

三亿人的自愿账单

用系统工程师的眼睛，拆开美国那台每年收上万亿的收税机器

王建硕

面向理工背景的好奇者

导读

用系统工程师的眼睛，拆开美国那台每年收上万亿的收税机器

先看一道题。你有三亿人，几百万家公司，一年要从他们手里收上万亿美元。规矩是：**没人上门收，你自己填张表，自己算欠多少，自己把钱寄过去。**

任何一个写过系统的人看到这里都该报警了。用户自己上报数据，自己算金额，自己决定交多少——这在工程上叫“客户端可信”，是安全课上第一节就被划掉的错误答案。它翻译成成人话是：把收银机搬到顾客家里，钥匙也给他，然后指望他每天准时把钱送来。

可这台机器真就这么运转着，而且几乎不出错。它甚至有个官方名字，叫 自愿遵从 (voluntary compliance) ——字面意思就是“你自愿配合”。

大白话

一个靠“你自愿”收钱的国家机器，收上来的钱多到能养一整个联邦政府，还常年准时。这不科学。工程师的直觉会立刻响：**“自愿”两个字底下，一定藏着不自愿的东西。**

这本书是干什么的

这不是一本报税手册。我不会教你抠哪一栏能省钱，也不会让你背税率表——那些东西每年都变，背了也白背。

我想干的是另一件事：**把这台机器拆开，看它到底怎么让“自愿”成立。**

拆开之后你会看到三根柱子，撑起了整个“不可能三角”：

- 源头预扣——钱在到你手里**之前**就被截走一部分。你从没摸到过那笔钱，也就无从赖账。
- 信息申报交叉核对——你的雇主、银行、券商，早就背着你把你挣了多少、赚了多少，一份份寄给了税务局。系统不必信你，因为它有别人替你作的证。
- 审计威慑——作弊被逮到的代价，乘以被逮到的概率，算下来是笔亏本买卖。于是大多数人自己把自己劝退了。

这三样东西，没有一样叫“信任”。合起来，却让三亿人心甘情愿地准时交钱。

那句你该记住的话

你以为你在报税，其实你只是在和它手里早就有的那份底稿对账。

每年春天你填的那张表，在你动笔之前，答案**基本上已经存在了**。你不是在申报，你是在核对——核对你填的数字，和它那份第三方替你作证的底稿，对不对得上。对得上，放行；对不上，机器自己就会给你寄信。

读完你能带走什么

两样。

一样是**看懂这台机器**——它天天从你身上收钱，你却大概从没看清它长什么样、靠什么运转。看清之后，你和它打交道的姿势会不一样。

另一样更值钱，是一套能搬走的思路。这三根柱子根本不是税务专利，它们是任何一个大型系统都在偷偷用的招：

- **预扣**=在钱的必经节点上截流，而不是等它跑光了再追。
- **信息申报**=用两个独立信息源对账，谁都不必被无条件信任。
- **威慑**=不追求抓住每一个人，只要让“作弊的期望收益为负”，多数人就会自己收手。

你要是做风控、做支付、做任何需要“在不信任的人群里维持秩序”的系统，这三招你迟早会用上。美国这台收税机器，只是把它们跑到了三亿人的规模上，跑了几十年。

那么，从哪儿开始

就那个最刺眼的悖论开始：一个不上门、靠你自己填表的系统，凭什么不崩？

翻开第一章，我们先把这道“不可能”的题，摆到台面上。

第一章 · 一台没人押着你交钱的机器

先说一件多数人没细想过的怪事。

美国联邦政府一年从纳税人手里收进来大约 **4 到 5 万亿美元**。这笔钱不是靠上门催的。没有几百万个税吏挨家挨户敲门，没人押着你去交钱。绝大部分情况下，是你自己坐下来，算清楚自己欠国家多少，然后主动把表填好、把钱寄过去。

更怪的是，这套制度官方的名字就叫自愿遵从（voluntary compliance）。一个靠“自愿”运转的收税系统，每年稳稳当当地抽走几万亿，还基本不出错。作为工程师，你听到“自愿”和“几万亿、不出错”放在一句话里，脑子里应该立刻亮红灯：这不对劲，这里面一定有机制被藏起来了。

“自愿”这个词，是个坑

第一件要拆穿的事：“**自愿**”不是字面意思。

它绝不是说“你想交就交、不想交就算了”。你要真这么理解，牢房在等你。这里的“自愿”是个法律术语，说的是**由谁来算这笔账**。

大白话

翻译成成人话：政府不主动跑来核定“你今年欠 8324 块”。它把“算账”这道工序外包给了你——你自己统计收入、自己扣减、自己算出该交多少、自己填表申报。政府默认你会老实算、老实交。这个“你自己算自己报”的模式，就叫自愿遵从。

对比一下你就懂了。世界上很多国家是政府核定制（也叫官方估税）——税务局先替你吧账算好，直接给你寄一张“你欠这么多”的账单，你照着交。那种模式里，算账的责任在政府。美国反过来，算账的责任在你。

用软件的话说：核定制是**服务端**算好结果推给客户端；自愿遵从是**客户端**自己算好结果提交给服务端，服务端只做抽查校验。美国把几亿次“算账”这个计算任务，分摊到了每一个纳税

人的头上。这不是偷懒，这是一个刻意的架构选择——服务端根本扛不住给三亿人逐个算账的负载，于是把计算下推到了边缘节点。

把它当一台分布式系统来看

一旦你接受“算账责任在客户端”这个设定，整个问题的形状就变了。它不再是一个税务问题，而是一个**分布式系统**问题。

把规模摆出来感受一下：

- 总人口约 **3.4 亿**，其中每年提交个人所得税申报表的，约 **1.6 亿个申报节点**（一对夫妻联合申报只算一个节点，所以申报数少于人口数）；
- 几百万家公司、合伙企业、雇主（机构节点）；
- IRS 每年要处理的，就是这大约 **1.6 亿份**个人所得税申报；
- 目标：从这几亿个节点，每年收上来准确的**数据**（谁挣了多少、扣了多少）和真金白银的**钱**；
- 硬约束：不能派几百万税吏上门，人力成本上根本不允许。

这就是一道教科书级的分布式系统难题：**几亿个不受你直接控制、还各有各的动机去少报的节点，如何从它们身上收集到准确的状态，并完成结算？**

大白话

说白了，IRS 面对的场景，跟一个超大规模后端要跟三亿个客户端“对账”没本质区别——每个客户端都自己上报一份数据，服务端要判断这份数据可不可信、有没有漏、有没有作弊，还要在几乎不逐个人工核查的前提下，保证整体账目是对的。

而且这些节点里有一批是**会撒谎的节点**。分布式系统里我们管这个叫拜占庭故障——节点不只是宕机或掉线那种“老实的坏”，而是会主动发送虚假信息、伪造状态。少报收入、多报扣除，就是纳税人这一侧的拜占庭行为。一个只会处理“诚实节点偶尔出错”的系统很好设计；一个要在**相当一部分节点有动机骗你**的前提下还保持整体正确，才是真正难的地方。

那它到底崩没崩？

如果这套“自愿+客户端自己算”的架构真有这么脆，按理说早该被薅穿了。可数据不支持“崩了”这个结论。

历史上，美国的自愿遵从率（voluntary compliance rate，指在没被强制追缴之前、纳税人自觉、按时、足额交上来的比例）长期稳定在**大约 83% 到 85%**。也就是说，应收的税里，八成多是大家自己老老实实、准时交上来的——没人押着，没上门催，就交了。

剩下那没交上来的一块，有个专门的名字叫税收缺口（tax gap）——就是“该收的”减去“实际自觉交上来的”那个差额。它确实存在，规模也不小，但注意：它是个**缺口**，是系统的漏损率，而不是系统的崩溃。八成多的自觉率，意味着这台机器的主干在稳稳地跑。

一个不上门、不预先给你算账、还允许你自己填表的系统，凭什么能让三亿人里的八成，年复一年主动把钱交对、交齐、交准时？

这就是这本书要拆的核心悖论。答案不是“美国人特别自觉”这种空话——靠道德是撑不起 85% 的合规率的。答案是：**这套系统在你没注意的地方，偷偷设计了三道机制，把“自愿”变成了几乎不可能作弊的结构**。所谓“自愿”，只是站在你视角看到的表象；系统内部，处处是让你没法赖账的暗桩。

三根柱子，先埋个头

后面的章节会一根一根拆，这里先让你知道它们的名字，好带着问题往下读：

1. **源头预扣**——在钱到你手里之前就先扣走一部分。等你拿到工资，税其实已经交过了。这一步把“要不要老实交”这个选择，从很多人手里直接拿掉了。
2. **信息申报交叉核对**——所有给你发过钱的第三方（雇主、银行、券商）都会背着你，向 IRS 递一份“我给了这个人多少钱”的小报告。于是 IRS 手里早有一份你的收入底稿。你申报时若和这份底稿对不上，机器立刻就看出来了。这是典型的**多副本对账**：同一笔收入，至少有两个独立来源上报，服务端一比就知道你有没有漏。
3. **审计威慑**——真正被查账的每年只有百分之几。但关键不在查多少，而在于它让作弊的**期望收益变成负数**：赌赢了省点小钱，赌输了连本带利加罚款加可能的刑责。理性人算完这笔

账，最优解就是老实交。

大白话

这三根柱子合起来干的事，其实很像分布式系统里保证数据可信的老套路：一是在数据产生的源头就把它固定下来（预扣）；二是让同一份数据有多个独立副本可以互相印证（信息申报对账）；三是用抽查加重罚，把“造假的期望代价”抬到高于“造假的期望收益”，让作弊在数学上就不划算（审计威慑）。

所以回到开头那个红灯：当“自愿”和“几万亿、不出错”出现在同一句话里，被藏起来的机制，就是这三根柱子。它们让绝大多数人**连作弊的机会和动机都没有**，剩下少数想赌的，也被威慑摠住。“自愿”两个字，是这台机器留给你的、精心设计的错觉。

接下来，我们一根一根把柱子拆开，看它们到底是怎么撑住这几万亿的。第一根，从你还没摸到工资就已经少了一块的那笔钱说起——**源头预扣**。

第二章 · 杀手锏：钱还没到你手里就被扣走了

先讲个反常识的事实

你问一个美国上班族：“今年交了多少联邦所得税？”十有八九答不上来。他能告诉你月薪多少、房租多少、Costco 一趟花了多少，但税——不知道。

这不是因为他懒或者笨。是因为**他从来没有亲手交过这笔钱**。钱在到他手里之前，就已经被扣走了。

这就是这台收税机器的第一根支柱，也是最狠的一招。工程上它有个朴素的名字。

源头预扣 (withholding)：雇主在给你发工资之前，先按估算替你把该交的联邦所得税扣下来，直接汇给国税局 (IRS)。你银行账户里到账的，是**税后**那一份。

大白话

说人话：你的工资单上写着月薪一万，但打到卡里的可能只有七千五。那两千五没进你口袋绕一圈再上交——它压根没经过你的手，直接从公司走到了政府。

为什么这一招是“杀手锏”

想象你在做一个电商系统，要向用户收一笔服务费。有两种收法：

- **方案 A (先花后交)**：让用户先痛快消费，月底给他发一张账单，指望他自己登录、掏钱、付清。
- **方案 B (网关截流)**：在支付网关那一层，用户每笔钱经过时，你直接把服务费扣下来，剩下的才放行到他账户。

做过收款的人都知道方案 A 有多恐怖：总有人拖、有人赖、有人月底真的拿不出钱。你得催收、得对账、得打官司。而方案 B 呢？用户根本没有“不付”的机会——钱流经你控制的节点，你在那儿截一刀就完事了。

预扣就是方案 B。而且它选的截流点特别妙：**雇主**。

雇主：一个被征用的免费收银台

全国三亿人里，绝大多数人的收入都来自一个共同结构——有个雇主，按月或按两周发工资。这个“雇主发钱”的动作，是一个所有工薪族都必经的节点。

政府干了件很聪明的事：它不去追三亿个纳税人，而是**让几百万家雇主替它当收银台**。雇主发工资时顺手把税扣了、汇给 IRS，还得填表报备。这活儿雇主一分钱报酬没有，纯义务，做不好还要罚。

大白话

相当于政府把收税这个苦活儿，外包给了公司的财务部门，而且不付外包费。你老板的 HR 系统，其实是 IRS 免费的分布式代收终端。

这里藏着整本书最关键的一个工程思想，值得单独拎出来：

这一招把一个**信任问题**，改成了一个**结构问题**。

“先花后交”依赖的是纳税人的自觉——你信不信三亿人到期都乖乖掏钱？不敢信。而预扣不需要任何人自觉，它靠的是钱流的物理结构：钱必须经过雇主这个闸门，闸门上装了个扣款阀。**自觉与否根本不进入这个方程。**

软件工程师对这个套路应该很熟：与其在业务逻辑里写一堆校验去“信任”上游传来的数据，不如在架构上让非法状态根本无法构造出来。让坏事在结构上不可能发生，永远比事后检查靠谱。

