- **Gas Remaining**: 还剩多少 Gas 可以用

### 小学生对照表

- PC = 手指指到课本第几行
- Stack = 叠起来的算式数字
- Memory = 白板
- Gas = 时间 / 体力

## Machine State 会不会留下来？

### 非常重要

不会。

### 为什么不会留下来？

- Machine State 只是「计算过程」
- 区块链只关心「最后结果」
- 不然链会爆炸大

### 一句话记住

Machine State = 用来算，不是用来存。

## 什么是 World State ?

### 一句话

World State = 整个以太坊世界当前真正的状态。

### World State 有什么？

World State 是一个巨大的资料结构，记录：

- 所有账户余额
- 所有合约的 Storage
- Nonce 等链上资料

### 小学生版

- World State = 全班总成绩单
- 每次考试后都会更新

## Storage 属于哪一种状态？

### 重点题

Storage 属于： World State

### Memory vs Storage

- **Memory**: Machine State (临时)
- **Storage**: World State (永久)

### 小学生口诀

- Memory: 下课擦掉
- Storage: 写进联络簿

## 交易执行时发生了什么？

### 完整流程（超级重要）

#### 一步一步发生的事

- 从旧的 World State 开始
- EVM 建立一个新的 Machine State
- 用 OPCODE 执行交易
- 计算结束后：
  - 改动写回 World State
  - Machine State 丢弃

### 小学生流程

- 拿旧成绩单
- 用草稿纸算
- 更新新成绩单
- 草稿纸丢掉

## 为什么要这样设计？

### 核心原因

为了：安全、确定性、可验证

### 三大好处

- 不同节点算得一模一样
- 状态不会被偷偷改
- 节点只需存最终状态

### 如果没有区分会怎样？

- 链太大
- 状态混乱
- 无法验证

## 本章小学生总结

### 一页背完

- Machine State: 交易执行时的临时状态
- World State: 区块链真正保存的状态
- Memory: 临时的 (Machine)
- Storage: 永久的 (World)
- 算完就丢 Machine, 只留 World

教学提示：理解 EVM 状态的关键，在于「计算过程」和「最终结果」一定要分开。