这机器是二战时装上去的

预扣不是天生就有的。1943 年之前，美国人交联邦所得税是“先花后交”的方案 A：这一年挣的钱，等到**第二年**一次性算总账、写张大支票交上去。

问题很直接：钱早花光了。一个人一整年没攒下税钱，到了第二年四月要一次性掏出一大笔，很多人根本拿不出来。欠税、追缴、拖欠成了常态。

然后二战来了。战争是个吞钱的无底洞，政府需要**快速、稳定、不断流**的现金流去买军舰和子弹。等一年后收一次，太慢也太不靠谱。

于是 1943 年的 《当期纳税法》（Current Tax Payment Act） 出台，把交税从“第二年算总账”改成了“当期、随发工资同步扣缴”。你一个月挣钱，这个月的税这个月就被扣走、汇给政府。现金流从一年一次的脉冲，变成了每两周一次的稳定水流。

大白话

“Current”就是“当期、实时”的意思。以前是批处理——攒一年数据到年底跑一次。改完之后是流式处理——钱一产生，税就实时结算了。

一个耐人寻味的名字

设计这套预扣机制的人里，有一个后来大名鼎鼎的经济学家：米尔顿·弗里德曼（Milton Friedman）——那个一辈子主张小政府、反对政府乱花钱的自由市场旗手。

年轻时他在战时财政部工作，参与把预扣制设计出来、让政府能高效收税。到了晚年，他公开表达过后悔。因为他亲手帮政府造的这台机器实在太好用、太“无痛”了，反而让政府扩张收税变得毫无阻力。

这个反差本身就说明了预扣有多厉害：连它的设计者、一个天然警惕政府的人，都承认自己造出了一个过于顺滑的抽水泵。

“无痛”是这台机器的润滑剂

现在说说预扣最精妙、也最政治性的一面：**它消除了交税的痛感。**

心理学上有个简单的事实：花已经到手的钱，很疼；从没到过手的钱，不疼。你要是每个月月底亲手给政府转两千五，你会疼得龇牙咧嘴，会关心每一分钱花哪去了，会上街抗议。但如果这两千五你从没见过，工资单上一行小字扣掉了，你连眉头都不皱一下。

更妙的还在后头。因为雇主扣的只是**估算**——它不可能精确知道你全年到底该交多少——所以往往会稍微多扣一点。到了报税季一算账，多扣的部分退还给你，这就是大家喜闻乐见的退税（tax refund）。

很多人一年到头就盼着这笔退税，拿到手像中了小奖，高高兴兴去买个新手机。但你冷静想想这笔钱到底是什么：

大白话

退税不是政府发给你的红包。是你自己一整年被多扣走的钱，被政府无息借用了大半年，年底原样还给你。你不但没占便宜，还倒贴了这笔钱这一年的利息和机会成本——却还满心欢喜地感谢它。

你品品这个设计有多绝：一个从你身上抽钱的系统，硬是被设计成让你到点盼着它给你“发钱”。掏钱的痛感被抹掉了，还倒装出一点占便宜的甜头。这就是为什么这台机器在政治上能几十年稳定运转——大多数人根本感觉不到自己在被收税，甚至对它有好感。

那扣多少，谁说了算？

雇主怎么知道该替你扣多少？靠你入职时填的一张表：W-4 表。

你在 W-4 上告诉雇主一些信息——比如你有没有要抚养的孩子、是不是已婚、有没有别的收入。雇主拿这些信息去估算你的全年税负，再折算成每次发薪该扣多少。

- W-4 填得让扣得**多**：平时到手少一点，年底大概率退税（你借给了政府）。
- W-4 填得让扣得**少**：平时到手多一点，年底可能要补交（政府来找你补齐）。

这只是个预估旋钮，拧大拧小影响的是年底多退还是多补。真正的年终结算是怎么对账的，是后面第 9 章的事，这里点到为止。

效果：几乎无从作弊

预扣的威力，最后落在一个数字上：工资类收入的税收**遵从率高达约 99%**——IRS 的数据显示，有预扣、又有第三方申报盯着的工资收入，错报率只有 1% 上下。作为对比，后面会讲到的自雇 / 个体经营收入，错报率能高到一半左右。

原因不难理解——你的工资根本没经过你的手，扣税这个动作由雇主完成、由雇主上报，你既没机会藏、也没机会赖。想少交？除非你和雇主合谋做假账，而雇主犯不着为你担这个法律风险。作弊空间被结构性地压没了。

这不是因为工薪族比别人更诚实，而是因为他们压根没有不诚实的机会。系统没给他们留作弊的接口。

但你肯定已经想到一个漏洞了：**那些不发工资的钱怎么办？**

自己开公司的、接私活收现金的、靠炒股分红过活的——这些收入没有一个“雇主”节点去截流，预扣这张网罩不到它们。对这部分人来说，钱是实打实先到了自己手里的，“先花后交”的诱惑又回来了。

系统怎么补上这个大窟窿？那就是下一根支柱要干的事——当政府没法在源头截流时，它换了一种玩法：让所有给你付钱的人，都替它盯着。这，是第三章的故事。

第三章 · 那些没被预扣的钱怎么办

上一章我们把**预扣**这根柱子立起来了：钱还没到你手里，雇主就先替政府截走一刀。这一章要问一个工程师听到就该警觉的问题——**这根柱子能截到的，到底是哪些钱？**

答案很朴素：只有**经过雇主这个节点**的钱。预扣的本质是“在必经之路上设卡”。你的工资每个月都要从公司账户流到你账户，这条路是固定的、可拦截的，于是政府在这里蹲点，一分不落。

大白话

像高速公路收费站。只要车必须走这条路，收费站就能收到钱。问题是——不是所有的车都上高速。

哪些钱不走雇主这个收费站

只要一笔收入不经过“雇主发工资”这个动作，预扣就够不着它。数一数漏出来的都有谁：

- **自雇 / 自由职业收入**：self-employment，就是你不是给某个老板打工领工资，而是自己就是那个收钱的人——freelancer 接活、开公司的老板给自己发钱、Uber 司机、送外卖的、小店主。钱是客户或平台直接付给你的，中间没有一个“雇主”帮你扣税。
- **投资收入**：银行给的利息、股票发的分红（dividend，公司把利润分一部分给股东）、你低买高卖赚的资本利得（capital gain，卖掉资产赚的差价）。这些钱从券商、银行流进来，也不经过任何雇主。
- **租金**：你有套房出租，房客把租金直接打给你。
- **现金交易**：餐厅服务员收的现金小费、小生意收的现金、街边摊——钱以现金形态直接进了口袋。

用工程语言说：预扣的**覆盖率**只有工资这一块。剩下这些收入，是收费站够不着的“匝道”。系统要不漏，就得给这些匝道打补丁。

补丁一：让你自己当自己的收银台

第一个补丁叫季度预缴 estimated tax。没人替你扣，那就你自己扣——一年分四次，把估算出来的税提前交给政府。

时间点大约是当年的 4 月、6 月、9 月，加上次年 1 月。你得自己估：这个季度大概赚了多少、该交多少税，然后主动打钱过去。等于你亲手给自己开了一个代扣收银台，扮演了本该由雇主扮演的那个角色。

大白话

打工的人，税是雇主每月替你“零存”进国库的。自雇的人没这个待遇，只能自己定四个闹钟，自己往里塞钱。

为什么非得分季度，不能年底一把交清？

因为政府要的是**现金流**，不是**年底的一笔巨款**。这和上一章预扣的逻辑是一条心：国家的开销天天在发生，工资阶层是每月即时供血，自雇的人凭什么可以攒到年底再给？那对现金流不公平，也给了你一整年拿着这笔钱去周转的便宜。

所以规则是：你得**边赚边交**，节奏尽量跟预扣对齐。少交或晚交，会吃 penalty（罚息）——本质就是你占用了这笔本该按时上缴的钱，得付利息。它不是道德惩罚，是给“延迟供血”标了个价。

补丁二：截不到，就想办法看得到

这里是全章最关键的一步，也是理解整台机器的枢纽。

回头看投资收入。它没被预扣，那它是不是就漏了？**没有**。因为券商、银行会主动把你这一年的利息、分红、卖股票的收益，用一张叫 1099 的表格**报给 IRS**——这就是信息申报（information reporting，第三方替你向税务局报一份账）。

预扣是“当场把钱扣下来”；信息申报是“钱先给你，但有人在旁边记了一笔账，账单同时寄给税务局”。你到年底自己报的数，得跟这份账对得上。

看出门道了吗？预扣和信息申报是**两根不同的柱子，干的是同一件事——让这笔钱在系统里“可见”**。工资靠预扣被看见，投资收入靠 1099 被看见。截流的方式变了，但那笔钱始终在系统的视野里没跑掉。

用可观测性（observability）的话讲：预扣是“拦截并记录”，信息申报是“不拦截、但记录”。只要有记录，你报的数就能被**交叉核对**——对不上，系统立刻知道。

那真正的黑洞在哪

把两根柱子叠一块，风险敞口就浮出来了。一笔钱的处境无非四种：

1. 被预扣了（工资）——看得死死的。
2. 没预扣，但有 1099 信息申报（利息、分红、平台付的钱）——照样看得见。
3. 没预扣，也没任何第三方替你报账——**这才是系统的盲区**。

第三种是什么？典型就是**现金收入**：现金小费、纯现金的小生意、地下经济。没有雇主扣，没有券商报，钱进口袋不留痕。要不要如实报，全凭你一句话——系统没有第二个信源来对质。

这就是那个反复出现的工程直觉：**系统能看见的地方，没人敢作弊；看不见的地方，才是黑洞**。

税收缺口：黑洞有多黑

把没收上来的税加总，有个专门的词叫**税收缺口**（tax gap，应收却没收上来的那部分）。工程师最该关心的不是它多大，而是它**分布在哪**——而它的分布，几乎是对上面那套逻辑的完美验证。

缺口的绝大部分，**不在工资阶层**。恰恰相反：

- 有第三方信息申报的**工资类收入**，错报率低到个位数百分比——因为它被 W-2、1099 盯得死死的，你想虚报都对不上账，没人愿意冒这个险。
- 而**缺乏信息申报的自雇 / 现金收入**，错报率能高到 **50% 的量级**——一半的钱可能报得不老实。

大白话

同样一个纳税人，工资那部分老老实实，自己接私活收的现金那部分就可能“忘了”报一半。不是人变坏了，是这两块钱一块在探照灯下，一块在黑影里。

这个巨大的落差，把整台机器的设计哲学摊在了桌面上：**逃税不是道德问题，是可观测性问题**。光照到的地方，合规率自动拉满；光照不到的地方，人性就开始试探边界。所以这台机器真正的工程目标，从来不是“抓坏人”，而是**尽量减少盲区**——把每一笔钱都拽进某根柱子的视野里。

埋下一章的线

到这儿，第二根柱子已经露头了：**信息申报**——那些 1099、W-2 表格，就是系统用来“看见”没被预扣的钱的眼睛。

可问题才刚开始：这些表格到底是谁、在什么时候、按什么规则报上去的？IRS 拿到成千上万张 1099，又是怎么跟你自己报的那份数**逐条对上**、一旦对不上就报警的？这套交叉核对的机器怎么转，就是下一章的事。

第四章 · 所有人都在给你打小报告

写到这里，你已经知道了预扣有个大窟窿。第 3 章结尾我们看到：预扣只覆盖那些有雇主、有券商、有平台替你把钱截下来的场景。一旦收入是现金、是没人报账的私活，预扣就是瞎的。

那系统怎么补这个窟窿？答案不是“让你多报点良心账”。系统根本不打算相信你。它另有一手：**它手里早就有一份不是你写的底稿。**

系统不问你挣了多少，它问别人

先说一件反直觉的事。你以为报税是“你告诉 IRS 你挣了多少”。不完全是。真正的主力信息，不是你报的，是**付钱给你的人报的**。

法律规定：任何一方付了你一笔够得着门槛的钱，就有义务向 IRS 打一份报告，说“我今年付了张三这么多钱”。你雇主、你银行、你券商、你接私活的甲方、收你钱的支付平台——全都是这个“打小报告”网络里的节点。

大白话

说人话：不是让你自己交收入，是让每一个给过你钱的人，背着你（其实也告诉你）向税务局递条子。你收了多少钱，税务局从别人嘴里就已经知道了。

两张核心表：W-2 和 1099

这些“条子”是有标准格式的表。你一辈子会反复见到两张。

W-2：雇主替你交代工资

W-2 是雇主每年给正式员工出的一张表。上面写着两件事：今年一共发了你多少工资，以及已经替你预扣了多少税（还记得第 2、3 章的预扣吗，就是那笔提前截走的钱）。

关键在于它**一式多份**：一份给你，一份给 IRS（联邦），一份给州税务局。同一个数字，同时躺在三个地方。你手上那份，和 IRS 手上那份，是从同一台打印机里吐出来的。

1099：一大家族的“非工资”表

1099 不是一张表，是一整个姓“1099”的家族。凡是不属于“雇主发工资”的收入，基本都有一张对应的 1099。常见的几个成员：

- 1099-NEC ——你接私活、做自由职业挣的报酬（NEC = 非雇员报酬）。这张就是专门用来照亮第 3 章那个“私活盲区”的。
- 1099-INT ——银行付给你的利息。
- 1099-DIV ——股票发给你的分红。
- 1099-B ——券商替你记录的卖出交易（你卖了股票赚了多少钱赔了多少钱）。
- 1099-K ——第三方支付平台（比如收款 App）替你汇总的收款额。

家族很大，但逻辑只有一条：**只要有第三方付钱给你，就有一张表在替这笔钱作证**。而且和 W-2 一样，1099 也是 IRS 和你各收一份——你收到的那张，IRS 早就收到了同一张。

工程本质：这是一套双源对账系统

现在把眼睛眯起来，从系统的高度看这件事，你会看到一个软件工程师再熟悉不过的结构。

每一笔收入，系统里都有**两条独立来源的记录**：

1. 付款方报的：“我付了张三 X 元。”
2. 收款方（你）报的：“我收了 X 元。”

这两条来自完全不同的两方，必须对得上。这在工程里有好几个名字，本质是同一件事。

大白话

复式记账 (double-entry bookkeeping)：账本里每一笔钱都记两次，一次记在付出方，一次记在收入方，两边必须平。谁想凭空捏造，另一边就对不上。

对账 (reconciliation)：分布式系统里两台机器各自记了一份流水，定期拉出来逐笔比，不一致就报警。

冗余校验：同一份数据存两个独立副本，用一个去校验另一个，防的就是单点造假或出错。

三个说法，一个内核：**同一份事实，被两个互不商量的来源各记一遍，然后互相印证。**

为什么这招狠：它把“信任”从系统里删掉了

普通的“你自己报”系统，命门是它必须**相信你**。你少报，它一时看不出来。信任是系统里最脆弱的一环——因为它是个单点，捅穿这一点，整个防线就塌了。

双源对账把这个单点干掉了。系统不再需要相信你，因为它有一份不是你写的底稿。工程师管这叫去信任 (trustless)：不是假设大家都诚实，而是把系统设计成——**就算你不诚实，也占不到便宜。**

想作弊，你得让两端同时说谎且对得上：既要甲方不给你出 1099，又要你自己不报，还得赌 IRS 那份也恰好是空的。可只要付款方按法律出了表，你这端就已经被钉死了——IRS 手里那份底稿，压根不问你同不同意。作弊的成本，被抬到了几乎不可能。

大白话

说白了：以前是“我说我挣了多少”，现在是“我来确认一下你手里那份数字对不对”。你以为你在申报，其实你在**对账**。呼应导读那句钩子——你只是在和它早就攥在手里的那份底稿对账。

盲区的真正原因，藏在“谁替谁报”里

这套机制强大，但有个内在的边界，正好解释了第 3 章那个现金盲区。

整个系统的燃料，是**第三方付款方生成的那张表**。有付款方、有表，系统就有底稿，就看得见，就能对账。可要是一笔收入根本没有第三方付款方——纯现金、私下交易、没人给你出

1099——那系统这一侧就是空的，没有底稿可对。

这不是系统偷懒，是它的结构决定的：**信息申报覆盖到哪，遵从率就高到哪**。有 W-2 的工资收入，几乎没人敢少报，因为底稿在那儿明晃晃摆着；纯现金收入的遵从率，则低得多——不是人变坏了，是那片区域系统本来就没长眼睛。

时间线：为什么报税季你俩几乎同时握着同一份纸

这里有个容易忽略但很妙的设计。这些 W-2、1099，法律要求通常在**次年 1 月底前**发出。而报税季也在次年初。

所以到了你坐下来报税那会儿，你手上刚收到这批表，IRS 手上也刚收到同一批表——**你俩几乎同时握着同一份底稿**。你填的那些数字，本质上是在照抄一份 IRS 也拿到了的稿子。这不是巧合，是系统故意让“你的申报”和“它的底稿”在同一时间点对齐，方便接下来做那件事：比对。

下一步：几十亿份表，怎么和一亿多份申报对上？

到这里，第二根柱子立住了：系统靠双源印证，把“相信你”工程化地换成了“不需要相信你”。

但一个尖锐的问题冒出来了。IRS 每年收到的第三方表，是**几十亿份**；个人申报，是一亿多份。这么两座数据山，怎么逐行比对、揪出对不上的那些？总不能靠人一张张翻。

答案是：靠机器，自动匹配。这套系统有个名字，叫 AUR——它就是下一章的主角。

第五章 · IRS 其实是一座对账工厂

先做个小测验。想象一下“美国国税局查你的税”这个画面——你脑子里浮现的是什么？

多半是这样：一间堆满档案的办公室，一个戴袖套的公务员，把你的报税表铺开，拿放大镜一行一行看，眉头一皱，“这个人有问题”。像警察查案，像老师批卷子，一份一份地过。

这个画面几乎全错。错在哪？错在你以为“查”是一件**由人来做**的事。

先算一笔账：人根本查不过来

美国每年有一亿多份个人报税表。跟这些表对应的第三方底稿（就是第 4 章讲的那些——你雇主报的 W-2、银行报的利息 1099、券商报的卖股票 1099），加起来是几十亿份。

现在你当一回 IRS 的运营总监。你手头有几万名员工，要用他们去核对“一亿多份申报 vs 几十亿份底稿”。假设一个熟练的人一天能认真核对 20 份表，一年 250 个工作日——一个人一年 5000 份。要覆盖一亿份，你需要两万人干整整一年，而且这还只是“看一眼”，不是深查。

结论很直白：**如果核查靠人，这件事在数学上就是做不到的**。所以它一定不是靠人做的。这就是理解 IRS 的钥匙——先承认人力这条路走不通，你才会去猜真正的机制是什么。

真相：IRS 是一座对账工厂

它真正的样子，不是警察局，是一座**数据处理工厂**。绝大多数“核查”里，没有任何一个人类看过你的表。看你的表的，是一段程序。

这段程序有个名字，叫 AUR（Automated Underreporter，直译“自动少报侦测程序”）。名字听着唬人，干的事其实特别朴素：**把你申报表上的每一个数字，和第三方报上来的底稿逐行比一遍**。

说人话，它就是在跑一句断言：

```
assert(你自己报的收入 == 第三方替你报的收入)
```

对得上，放过；对不上，标红。就这么简单。

你是工程师，这个画面你太熟了。它就是：

- Git 里的 `diff` ——把两个版本摆一起，逐行找出哪里不一样；
- 单元测试里的 `assert` 断言——预期值对不上实际值，测试就红；
- 数据库的**外键约束**——你插一条引用了不存在记录的数据，数据库直接拒绝，因为它对不上主表；
- 财务半夜跑的**对账脚本**——把这边的流水和那边的流水拉齐，把差异行吐到一张报表里。

AUR 就是 IRS 那台半夜跑批的对账脚本。它不知疲倦，不吃午饭，不会看到大人物的名字就手软。你报利息收入 200 块，银行报上来说付了你 800 块——diff 出来一行差异 600，标红。你压根忘了申报的那张券商 1099，在你的表里根本没有对应行——外键对不上，标红。

标红之后：那封著名的信

这里要澄清一个关键的误解，也是很多人怕 IRS 怕错了地方的原因。

AUR 匹配出差异之后，**发生的不是“审计”，而是自动生成一封信**。这封信有个编号，叫 CP2000 通知——它是 IRS 最有名的一封信，本质是机器打印出来的一张差异报表，寄到你家信箱。

“我们的记录显示，你 2024 年少报了一笔 600 美元的利息收入。根据我们的计算，你需要补税 X 元。如果你同意，请缴款；如果你不同意，请解释。”

翻译成工程语言：这是一封**自动化的差异告警邮件**。就像你的监控系统半夜发现指标对不上，自动给你发了封告警——发信这个动作里，没有人参与。整条流水线是：底稿入库 → 跑批比对 → 命中差异 → 模板套差异数字 → 打印 → 寄出。从头到尾，可能没有一个 IRS 的人类看过你的名字。

所以，很多人收到 CP2000 吓得半死，以为“我被 IRS 盯上了、要被审计了”。不是的。你不是被人盯上，你是被一段脚本 `diff` 了一下，命中了一行。这跟“审计 (audit)”是两码事——审计是有人真的坐下来深挖你的账，那要稀有得多、也贵得多，是下一章的主题。

量级感：这台机器一年吐多少封信

给你一点数量级的感觉（记住是量级，别当精确数）：

- IRS 每年发出的这类**自动通知，是以百万计的**——每年几百万封 CP2000 这样的机器信，稳定地从工厂里流出来。
- 而真正由人上门、坐下来深查的**外勤审计，只占全部申报的百分之零点几**。绝大多数纳税人一辈子都碰不上一次真人审计。

把这两个数摆一起，你就看懂了整个系统的设计意图：**能用机器批量对账解决的，绝不动用一个人**。人是最贵、最不可复制的资源，留给机器搞不定的硬骨头。

工程洞见：从 $O(\text{人力})$ 降到 $O(\text{算力})$

如果这一章你只带走一句话，是这句：

这套系统的本质，是把“抓错”这件事，从 $O(\text{人力})$ 降到了 $O(\text{算力})$ 。

大白话

拆开说：

人工审计是 $O(\text{人力})$ ——你想多查一倍的表，就得多招一倍的人，成本随覆盖率线性甚至超线性上涨，而且人会累、会错、根本铺不满一亿份。这条路天生**不可规模化**。

自动匹配是 $O(\text{算力})$ ——多跑一亿份，无非是多买点服务器、多跑几个钟头批处理。边际成本低到可以忽略。于是覆盖率可以一路拉满，**几乎每一份“有第三方底稿可对”的申报，都被机器比过一遍。**

这才是 IRS 真正的威慑来源，而且它跟大众的想象正好相反。真正让你不敢乱报的，不是“我们可能会派个人来查你”——那概率低到像中彩票。真正的威慑是：**机器已经把你和底稿比过了，而且是每一份都比。你逃不过 diff。**

威慑不来自“抓的力度大”，而来自“抓的覆盖率接近 100%”。一个低概率的重罚，和一个近乎必然的对账，后者对老实人的塑造力强得多。

但这台机器有个天生的盲区

现在，聪明的你应该已经嗅到破绽了。

AUR 这套魔法，全部建立在一个前提上：**存在一份第三方底稿，能拿来和你的申报做 diff。** 比对的本质是“两个来源对齐”，那要是根本只有一个来源呢？

回想第 3、4 章反复敲的那个词：现金。你收的一笔现金收入，没有雇主替你报 W-2，没有平台替你报 1099，第三方那一栏是**空的**。这时候 AUR 拿什么跟你 diff？它手里只有你自己报的那个数，没有第二个数来对齐。

大白话

用外键的比方说：AUR 能抓的，是“你引用了一条和主表对不上的记录”。但如果这条收入**压根没进过任何主表**——没人替你登记过——那就没有约束可以校验，机器连“这里应该有一行”都不知道。它不是漏抓，是根本瞎了。

所以这台对账工厂再强，也只能覆盖“有底稿可对”的那部分世界。对于机器 diff 不出来、却又透着可疑的申报——比如收入和生活方式明显对不上、但账面上找不到具体差异行——就

得动用那件最贵、最稀缺、也最让人害怕的武器：**真人审计**。

机器负责用近乎零成本覆盖所有能对账的部分；人负责在机器失灵的地方，做昂贵而稀有的深挖。这两者怎么分工、审计的概率为什么低到那种程度反而更划算——那是一道漂亮的经济学题目，我们下一章接着拆。

第六章 · 抽查与威慑：为什么你不敢赌

先给你一个数字，你大概会愣一下。

IRS（美国国税局）每年真正动手审计 audit——也就是派人翻你的账、要你的凭证、逐条核对你有没有少报——的比例，低到吓人。全体个人申报里，大概只有**千分之三到千分之五**被审。近几年因为预算被砍，这个数还更低。你没看错：一百个报税的人里，被真人查账的连一个都不到。

那问题就来了。既然 99% 以上的人这辈子都不会被查，为什么大家还老老实实交税？换成你写代码：如果你知道某个校验分支 99% 的请求根本不会走进去，你还会那么认真对待它吗？

这一章讲的就是这台收税机器的第三根柱子——也是最反直觉的一根：**威慑**。前面第五章讲的 AUR 是机器自动比对底稿、几乎必抓有据可查的差异；这一章讲的是稀缺得多的东西——真人审计。它靠的不是“抓住每个人”，而是“让作弊这件事，算完账根本不划算”。

把“要不要作弊”当成一道期望值题

软件工程师最熟的思考工具之一就是期望值 EV (expected value)：把每种可能结果的收益，乘上它发生的概率，加起来。赌不赌，看这个加权平均是正是负。

现在用它来算“要不要少报一笔收入”：

作弊的期望收益 = (逃掉的概率 × 省下的税) - (被抓的概率 × (补税 + 罚款 + 利息 + 可能的刑责))

大白话

就是：赢了能省多少 × 赢的概率，减去 输了要赔多少 × 输的概率。跟你评估一个高风险优化值不值得上线，是同一个算式。

机器设计者的全部功夫，就是想尽办法把这个式子的结果**压到负数**。一旦它是负的，一个理性人算完，自己就不作弊了——不需要谁盯着他。这不是我拍脑袋的说法，经济学里有个经典模型叫 Allingham-Sandmo 逃税模型（两位经济学家 1972 年提出的），把逃税当成一场赌博来建模，结论正是这个：**决定你诚不诚实的，很大程度上不是道德，是赔率**。名字提一句就够了，数学不展开。

三根杠杆：怎么把这个期望值压成负的

光有个式子没用。真正让几亿人乖乖的，是三个把赔率往作弊者不利方向掰的杠杆。

杠杆一：罚则不对称——赢一点，输很惨

你少报省下的是小钱。可一旦被抓，账单是这样的：先把税补上，再加**利息**（从当年欠税那天起算），再加一笔**罚款**——常见是少报金额的 20%，认定是欺诈能到 **75%**。情节严重的逃税，是**刑事**，会坐牢。

看清楚这个结构：赢，你赢一点点；输，你连本带利再加大几成，甚至进监狱。这是**故意设计的不对称赔率**。就像一个期望值本来接近零的赌局，庄家把“输”那一侧的赔付放大好几倍——哪怕被抓概率很低，一乘上巨大的损失，整个期望值就被拖到水下。工程上你也这么干过：给危险操作加一道极重的失败惩罚，让没人敢碰。

杠杆二：概率不是零，而且你算不准

千分之几听起来可以忽略。但注意两点。

第一，它**不是零**。第二，也是更狠的——**它不透明**。你不知道今年会不会轮到你，你也不知道自己的申报在系统眼里有多扎眼。再叠加上第五章的 AUR：凡是有第三方底稿（公司报给 IRS 的工资、券商报的收益）的差异，几乎是必被发现，那部分的“逃掉概率”直接趋近于零。

大白话

说人话：有据可查的地方你基本没有安全边界，没据可查的地方你又不知道自己会不会被抽中。这种“算不准”本身就是武器。

不确定性本身就是威慑。一个理性作弊者最想要的是**能精确算出自己的安全边界**——省多少、被抓概率多少、赔多少，全知道了才好下注。IRS 偏偏不给你这个透明度。你连自己在赌桌上的胜率都不知道，理性的做法就是别上桌。

杠杆三：追溯期——这不是一锤子买卖

你可能觉得，今年蒙混过关了，就安全了。不是。

- IRS 一般能**回查 3 年**。
- 如果你少报的金额比较大（漏掉的收入超过一定比例），能**回查 6 年**。
- 如果认定是**欺诈**或者你根本没报——**无限期**，永远可以翻旧账。

所以作弊不是“当年闯关成功就清零”，它是一颗**埋好几年的定时炸弹**。你今天省下的钱，未来好几年里随时可能连本带利被引爆。把时间维度加进期望值：被抓的概率不是“今年的抽查率”，而是“未来 3 到 6 年任意一年被翻出来”的累计概率——比你以为的高不少。

而且审计根本不是随机抽的

到这里你可能还想赌一把：千分之几嘛，运气问题。这是最大的误解。**审计不是抽奖，是精准打击**。

IRS 给每一份申报打一个“可疑分”。历史上这套东西叫 DIF 分 (Discriminant Function, 判别函数打分)——本质就是一个统计模型，看你这份申报和“正常人应该长什么样”偏离多远，偏得越离谱，分越高，越优先被人工审。

大白话

这玩意你天天都在写：它就是一个异常检测 anomaly detection / 风控模型。跟银行的反欺诈评分、跟邮箱给邮件打的垃圾分，是一个套路——不是逐个检查每一封，而是先算个分，把人力集中扑到最可疑的那一小撮上。

结果就是：审计概率在人群里**极度不均匀**。一个只有一份 W-2 工资单、什么都对得上的普通工薪族，被查概率低到几乎可以不计——反正也没什么可查的。而**高收入、申报复杂、扣除异常（比如收入十万却报了五万慈善捐赠）**的人，被查概率高出一大截。机器把最贵、最

稀缺的资源——真人审计员的工时——导向**期望产出最高**的目标：查一个高收入复杂申报，追回的税可能是查一百个工薪族的总和。

工程洞见：用一点点稀缺资源，撬动全局诚实

退一步看整台机器，你会看到一个非常漂亮的杠杆结构。

IRS 从来没打算查所有人，它也查不起。它要做的只有一件事：让每一个人都相信“被查的期望代价 > 作弊的期望收益”。只要这个信念成立，绝大多数人根本走不到作弊那一步——他们自己在脑子里算完就放弃了。

这就是威慑的美妙之处：**它是一种极其廉价的全局效应。**你只需要投入少量真实执法（真的查、真的罚、真的有人坐牢，让威胁可信），配上重罚（放大失败一侧）和不透明（让人算不准安全边界），就能把几亿人稳稳摁在诚实这一侧。真人审计像是压舱的那一小块铁——量不大，但整条船的平衡靠它。

反过来，这也解释了一件让很多人担心的事：为什么 IRS 预算被削、审计率往下掉，是个警报。因为**威慑是有弹性的**。它不是一个开关，而是一个赔率。你把执法这一侧抽掉一点，作弊的期望收益就往正数挪一点；挪过某个临界点，理性的算计结果就从“别赌”翻成“值得赌一把”。到那时遵从率不会缓慢下降，而是可能像堤坝渗水一样，先是几个人，然后一片人，一起滑坡。

所以别被那个千分之几骗了。这台机器真正的护城河，从来不是它抓住了多少人，而是它让剩下所有人算完账之后，自己选择不作弊。**最高效的执法，是那种你几乎不需要动用的执法。**

第七章 · 把税基一层层削出来

前六章我们讲的都是一件事：这台机器怎么**看见**你的每一笔钱。工资、利息、卖股票的收益，全都被各种上报单据钉死在案，你想瞒也瞒不住。好，现在假设机器已经把你今年挣的每一分钱都摆在桌上了。下一个问题来了——

看见了这么一大坨钱，到底该收你多少税？

这不是一步算出来的。它是一条**流水线**：钱从上游进来，中间被好几道工序一层层削，最后从下游滴出来的那一小滩，才是你真正要交的税。这一章，我们就把这条管道拆开，一个 stage 一个 stage 地看。

先把整条管道画出来

如果你写过数据处理的 pipeline，这张图会让你很舒服。数据从一头进，每一级做一次变换，交给下一级：

1. **毛收入**（入口）：你今年挣的全部。
2. 减去各种**扣除** → 得到**应税所得**。
3. 把应税所得**套上税率** → 算出一个**初步税额**。
4. 再用**抵免**去砍这个税额 → 得到**最终该交的税**。

请你先把这四级的顺序记牢，因为**顺序是硬的**——哪一步在前、哪一步在后，直接决定算出来的数对不对。后面我会专门讲为什么不能乱序。先从入口那个词说起。

入口：毛收入

毛收入（gross income）就是你一年挣到手的全部钱堆在一起：工资、银行利息、股票分红、卖东西赚的、租房收的……全算进来。

大白话

说人话：还没开始“减”任何东西之前，你账面上那个最大的数，就是毛收入。它是管道的进料口，后面每一道工序都在这个数上往下削。

注意：**毛收入不是税基**。没人会按你挣的全额来收税。真正被拿去乘税率的那个数，是被削过之后的“应税所得”。怎么削？靠下一级——扣除。

第一道工序：扣除，砍的是“被征税的钱”

扣除 (deduction, 也叫抵扣)：**从你的收入里减掉一块，让“被拿去算税的钱”变少**。它减的是**税基**，不是税额。

这句话是本章第一个坑，我们把它掰开。假设你挣了 8 万，系统允许你扣掉 1 万，那么真正进入下一级、要被套税率的，只有 7 万。那 1 万块，相当于“当作你没挣过”，从被征税的范围里被划掉了。

关键问题：这 1 万的扣除，到底帮你省了多少税？

不是 1 万。扣除省下的税，等于：

省下的税 = 扣除额 × 你的边际税率

举个具体的：你处在 22% 这一档，扣掉 1000 美元，你实际少交的税是 $1000 \times 22\% = 220$ 美元。不是 1000，是 220。剩下那 780，本来就不会因为这 1000 而被收走。

大白话

说人话：扣除不是直接从账单上划掉一笔钱，它是先把“要被打税的原价”砍下去一块，你省的只是这一块**本来会产生那点税**。你在的税档越低，同样的扣除就越不值钱。

第三道工序：抵免，砍的是“要交的税”

(先跳过中间那道“套税率”，因为它得等抵免登场才好对比。)

抵免 (credit, 税收抵免): **直接从你算出来的最终税额里减掉的钱——一块钱抵免, 顶一块钱的税。**

还是那 1000 美元。如果它是**抵免**而不是扣除, 你少交的税就是**整整 1000 美元**。跟你在 22% 还是 37% 哪一档, 一点关系都没有。它不碰税基, 它碰的是账单最底下那个“应付”数字, 直接减。

现在把两者摆一起, 本章最该钉死的一句话:

扣除减的是“被征税的钱”, 抵免减的是“要交的税”。

同样是名义上的 1000 美元, 抵免值 1000, 扣除只值 220 (22% 档下)。**抵免比等额的扣除值钱得多。**工程师最容易把这俩搞反, 记住这个类比就不会错了:

- **扣除 = 打折前先从原价里砍一块。**你砍的是“要被算钱的基数”, 最后省多少还得看税率这个“折扣比例”。
- **抵免 = 账单最后甩上来的一张代金券。**面值 1000 就抵 1000, 跟前面怎么算的没关系, 直接从应付金额里扣。

那道被跳过的中间工序: 套税率

回到管道中间。收入被扣除削成“应税所得”之后, 机器要把它**乘上税率**, 才得出一个初步税额——这就是抵免开砍之前的那个数。

你可能会问: 不就是乘个百分比吗? 没那么简单。这里藏着一个词叫**边际税率**, 也就是上面算扣除省税时用的那个 22%。它**不是**你收入的整体税率——这是全书最容易把人绕晕的地方, 我留到第八章专门拆。这里你只要先接受: 套税率这一步不是一刀切的乘法, 而且“扣除到底省多少”完全取决于这个边际税率。

扣除的默认值: 标准扣除 vs 逐项扣除

回头看“扣除”这道工序, 系统其实给了你两种用法, 很像软件里的**默认配置 vs 手动配置**。

标准扣除 (standard deduction) 是那个“懒人默认值”: 你什么都不用列, 系统直接允许你从税基里砍掉一个固定额度。2024 年, 单身大约 1.46 万美元, 已婚联合报税大约 2.9

万美元（记个量级就行）。一键生效，零成本。

逐项扣除（itemized deduction）是“手动配置”：你不用那个默认值，而是把自己一年里能扣的东西一笔笔列出来——房贷利息、交给州里的税、慈善捐款等等——加总起来当扣除。前提是你得留着凭证、一项项填。

规则很简单：**两个数谁大用谁**。你手动列出来的总额如果超过标准扣除，就用手动的；否则就吃那个默认值。

大白话

说人话：绝大多数人直接用标准扣除。因为这个默认值被系统**故意调得很大**——大到很多人辛辛苦苦列一堆凭证，加起来还没它高。于是对多数人来说，手动配置纯属白忙，默认配置又大又免费。这正是系统设计者想要的：让绝大多数人一键报税。

为什么顺序不能乱

现在把整条管道再走一遍，你就明白为什么这四级的顺序是**硬**的：

1. **先削税基**：从毛收入里减掉扣除（标准或逐项），得到应税所得。
2. **再套税率**：拿应税所得去乘税率，算出初步税额。
3. **最后砍税额**：用抵免直接从税额里减，得到最终该交的税。

为什么不能换？因为这两种运算作用在**不同的量**上。扣除作用在“税率还没上场”的税基上，它必须排在乘税率**之前**——你得先知道要打税的钱有多少，才能打税。抵免作用在“税率已经算完”的税额上，它只能排在乘税率**之后**——税都还没算出来，你拿什么去抵？

把抵免错当扣除塞进第一级，或者把扣除错当抵免放到最后，算出来的数会差出一大截。这就是为什么工程师读税法，最先要建的不是记住哪条能扣多少，而是**建对这条管道的stage 顺序**。

这一章削出了什么

我们把一坨毛收入送进管道，看它被两种完全不同的运算削了一遍：扣除砍税基（省的税还要打个边际税率的折），抵免砍税额（一块顶一块）。顺序不能乱，先削基、再算税、后抵

免。

但你一定注意到了：算扣除到底省多少，我一直在用一个没交代清楚的东西——“**边际税率**”。为什么它是 22%，而不是你收入整体的税率？为什么同样挣钱的两个人，同一笔扣除省下的税可能不一样？下一章，我们就专门拆这个让无数工程师第一次读税表时当场卡住的东西：**你收入的每一块钱，其实不是按同一个税率收的。**

第八章 · 边际税率没有把你的全部收入吃掉

先说一个我几乎每周都能听到一遍的说法。饭桌上，或者公司茶水间，有人一脸严肃地告诉你：

“我老板要给我加薪，可是一加就跳进 24% 那一档了，全部工资都得按 24% 交税，算下来还不如不加。我准备拒了。”

请你记住这一章的结论，它可以帮你身边的人省下真金白银：**这个说法是错的，而且错得离谱。加薪永远不会让你到手的钱变少。**拒绝加薪去省税，就像为了少交电费而把自己家的电闸拉了——省是省了，你也别用电了。

之所以这么多聪明人会栽在这里，是因为“税档”这个词天生带着误导性。它听起来像一个开关：你一旦踩过某条线，整个人“掉进”新的一档，从此全部收入按新税率算。但真实的机器根本不是这么设计的。它是个分段函数。

税额不是“总收入 × 一个数”

我们先把这台机器的核心逻辑说清楚。美国联邦个税用的是累进税率 (progressive / marginal brackets)——听着唬人，拆开就一句话：

大白话

你的收入被切成一段一段，每一段套用它自己那一档的税率。不是拿总收入去乘最高那一档。

想象一排水桶，从矮到高排好。你挣的钱像水一样往里倒：先把第一个桶倒满，再溢到第二个桶，再溢到第三个……**每个桶有自己的“税率标签”**。第一个桶按 10% 抽，第二个桶按 12% 抽，第三个桶按 22% 抽。你的钱流进哪个桶，就只有“待在那个桶里的那部分”按那个桶的税率交税。前面桶里的钱，早就按它们各自更低的税率结算完了，谁也不会因为你后来挣多了而被“追溯”提税。

用工程师的话说，税额 T 是收入 x 的**分段函数 (piecewise function)**：你把收入按门槛切段，每段乘以本段税率，最后把各段的税**加起来求和**。它绝不是 $T = x \times \text{rate}$ 这种一刀切的乘法。

算一遍给你看：5 万美元到底交多少

光说不练是耍流氓。我们拿一个具体数字算给你看。用美国联邦单身档位的**近似值 (取整、仅用于说明原理，别拿去报税)**，假设你的应税所得是 50,000 美元。档位大致这样切：

- 0 到 11,000：按 **10%**
- 11,000 到 44,725：按 **12%**
- 44,725 到 95,375：按 **22%**

你的 5 万块钱，倒进这三个桶里，分段算：

- **第一段** (0 → 11,000，共 11,000)： $11,000 \times 10\% = 1,100$
- **第二段** (11,000 → 44,725，共 33,725)： $33,725 \times 12\% = 4,047$
- **第三段** (44,725 → 50,000，共 5,275)： $5,275 \times 22\% = 1,160.5$
- **加起来**： $1,100 + 4,047 + 1,160.5 = 6,307.5$ 美元

注意最后一段：你处在 22% 这一档，但只有跨过 44,725 门槛的那 5,275 块钱按 22% 交，不是整整 5 万都按 22%。

现在算你的真实税率： $6,307.5 \div 50,000 \approx 12.6\%$ 。

看清楚了吗？你嘴上说自己“在 22% 那一档”，可你实际交出去的，只占全部收入的 **12.6%**。如果真按 22% 一刀切，你得交 11,000 块——机器实际只收了你 6,307.5，几乎少了一半。这不是优惠，这就是累进税率本来的算法。

两个词，钉死它们的区别

上面这一算，其实无意中亮出了两个最容易被搅混的概念。把它们分清楚，你就再也不会被“税档”吓到了。

边际税率：你“下一块钱”的税率

边际税率（marginal rate）是你**再多挣 1 块钱**要被抽走的比例——也就是你所处的最高那一档。上面例子里，你的边际税率是 22%。它管的只有一件事：**你多挣的那一块划不划算**。你要不要加班、要不要接那单私活、要不要接受这次加薪，看的都是边际税率。因为你多出来的钱，全都落在最高那个桶里。

有效税率：你的平均税率

有效税率（effective rate）是**全部税额 ÷ 全部收入**——你交的税平摊到每一块钱上，到底是多少。上面就是 12.6%。因为你前面每一段都按更低的档走过，所以有效税率**永远、必然低于**你的边际税率。它俩相等只有一种情况：你穷到全部收入都待在第一个桶里，从没溢出来过。

一句话钉死：**边际税率是斜率，有效税率是平均。**

换成工程师熟悉的画面：把“税额随收入增长”画成一条曲线。**边际税率是这条曲线在你脚下这一点的斜率——你最新踩的那级台阶有多陡；有效税率是从原点到你这一点连一条直线的斜率——你从起步到现在的平均坡度**。台阶越往上越陡（边际在升），但你走过的平均坡度，永远被前面那些平缓的台阶拉低着。用速率的话说：边际是你此刻的**瞬时速度**，有效是你的**全程平均速度**——只要你起步时慢过，全程平均就永远追不上此刻的瞬时。

“跳档就亏”为什么是彻底的谣言

现在回到开头那位准备拒绝加薪的朋友。他的错误可以一句话点破：**新的一档，只作用于超过门槛的那一部分钱，动不了你门槛以下的任何一分。**

假设加薪让他从 44,000 涨到 45,000，刚好踩过 44,725 那条线进了 22% 档。他慌什么？他慌的是“全部 45,000 都按 22% 交”。可实际上：这 1,000 块加薪里，只有超过 44,725 的那 275 块进了 22% 桶，被多抽了几十块；其余部分该 12% 还是 12%。他到手的钱**照样比加薪前多**，只是多的那一小段被抽得比之前狠一点点而已。

把它推到极端你就彻底安心了：**加薪 1 块钱，最坏的情况，也不过是这 1 块被按最高档抽走几毛，你净到手仍然是正的**。因为美国联邦边际税率从没到过 100%——多挣，永远是净

赚。跨进更高一档绝不会让你的税后收入反而减少，这在数学上是不可能的。

大白话

严谨补一句：确实存在“多挣一点点反而净收入下降”的怪现象，但那不是税率累进造成的。它来自某些补贴、抵免的悬崖式退坡（benefit cliff）——收入一过某条线，整笔福利被一刀砍掉。那是福利设计的锅，是另一台机器的 bug，跟这里的分档税率没关系。税率本身，永远不会惩罚你多挣。

工程师视角：累进是把“公平”编码进了函数结构

退一步看这台机器的设计意图，你会发现累进税率其实很优雅。它想表达一个价值判断——“收入越高的那几段，抽得越狠”——但它**不用惩罚“多挣”这个动作本身来实现**。它把这份“公平”直接编码进了分段函数的结构里：越靠上的段，税率标签越高；可任何一段都不会高到让你在这一段的净收入变负。于是它同时满足了两件看似矛盾的事：高收入者的有效税率被越拉越高，而每一个人多挣一块钱永远还是赚。

所以，理解了边际和有效的区别，你收获的不只是不再被“税档”这个词吓住。你还拿到了一把钥匙：既然一个人的有效税率永远低于他的边际税档，那为什么我们总听说“富人的有效税率反而比中产还低”？——那已经不是分档能解释的了，它和收入的**种类**有关（工资 vs 资本利得），而不是收入的多少。那扇门，我们留到第 10 章再推开。

第九章 · 那张 1040：一年一次的系统结算

到现在为止，前面八章讲的每个零件都在各自转：雇主每个月从你工资里预扣一笔（第2章），券商年底把你的分红利息报给 IRS（第4章），你花的房贷利息、你交的养老金，都在悄悄改变你该交多少税（第7章）。但这些零件从来没在一个地方碰过头。

它们碰头的那一天，叫报税。碰头用的那张纸，叫 1040 表——美国个人所得税的年度申报表，一年填一次。

这一章，我们不学新机制，我们把前八章的零件全拼起来，看它们在这张纸上怎么咬合成一台机器。

先回答那个憋了八章的问题：既然都扣了，为什么还要报？

这是每个第一次工作的人都会冒出来的疑问：工资全年都被预扣了，IRS 手里连我的收入底稿都有了（第4章那份信息申报），凭什么还要我每年费半天劲填一次表？

答案就一句话：**预扣是按估算预付的，不是精算的。**

你入职时填了一张表（W-4），雇主拿它估你一年大概挣多少、大概该交多少税，然后每个月按这个估算，从工资里先扣一笔上缴。注意这个词——**估算**。雇主不知道你年中买了房、生了娃、捐了钱、炒股亏了；它只知道你的工资。所以它扣的那个数，几乎永远不等于你真正该交的数。

大白话

打个你天天在用的比方：预扣就像手机话费的预付款——你每月先充一笔进去，运营商先扣着。但你这个月到底打了多少电话、用了多少流量，得月底出账单才算得准。充的钱和实际花的钱，几乎不可能刚好相等。

1040 就是年底出的那张**精确账单**。它把你这一年真正该交的税一分一厘算清楚，再和你全年已经预付的钱一减：多退少补。这个“算清楚再减、结清差额”的动作，工程师有个词——

结算 / 对账 (reconcile)：把两份数据（该交的 vs 已交的）拉到一起核对，把差额 settle 掉。

所以报税不是重新交一次税。它是**结账**。你全年一直在预付，1040 是把这一年的账一次性对平。

把 1040 读成一条结算脚本

还记得第7章那条计税流水线吗？1040 干的事，本质就是把那条流水线在纸上跑一遍。你可以把这张表看成一个**结算脚本**，从上往下一步步执行，每一步都对应前面某一章讲过的零件：

1. **汇总全部收入**。工资、利息、分红、炒股赚的、接私活的……各种来源加到一起，得到总收入。这里有个隐藏的约束：你填进去的每个数，最好和第4章那份 IRS 已经拿到的底稿对得上——因为你其实是在**照抄一份系统里已经有的数**。对不上，AUR 那台对账机器（还记得第5章吗）就会亮红灯。
2. **减去扣除，得应税所得**。从总收入里减掉标准扣除或逐项扣除（第7章）——就是那笔“这部分收入不计税”的免征额。减完剩下的，叫 应税所得 (taxable income)：真正要拿去算税的那部分钱。
3. **套累进税率，算出税额**。把应税所得喂进第8章那个分段函数：第一段收入按低税率、往上一段按高一点的税率……逐段累加，得出一个初步税额。
4. **减去抵免，得最终该交的税**。抵免（第7章）是直接从税额里减的钱——不是减收入，是减税本身，一块钱抵免顶一块钱税。减完，你得到这一年真正该交给政府的总数，这个数有个正式名字：应纳税额 (tax liability)。
5. **拿“该交的”减去“已预付的”，结算**。最后一步，也是全表的高潮：用应纳税额，减去你全年被预扣的 + 主动预缴的（第2、3章）那笔钱。结果是正数，说明你预付得不够，**补缴**；结果是负数，说明你预付多了，**退税**。

从上到下一条线走完，前八章的零件全用上了。这就是为什么 1040 看着吓人，其实它只是一份把流水线抄在纸上的对账单。

退税不是天上掉钱

很多人把退税当成一年一度的红包，甚至专门把预扣调高一点，图年底“退一大笔”的爽感。这里必须把话说白（呼应第2章）：

退税不是政府发钱给你。退税是你全年被**多扣了**，现在系统把多收你的那部分原样还回来。

大白话

还是话费那个比方：月底账单出来，发现你充多了，运营商把多的退你——没人会觉得这是运营商送你钱。退税一模一样。你那笔钱本来就是你的，只不过被系统提前借去、无息占用了大半年。

反过来，补缴也不是罚你。它只是你全年预付得不够，现在把差额补齐而已。退税和补缴，是同一个动作（settle 差额）的两个方向，本身没有好坏——只说明你的预扣估得偏高还是偏低。

一个绕不过去的怪事：为什么系统不直接帮你算好？

你可能已经想到一个尖锐的问题：既然 IRS 手里早就有你的工资底稿、你的利息分红底稿（第4章），它其实已经知道了你收入的大部分。那它为什么不干脆**直接把税给你算好**，你看一眼没问题、点个“确认”就完事？

这不是空想。很多国家就是这么干的——系统把它掌握的数据预先填好，生成一份 预填申报（税务局先替你填好，你只需核对确认），大多数人几分钟就报完了。

技术上，美国完全做得到。制度上，却偏偏要你自己（或者花钱买 TurboTax 这类软件）把一份 IRS 其实已经有的数，费劲地再敲一遍。

这中间的落差——**技术上本可自动，制度上却没有**——不是技术问题，是利益和游说的问题。这台机器身上这个刺眼的 bug 是怎么被人为保留下来的，我们留到最后一章专门拆。这里先埋个钩子，你记住这个别扭的事实就行。

工程洞见：1040 是这台系统的“最终一致性对账点”

现在退远一点，用系统工程师的眼睛看整台机器。

这是一个**分布式系统**：一年里，无数个节点各自往里写数据，而且是**异步**写的——雇主每个月异步地扣你的税、报你的工资；券商年底异步地报你的投资收益；你自己在一年中的某天买了房、捐了款，异步地产生了扣除。这些写入发生在不同的时间、不同的地方，彼此并不知道对方存在。

大白话

最终一致性 (eventual consistency) 是分布式系统里的一个说法：各个节点的数据在某个瞬间可能对不齐、乱糟糟，但只要给它们一个统一收口的时刻去核对，最终会归到一致。

1040，就是这台税务系统**每年一次的最终一致性对账点**。全年那些散落各处、异步写入的数据，到报税这一刻被全部拉到一起，做一次总核对：该交的算清、已交的减掉、差额结清。对完这一次，这一年的账才真正commit（提交入库、落定生效）——在此之前，你和系统之间那本账，一直是没结平的、悬着的临时状态。

所以，“既然都扣了为什么还要报”这个问题，最干净的答案是：**预扣是全年那些异步的、按估算的临时写入；1040 是把它们收口、对平、commit 的那一下**。没有这一下，全年那堆各自为政的数据就永远不算完成。报完税，系统才对你这一年说一句：账，平了。

不过，这条结算脚本里藏着一个我们一直没细究的假设：它默认你所有的收入都按同一套税率去算。可现实里，同样是报税、同样把 1040 从头走一遍，有人有效税率二十几个点，有人上亿身家却不到二十。差别不在他们填错了表，而在他们的钱是**哪一种钱**。下一章，我们就去看那个让无数人费解的现象：为什么最有钱的人，税率反而可能最低。

第十章 · 为什么巴菲特税率比他秘书还低

第八章我们刚证明了一件挺符合直觉的事：税率是累进的，你挣得越多，落在高档里的钱越多，**有效税率**——你真正上交的钱占总收入的比例——就越往上爬。挣 5 万的人有效税率可能十几个点，挣 500 万的人能到三十几个点。多挣多交，天经地义。

然后现实甩过来一记耳光。

沃伦·巴菲特（Warren Buffett），伯克希尔·哈撒韦的掌门人，全球排前几名的富豪，公开说过一句让很多人下巴掉地上的话：**他缴税的有效税率，比他办公室里的秘书还低**。他不是自嘲，也不是在偷税——他掏出报税表算给你看，是真的低。

秘书年入几万，有效税率二十几个点；巴菲特一年进账几千万上亿，有效税率反而不到二十。这跟第八章说的“多挣多交”完全拧巴了。累进税率不是保证越有钱交得越狠吗？怎么到了金字塔尖，曲线突然掉头往下？

问题出在：第八章只讲了收入的一个**维度**——多少。分档、累进、有效税率，管的全是“你挣了多少钱”。但它漏掉了另一个维度，一个同样决定你交多少税的维度：**这钱是哪种钱**。

美国把“收入”劈成了两条轨道

这是本章的地基，先钉死。在美国的税法眼里，你口袋里进来的钱不是一种东西，是两种，走两套完全独立的税率系统。

第一种叫 劳动所得 / 普通所得 (ordinary income)。

大白话

说人话：你干活换来的钱。工资、薪水、奖金、自雇接活赚的钱——凡是“人出力气/出脑子换来”的，都算这一类。

这类钱走的正是第八章那套累进阶梯，最高档踩得很高，联邦层面能到 **37%**。你越会挣工资，越往这个高档里填。

第二种叫 资本利得 (capital gains)。

大白话

说人话：你低价买进一样资产、高价卖出，中间赚的那个差价。股票 10 块买进 30 块卖出，那 20 块就是资本利得。房子、公司股权、比特币，同理。这是“钱生钱”赚来的，不是你出力换来的。

关键来了：只要你**长期**持有——一般指超过一年再卖——这笔资本利得不走那套 37% 的累进阶梯，而是走一套**单独的、低得多的**税率。联邦长期资本利得就三档，大致是 **0% / 15% / 20%**。你没看错，同样是几百万的收入，一个天花板 37%，一个天花板 20%。

（补一句以免误解：**短期**——买进一年内就卖的——资本利得不享受这个优待，得按普通所得那套高税率算。低税率是专门奖励“长期持有”的。）

一句话钉死：

你靠劳动挣的钱，和你靠资本挣的钱，走的是两套系统。后者（长期持有）明显更低。

工程师读到这里应该已经嗅到味道了：这不是一套费率表，是**两套费率轨道**。同样一块钱进账，贴的标签不同，走的轨道不同，最后交的税差出十几个百分点。

所以巴菲特和他的秘书，根本没在同一条轨道上

现在悖论自己就解开了。

秘书的收入几乎百分之百是工资——纯劳动所得，一分不少地走那套累进阶梯，往高档里爬。巴菲特呢？他的钱绝大部分不是工资，是**投资赚的差价和股息**——资本利得那一轨。

于是哪怕巴菲特一年的收入是秘书的成千上万倍，他那一堆钱适用的**轨道本身**就是低费率那条。收入的量级碾压秘书，但收入的**类型**把税率轨道换掉了，一换就换到了地板价。

请注意，这里没有任何违法操作。巴菲特没藏钱、没做假账、没钻黑洞。他的低税率是白纸黑字的税法**设计出来的必然结果**，不是漏洞被他捡到了。规则就长这样，他只是恰好站在了低费率轨道的入口。

富人怎么合法地把“收入”搬进低税那条轨道

既然两条轨道费率差这么多，一个理性玩家会干嘛？当然是想办法让自己的钱尽量走便宜那条。这在税法上有个正经名字叫**税务筹划**，是合法的，跟逃税（藏收入、造假）是两码事。手法主要几条：

一、少拿工资，多靠持有资产增值

工资是劳动所得，最贵的那轨。所以很多富豪的名义工资低到离谱——你听过的“象征性 1 美元 CEO”就是这套路，一年工资就 1 块钱。他们的财富根本不靠工资，靠手里那一大把公司股权自己涨。股权涨的钱，将来是资本利得，便宜轨。

二、涨了但不卖，就压根不算收入

这条是最狠的，也最反直觉。

资产涨价，在你**卖出之前**，税法根本不认为你赚了钱。你持有的股票从 1000 万涨到 1 个亿，浮盈 9000 万——只要你不卖，这 9000 万**不算收入，一分税不交**。这叫 未实现收益 (unrealized gain)。

大白话

说人话：账面上浮盈，和钱真落进兜里，是两回事。税只在你“实现”——也就是真卖出——那一刻才被触发。不卖，就永远停在浮盈状态，永远不产生应税事件。

工程师最好懂这个：未实现收益就是**账面浮盈但没 commit**。数字在内存里变了，但你没执行卖出那一步，没有一次真正的 `commit(sell)`，系统就不会生成应税事件。于是一个人可以账面暴富、身家千亿，当年真金白银交的税却接近零——因为他一股没卖。

三、“买入、借贷、去世”——把税基整条抹掉

最后这条策略有个很直白的名字，叫 Buy-Borrow-Die（买入、借贷、去世），给个概念就行，不展开太深：

- **买入**：买下资产并长期持有，靠它增值，坚决**不卖**——不实现收益，不触发税。

- **借贷**：要花钱怎么办？拿资产去银行做抵押、借现金花。**借来的钱不是收入**，天然不交税。等于既拿到了现金，又没触发那笔浮盈的税。
- **去世**：人走了，资产传给继承人，这时触发 成本基准重置 (step-up in basis) ——继承时把资产的“成本价”直接重置成当前市值。你当年 1000 万买的、现在值 1 个亿的股票，在继承人手里“成本”就算 1 个亿了。那之前 9000 万的浮盈，**永远不会被征资本利得税**，凭空蒸发。（严格说，特别大的遗产另有一道遗产税要操心，但那条浮盈本身确实躲开了所得税这一轨。）

整条链下来，一大笔财富从头到尾没走过一次高税轨道，最后连那条低税轨道都绕开了。合法，且结构性地把税基抹掉。

为什么要这么设计？两面都听一遍

看到这你可能想开骂了——先别急，这套设计不是天上掉下来的，有它明面上的道理，也有它挨批的地方，两面都摆出来。

官方那一面（为什么该低）：

- 低税是想**鼓励人去投资、去承担风险**。你把钱投进公司、买股票、办企业，是把资本押上去冒险，赚了归你但税收轻点，算是奖励这种冒险。
- 资本利得里有一部分其实是**通胀**——你 10 年前的钱和今天的钱不一样值钱，账面差价有一块是货币贬值撑起来的假利润，全额征税不太公道。
- 还有一层：公司赚的钱在公司层面已经交过一次**企业税**了，股东再就股息/增值交一遍，等于同一笔钱被征两回。低税率算是对这种**双重征税**的缓解（这条我们留到第十二章细掰）。

批评者那一面（为什么不公）：

- 它让**靠钱生钱的人**（食利者）税负低于**靠出力挣钱的人**（打工者）。同样进账一百万，你上班挣的要交三成多，人家躺着让资产涨的只交两成，凭什么？
- 它直接违背了累进税率“多挣多交”的朴素直觉——到了最顶端，曲线反而掉头向下。
- 财富越集中的人越有资产可持有、可抵押、可继承，这套玩法本身就是**越有钱越省税**，加剧不平等。

这里没有标准答案。它是一个真实的张力：一边是“别掐死投资和冒险”，一边是“别让资本的税负低过汗水”。谁重谁轻，是政治账，不是数学题。

工程师的收尾：规则决定行为

把这一章抽象成一句系统语言：

系统按“收入的类型”给了不同的费率，于是理性玩家一定会把自己的收入尽量重新归类到低费率的那一档。

这不是 bug，是激励结构**照定义**会长出来的东西。你给某一类行为定了低价，人就会拼命把自己的行为往那一类上靠——这是系统设计的铁律，不是道德问题。

做过云的工程师对这个再熟悉不过：云服务常常按调用类型 differential pricing（差别定价）——同样是搬数据，走这个接口贵、走那个接口便宜。结果呢？没人会傻乎乎走贵的，所有人都把负载往便宜那类迁，迁到运营商都得回头调价。你定的价格，直接决定了用户把负载摆在哪儿。

税法这台机器一模一样。它把“劳动”定了高价、把“资本”定了低价，那么手里有选择权的人——也就是有大把资产、能决定拿工资还是拿股权的富人——自然会把自己的收入尽量塞进“资本”那条便宜轨道。巴菲特税率比秘书低，不是哪个人坏，是**规则决定行为**：你怎么定价，玩家就怎么走位。

秘书没有选择权，她的收入天生只有“工资”这一种，只能走贵轨。巴菲特有选择权，于是他走了便宜轨。悖论的底，从来不在人品，在于这台机器给两种钱贴了两种价签。而且工资这条“贵轨”上，其实还压着另一道你没细看过的扣款——下一章我们就去揭开它。

第十一章 · 另一条你没注意的扣款管道

前面几章我们一直在拆一根管子：联邦所得税。工资进来，标准扣除削一刀，分档累进抽一层，抽完进联邦大池子。你可能已经默认：“工资单上被政府拿走的那部分 = 所得税。”

不是。你的工资单上，其实并排跑着**两条独立的扣款管道**，各扣各的，各有各的规则，各有各的去处。所得税只是其中一条。另一条，很多人从没认真看过它的名字——而对不少人来说，它扣得比所得税还狠。

先看你工资单上那行没人细看的字

下次拿到工资单，除了“Federal Income Tax”那一行，你会看到另外几行缩写，通常写着 FICA，或者拆成 Soc Sec 和 Medicare。这就是第二条管道。

工资税 payroll tax，说白了就是“专门盯着你这份工资、在发钱那一刻当场扣走”的税。在美国它主要指 FICA（Federal Insurance Contributions Act，联邦保险捐款法案下的那笔扣款）。名字里带“保险”两个字不是随便起的——待会你会看到，这笔钱和普通的税不太一样，它是有指定用途的。

FICA 由两块拼成，我们一块一块拆。

第一块：社会保障税，喂养老金

社会保障税 Social Security tax，就是给全国养老金池子交的钱。你现在交的，直接发给现在正在领养老金的老人；等你老了，那时候上班的人交的钱再发给你。它是一个“当代人养上一代人”的循环。

税率大约是这样（都是近似值，逐年微调）：你自己出约 **6.2%**，你老板再出约 **6.2%**，加起来 **12.4%**。

但它有个关键特征：封顶 wage cap。社保税只对你年收入的**前面一段**征。2024 年这个封顶线大约在 16.8 万美元。也就是说，你收入超过这条线的部分，一分社保税都不再交。记住这个“封顶”，它是本章最反直觉那个结论的开关。

第二块：医保税，喂老年医保

医保税 Medicare tax，交给老年医保池子（Medicare，65 岁以上那套联邦医保）。税率你出约 **1.45%**，老板再出约 **1.45%**，合计 **2.9%**。

和社保税最大的区别：**医保税没有封顶**。你挣多少，就按这个比例一路扣到底，没有那条 16.8 万的线。（收入特别高的人还要多交一点点附加税，这里点一句就够，不展开。）

那“老板出的一半”，到底是谁在出？

你从工资里肉眼看到被扣走的，只是“你自己那一半”： $6.2\% + 1.45\% \approx 7.65\%$ 。另一半，账面上是老板出的。听起来像老板替你分担了一半，挺好。

但这里有个经济学家几乎一致同意的洞见，工程师会特别喜欢，因为它区分了“接口上写的谁”和“实际负载压在谁身上”：

大白话

名义上谁交钱（法律规定谁往政府那儿打款）和实际上谁承担这笔成本（谁的钱包真的变薄了），是两码事。这个区别有个专门的词叫 税负归宿 tax incidence——税这个“负载”最终落在谁头上。

对老板来说，“给你的工资”加上“替你交的那 7.65%”是同一笔用人成本。羊毛出在羊身上：如果没有这条规定，那笔钱本来大概率会变成你更高的工资。所以主流看法是——雇主出的那一半，最终也是从你的工资里出的，只是你未在工资单上看见它而已。你真实被抽走的，接近**全额 15.3%**，不是 7.65%。

洞见一：这笔钱是“专款专用”的，不进大池子

这是 FICA 和所得税最结构性的区别。

所得税进的是联邦**一般预算池**——一个大水池，军费、修路、发工资、还利息，从里面随便舀（这池子怎么花，第 15 章专门讲）。

FICA 不是。它是**专款专用 earmarked** 的：社保税只能进社保信托基金，医保税只能进医保信托基金，名义上就只喂“养老金”和“老年医保”这两件事，不许挪去修航母。

用软件预算打比方：所得税是公司那个“一般经费池”，谁有理由都能来申请。FICA 是财务给你单开的两个**指定用途专项账户**，账户名就写死了“养老”和“老人医保”，转账时科目对不上就是转不出去。两条管道，两个账本，井水不犯河水。

洞见二：它几乎是“平税”，甚至累退——和所得税正好反着来

这是全章最该记住的一点，也是最反直觉的。

回忆第 8 章讲的所得税：它是**累进**的——有一大块标准扣除先给你削掉，然后低收入落在低税档，越往上税率越高。它天生“照顾低收入、多抽高收入”。

FICA 是反过来的。两个原因：

1. **从第一块钱就开扣，没有免征额。**所得税有大额标准扣除，能把低收入者削到几乎不交。FICA 没有这个东西——你赚到的第一美元，就要交 7.65%。没有低档，没有起征点，一上来就是满税率。
2. **社保税还封顶，所以对高收入者累退。**年入 5 万的人，全部 5 万都要交社保税；年入 50 万的人，只有前面约 16.8 万交社保税，剩下的 33 万完全免。收入越高，被社保税打到的比例反而越小——这就是**累退 regressive**：越有钱，这项税的有效税率越低。

把这两条合起来，会得出一个很多人不知道的事实：

对大量中低收入的工薪族来说，一年交的 FICA，比交的联邦所得税还多。

为什么？因为所得税用大额标准扣除加低税档，能把他们的所得税削到很低甚至接近零；而 FICA 从第一块钱就开扣、没有任何减免。于是对底层劳动者，**他一年最大的那一笔联邦税，往往不是所得税，而是工资税**。如果你只盯着“所得税”去衡量一个普通工人的负担，会严重低估——你漏看了那条更粗的管道。

和第 10 章接上：这条管道只打劳动，不碰资本

还有一层优待，得和上一章连起来看。

FICA 只打**劳动所得**——工资、自雇收入。你靠炒股、收租、拿分红赚的钱，也就是资本利得那一挂，**完全不进这条管道**，一分 FICA 都不用交。

把第 10 章和这章叠起来，靠资本赚钱的人享的是**双重优待**：那份收入既走更低的资本利得税率（第 10 章），又整个躲开了 FICA。而靠双手挣工资的人，两条管道一条都躲不掉。同样赚一笔钱，来源是“劳动”还是“资本”，真实税负能差出一大截。

顺带说一句自己给自己发工资的人：自雇税 self-employment tax。自雇者没有老板替他出那一半，所以两边得自己全扛，合计约 **15.3%**。前面说的“其实是你在出全额”，对自雇者是明账——他直接就交了全额。

工程视角：税不是一根管子，是一叠并行子系统

把这章收进系统模型里。

FICA 是一个和主系统（所得税）**并行运行、独立计费、目的绑定**的子系统：

- **并行**：它和所得税同时在同一张工资单上跑，互不干涉，各算各的。
- **独立计费**：它有自己的费率表（固定比例 + 封顶），和所得税那套累进分档完全是两套逻辑。
- **目的绑定**：它的输出被 earmark 死了，只流向养老和老年医保两个专项账户，不进大池子。

所以“税”从来不是一个单一管道。它是好几条各有各计费规则、各有各去向的管道，叠在同一张工资单上。想算清一个人真实被抽走多少，你得把这些平行管道**全部加起来**看——只盯着所得税那一根，对底层劳动者会严重失真。

下一次有人跟你说“我们这个收入档所得税不高”，你可以反问一句：那 FICA 呢？对很多人来说，答案就藏在那条他从没细看过的管道里。到这儿，个人这个节点我们算是拆透了。下一章，我们把镜头抬高，去看系统里另一大类节点——公司，是怎么被课税的。

第十二章 · 公司这个节点怎么被课税

前面十一章，我们盯着的纳税节点全是**自然人**——一个有身份证、有工资单、会退休、会死的人。系统对这类节点的处理我们摸得差不多了。现在把镜头往上抬一点，你会发现税务这台机器里还挂着另一大类节点：**公司**。

公司这个节点有意思在哪？一块钱的公司利润流进系统，被征税的方式居然**不是唯一的**。同一块利润，有的公司会被征两次，有的只征一次——而且不是系统随机决定，是公司自己在注册的时候**选**的。这在工程师眼里属于典型的“这设计怎么这么怪”，但拆开看，它其实很规整。

两条路：利润怎么流出公司

把一家公司想象成一个装利润的容器。一年下来，收入减掉成本，容器里剩下一笔利润。问题是：这笔利润要交税，在哪一层交、交几道？税法给了两条截然不同的路。

路一：C 型公司——公司先交，股东再交

C 型公司（C-corp）是大公司的默认形态，你能叫得出名字的上市公司基本都是它。它的关键特征是：**公司本身就是一个独立的纳税节点**，在税法眼里跟一个“人”差不多，有自己的报税义务。

于是利润要过两道关卡：

1. 公司赚了 100 块利润，先交一道**企业所得税（corporate income tax）**——就是公司自己对着这笔利润交的税。2018 年之后美国联邦税率大约 **21%**。交完，容器里剩 79 块税后利润。
2. 公司把这 79 块作为**股息（dividend）**——也就是“分红”，公司把税后利润按股份发给股东——分给股东。股东拿到这笔钱，**还要再**就它交一道个人税。

看清楚了：同一块钱的利润，在公司层被抽了一次，流到个人手里又被抽了一次。这就是税法里著名的**双重征税（double taxation）**。

大白话

就像一笔数据要过两道收费站：第一道在“公司”这个关卡收 21%，第二道在“个人”这个关卡再收一次。同一块钱，过两次闸机，刷两次卡。

路二：穿透实体——跳过公司，只在个人交

另一条路叫穿透实体 (pass-through entity)。合伙企业、S 型公司 (S-corp)、LLC、个体经营，绝大多数中小企业走的都是这条。

它的特征正好相反：**公司本身不交企业所得税**。利润“穿透”公司这层壳，直接算成背后股东或合伙人的个人收入，只在个人层交一道税。公司在这里不是一个纳税节点，只是个管道。

大白话

这就是工程师天天用的 pass-through ——透传代理。请求打到代理上，代理不处理、不截流，直接把它透传给后面真正干活的服务。这里公司就是那个透明代理，利润直接流到背后的人身上，按**个人税率**征，管道这一层不截一道税。

一句话把两条路钉死：

C-corp 是“公司先交、股东再交”，两道；穿透实体是“跳过公司、只在个人交”，一道。

既然双重征税更亏，为什么还有人选 C-corp？

工程师看到这里第一反应肯定是：那不选 C-corp 不就完了？被征两道税，图什么？

因为“少交一道税”不是唯一的目标函数。C-corp 换来的是另一组好处：

- **便于融资和上市**：风投、公开市场都习惯往 C-corp 里投钱，股份结构清晰、能发不同类别的股票。你想拿 VC 的钱、想 IPO，基本就得是 C-corp。
- **无限存续**：公司是独立实体，股东换来换去公司照样活着，不因某个人退出而散伙。

- **利润留存能延迟第二道税**：这条对工程师特别关键——第二道税是在**分红那一刻**才触发的。只要公司把税后利润留在容器里继续投资、不分给股东，那道个人税就**暂时不发生**。所以一家高速再投资、根本不分红的成长型公司，实际上长期只承担了那 21% 的公司税——注意是**延迟**那第二道税，不是免掉：等哪天真分红了，个人税照样触发。
- **21% 这个税率本身不高**：对于要把利润留下来再投资的公司，21% 的留存成本可能比塞进个人高边际税率还划算。

而穿透实体呢，避开了双重征税，但也有它的代价：利润**无论你分不分给自己，当年都算你的个人收入**。公司今年赚了钱、一分没往外拿、全留着扩张，你照样得按这笔利润交个人税——现金还在公司账上，税单已经到你个人头上了。而且这笔利润叠加到你其它收入上，很可能把你顶进更高的个人边际税率。

所以“选哪种形态”根本不是“哪个省税”这么简单，它是一道实打实的**税务优化题**：你的规模多大、要不要融资、利润是分掉还是留着再投、你个人的边际税率多高——不同的输入，最优解不同。这正是我们在第 10 章反复讲的那句话：**规则决定行为**。系统摆出几种规则，理性的节点就会挑对自己最省的那一种。

回收第 10 章埋的那根线

还记得第 10 章讲个人拿股息和长期资本利得时，享受的是那套 0%/15%/20% 的低税率吗？当时留了个“为什么它们比工资税率低”的问号。现在可以收线了。

正因为 C-corp 的利润在公司层**已经被 21% 征过一次**，等它作为股息流到个人手里、或者你卖掉升值的股票兑现资本利得时，如果再按普通工资那套高税率征第二道，那这块钱被啃得就太狠了。所以个人层给它一个较低的税率，本质上是**对双重征税的一种部分补偿**——承认“这块钱在上一层已经交过了”。这两章的规则，是咬合在一起的。

顺带说一句：企业怎么合法把利润做小

有件事得点明白：企业所得税征的是**利润**，不是营收。

大白话

利润 = 收入 - 可扣除成本。收税机器盯的是这个减法的结果，不是收入本身。

这个公式一摆出来，公司的动力就全写在脸上了：**把等号右边的“可扣除成本”尽量做大，利润这个税基自然就小了**。研发投入、借钱的利息、设备的折旧、给员工的薪酬……凡是能算进可扣除成本的，公司都有强烈动机往里塞。这不是作弊，是规则允许的减法。

再往上一层，跨国公司还能玩利润转移（profit shifting）——把利润在账面上挪到税率低的国家或地区，让高税地的账面利润看起来很薄。这就是为什么你会看到一些巨头，账面盈利上千亿，实际交的税却少得离谱。这背后是一整套系统级的漏洞，**我们留到第 16 章再动手拆**，这里只把概念放这。

工程洞见：公司是一个“可选类型”的节点

把这一章浓缩成一句工程师能记住的话：

在税法眼里，公司不是一个固定对象，而是一个可以选择“类型”的实体。你注册成什么类型，就决定了利润按哪套规则流动、在哪一层被征税。

这特别像我们给同一段业务逻辑挑部署形态或账户类型：同样的代码，跑在这个账户类型下是一套计费点和费率，换个类型就是另一套。系统预先备好了 C-corp、S-corp、LLC、合伙、个体这几个“实体类型”作为**配置项**，创业者做的事，就是根据自己的目标——要融资还是要省税、利润要分还是要留——从菜单里挑一个最省税又最匹配业务的配置。

所以，公司这个节点比个人灵活得多。个人几乎没得选，生下来就是那套规则；公司却能在注册那一刻，为自己选一条利润的流动路径。而且到目前为止，我们讲的这一整套规则——所得税、资本利得、企业税——都还只是**联邦**这一层的。别忘了，你脚下还踩着一个州。下一章，我们就去看那五十个各自为政、规则打架的平行税务系统。

第十三章 · 五十个平行的税务系统

前面十二章，我们一直在拆一台机器：**联邦**那台。IRS、联邦所得税、预扣、抵免……好像美国人交完这一份，账单就结清了。

但如果你真在美国上过班、发过工资单，会发现一件事：工资条上不止一行“税”。有一行写着 Federal，下面还有一行写着 State（有时候再下面还有一行 City）。这不是同一笔钱被拆开记账，而是**两个完全独立的政府，各自向你伸手**。

先搞清楚一件事：这是两级独立的收税主体

美国是联邦制 federalism。

大白话

大白话：不是“中央有分公司在各地”，而是“联邦政府”和“五十个州政府”是五十一个各自独立、各自立法、各自收税的实体。联邦不能命令加州怎么收税，加州也管不着联邦。它们平级，各收各的。

所以你交的所得税，往往不是一份，是两份叠加：先给联邦交联邦所得税，再给你所在的州交一份州所得税 state income tax。它们不是二选一，是**叠在一起**的——联邦收 22%，州再收 6%，那你这一档实际被拿走的就是接近 28%（细节上州税还能在联邦层面抵一点，但大框架就是相加）。

如果你住在纽约市，恭喜，头上还有第三层：**市级**所得税。

大白话

工程师的类比：一个请求进来，要依次穿过好几层独立的中间件/网关。第一层网关（联邦）收一道过路费，第二层网关（州）再收一道，第三层（市）可能又收一道。每层都是独立部署、独立配置的，你没法跟其中一层说“我在上一层已经交过了”。

五十个州 = 五十套彼此不兼容的规则

如果只是“联邦一套、州再来一套”，那还好。真正让人头大的是：**五十个州不是同一套规则的五十份拷贝，而是五十套各自独立演化、字段定义都不一样的系统。**

差异有多大？举几个极端：

- 有大约九个州**压根没有州所得税**——德州、佛州、华盛顿州、内华达州等等。你在那儿挣工资，州这一层收你零。它们靠别的税种补回来。
- 另一头，加州最高一档税率超过 **13%**。同样挣一百万，在加州和在德州，光州税就差出十几万美元。
- 税率是几档、每档从哪切、能扣什么、能抵什么——各州一套。连“**什么算应税收入**”这个最底层的定义，各州都可能不一样。

大白话

翻译成成人话：这就像五十个微服务，每个都自己定义了一套数据格式（schema）。有的服务连“收入”这个字段的类型和口径都跟别人不同。没有一个全局统一的表结构，也没有谁有权强制统一。

而且“税”不止所得税这一种

就算你搬到没有州所得税的州，也别高兴太早。州和地方还有几条平行的收税管道：

- 销售税 sales tax：你买东西时结账多加的那几个百分点。各州、甚至各县各市税率都不同，有的州干脆没有。没有州所得税的州，往往就是靠这个补窟窿。
- 财产税 property tax：按你房产的估值，**每年**收一笔。这笔钱主要归地方（县、市、学区），是公立学校和市政的主要钱袋子。德州没有州所得税，但财产税出了名的高——羊毛总得从某只羊身上出。

所以“哪个州税低”从来不是看单一税率，而是看这几条管道加起来、落到你这种收入结构的人身上，总共被拿走多少。

一句话记住：**联邦是一套规则，州是五十套彼此不兼容的规则，叠在同一个纳税人身上。**

税负差这么大，人就会“用脚投票”

既然同样的收入在不同州被拿走的钱能差出十几个百分点，理性人当然会算这笔账。于是就有了**用脚投票**：搬家。

这不是段子，是正在发生的迁徙。大公司把总部从加州搬去德州，富人和高收入远程工作者往佛州、德州、田纳西州跑——因为这些州州所得税是零。州与州之间，其实在打一场税收竞争 tax competition的仗。

大白话

大白话：州的“税基”就是它能收税的人和公司。人和公司是会跑的。所以低税州用低税率去抢别人州的税基，高税州则担心把人赶跑。这就是一群独立节点为了抢用户而竞相调参数——只不过参数是税率，用户是纳税人。

但套利没那么简单：两条线都想抓住你

你可能以为，搬个家、把驾照换了，就干净利落地脱离原来那个州了。没这么容易。因为州主张“你归我管、该给我交税”，用的是**两条独立的线**：

1. 税收居民身份 residency：你是哪个州的税务居民（大致看你主要住在哪、重心在哪）。你的居住州，倾向于对你**全球所有收入**征税。
2. 收入来源地 source：这笔钱是在哪个州挣的。钱在哪个州产生，那个州就想对这部分收入征税，**不管你人住哪**。

两条线一叠，麻烦就来了：你住 A 州（居民线，A 想收你全部），但钱在 B 州挣的（来源线，B 想收这部分）。同一笔收入，**两个州都伸手**——这就是**双重征税风险**。

系统当然打了个补丁来对齐：**他州税抵免**。你的居住州 A 一般会说：你在 B 州已经交的那部分税，我给你抵掉，别重复收。

大白话

类比：两个微服务对同一笔订单都记了一笔账，靠一个“抵免协调”协议来对账，把重复的那部分冲掉。能对上账，但对账逻辑本身就是复杂度的来源，接缝处一堆边界情况。

远程办公和跨州通勤，把这套本就别扭的协议搅得更乱。有些州甚至有一条叫便利雇主规则 convenience rule的东西（名字你记一下就行，细节能写一整章）——大意是：哪怕你人在家远程办公，只要是“图自己方便”而不是雇主要求，你的收入照样算在雇主所在州挣的。于是你人在新泽西的沙发上敲键盘，纽约州还是要收你的税。

把这些叠起来，你就得到一个真实存在的倒霉蛋：**住新泽西、在纽约上班、公司注册在特拉华**。他一个人要同时面对好几个州的税务规则和对账逻辑。

工程师的总结：一个没有全局 schema 的多系统并存架构

把这一章拉到最高的抽象层看，事情其实很清楚：

大白话

联邦想统一（一套规则多好），但美国宪法一开始就把征税权分给了各州。于是你得到的不是一个统一系统，而是**五十个各自演化、字段定义都不一样的子系统**，硬要拼在一起给同一个纳税人做结算。

它们之间靠一些“协议”勉强互操作：居住地规则决定谁是你的主账户，他州税抵免负责对账去重。**能跑，但集成层惨不忍睹**——所有的复杂度、漏洞和套利空间，全都藏在这些接缝里。跨州套利者钻的，正是这些子系统定义不一致、协议对不齐的缝。

最后一句最关键，别把它当 bug：

这种复杂度**不是设计失误，而是“地方自治”这个特性的必然代价**。你想要每个州能按自己的意愿决定收多少税、怎么花，就必然得接受五十套不兼容的规则叠在一起。统一和自治，在这里是不能同时满足的两个约束。

所以下次看到那张工资单上 Federal、State、City 三行税，你该知道：那不是同一台机器的三个模块，而是三台各自独立、各自立法、还在互相竞争和对账的机器，恰好都盯上了你这一份收入。而无论联邦还是州，它们抽上来的钱最后去了哪、又为什么总不够花，是我们接下来两章的事。

第十四章 · 用税表发钱：藏在系统里的福利管道

前三十三章，我们一直在看同一件事：一台巨大的机器，把钱从三亿人的口袋里，一点一点抽进国库。预扣、信息申报、1040 表——全是“抽水”的管道。

这一章要讲一个反转：**同一台机器，能反过来给人发钱**。而且它发钱走的不是另建一套福利系统，就是你已经熟得不能再熟的那张报税表。

先复习：抵免是“从税里直接减”

回忆第 7 章。抵免 (credit) 不是从你的收入里减，而是从你**算出来该交的那笔税里**直接减。你该交 3000，有一笔 1000 的抵免，那你只交 2000。一块钱抵一块钱，实打实。

但普通抵免有个天花板：**最多把你的税减到 0，减完为止**。你该交 800，却有 1000 的抵免？对不起，减到 0 就停了，剩下的 200 作废，政府不会因此欠你钱。这种叫不可退还抵免 (non-refundable credit)——它只能帮你少交，不能帮你多拿。

可退还抵免：减到 0 还不停，政府倒找钱给你

现在看它的孪生兄弟，可退还抵免 (refundable credit)。规则只改了一个字：如果抵免额**超过**了你本来该交的税，超出的那部分——政府倒找钱给你。

大白话

还是那个例子：你该交 800，抵免有 1000。不可退还的减到 0 就停，200 蒸发。可退还的呢？先把 800 抵光，剩下的 200 政府直接开张支票寄给你。你不但一分税没交，还净赚 200。

请注意这时候“退税”这个词已经变味了。它退的**不再是你多缴的钱**——你可能压根就没缴过那么多。它退的是政府**净发给你的一笔补贴**。同一个动作、同一张支票，方向却掉了个个儿：管道在反向泵水。

主角登场：EITC，专发给“有工作的穷人”

可退还抵免里块头最大的一个，叫 EITC (Earned Income Tax Credit，劳动所得税收抵免)。名字绕，意思很直白：**它是专门发给“有工作、但收入低”的人的一笔钱**，尤其照顾带孩子的家庭。

因为它是可退还的，对很多低收入家庭来说，报税季领到手的那笔钱，根本不是“退还我多缴的”，而是政府**实打实发的几千美元**。这不是省税，这是进账。

它有多大？EITC 是美国规模最大的反贫困项目之一——每年惠及大约两三千万个家庭，发出去的钱在数百亿到上千亿美元这个量级（数字是近似）。它不是边角料，它是这台机器“反向发钱”这个方向上最粗的一根管子。

形状里藏着一个精妙设计

普通的福利常常是“你越穷，发你越多”。听起来公平，可它有个副作用：一旦你去工作、收入涨了，福利就砍，等于变相惩罚你去挣钱。人就容易卡在“不如别工作”的坑里——这叫福利陷阱。

EITC 的设计者故意躲开了这个坑。它给钱的形状分三段：

- **递增段 (phase-in)**：一开始，你挣得越多，政府补你越多。这是故意的——先给你一个“去工作、多挣钱反而更划算”的正向激励。
- **平台段**：收入到一定水平，补贴涨到顶，稳住不动。
- **退坡段 (phase-out)**：收入继续往上升，补贴慢慢往下退，退到某条线以上就一分没有了——因为你已经不算“需要帮”的那群人了。

关键在**你必须先有劳动收入才能拿**，而且在递增段里，越工作拿越多。它等于把“发福利”和“逼你去工作”这两件事缝在了一张表里：不工作，一分拿不到；工作了，还奖励你。

这股味道，经济学家弗里德曼早在几十年前就描述过，他管它叫负所得税 (negative income tax)：收入高于某条线，你向政府交税（正的）；收入低于那条线，政府反过来倒给你钱（负的税）。EITC 就是这个思想落地的一个变种。

核心问题：为什么发福利，要伪装成“退税”？

到这儿该问最要命的一句了：政府想给穷人发钱，为什么不干脆建个福利局，非要把它塞进报税表，伪装成一笔“退税”？

答案是纯工程的。你想想，要判断“谁够穷、够资格领这笔补贴”，你需要什么数据？收入、家庭结构、有几个孩子、孩子多大——**这些数据，报税系统里全有了**。前十三章那套预扣、信息申报、1040，早就把每个人的收入和家庭摸了个底朝天。

复用已有的基础设施，是最省的工程选择。另建一套福利机构意味着什么？重新招人、重新采集所有人的收入证明、重新核验身份、重新搭一条发钱的渠道——把 IRS 已经干过一遍的活儿，全部再干一遍。纯粹的重复造轮子。

大白话

写过后台的都懂这个取舍：你要做退款，会专门为退款重新搭一套支付系统吗？不会。你会复用现成的支付网关，正向能扣款，反向就能打款——同一条链路，同一套身份和金额校验，加个 refund 分支就完事了。EITC 就是在报税这条“支付网关”上，加了一个反向打款的分支。

收钱和发钱，共用同一套 pipeline、同一张 1040 表、同一个身份与收入核验。你已经填过收入了，系统顺手就能判断“这人该倒找钱”，然后把支票发出去。省得漂亮。

别只夸：这台反向泵也漏水

复用基础设施省是省，但把福利缝进报税表，代价是真实的。

- **该领的人没去领。**报税表本就复杂，普通人根本搞不清自己够不够 EITC 的资格。结果是一大批够格的人白白错过——这个“够格的人里真去领的比例”，专业叫领取率 (take-up rate)，EITC 的领取率一直不够高。福利发不到最需要的人手里，等于漏水。
- **错发、多发不少。**规则太绕（孩子算谁的、收入怎么界定、住满多久），算错的概率就高。这类错误发放 (improper payments)在 EITC 上比例一直偏高——有的是纳税人算错，有的是钻空子，钱发歪了。
- **让本该简化的报税更复杂。**把“你该交多少税”和“你该领多少补贴”绑在同一张表上，这张表就更难填了。省了政府建机构的钱，成本转嫁给了每一个坐在表格前发懵的人。

工程洞见：它其实是一个“人与钱的双向接口”

把这一章的东西收束成一句话：

一旦你建成了一套触达全民、掌握每个人收入的基础设施，它就不再只是一个“收税工具”，而是一个通用的“**人与钱的双向接口**”。

收税，只是这个接口的一个方向。反过来，发 EITC 补贴、做收入再分配、甚至疫情期间那几轮直接打到账户里的纾困支票——走的全是这同一条管道。政府不需要为“发钱”重新认识你一遍，因为通过报税，它早就认识你了。

这才是这台机器真正的威力所在：它是政府触达每一个公民钱包的**那根管子**。正向能抽水，反向能泵水。你以为你在跟一台收税机器打交道，其实你面对的，是一个双向的阀门——今天它从你这儿收钱，明天它就能给你打钱。用的，是同一根管子。

第十五章 · 收上来的钱去哪了

前面十四章，我们一直站在这台机器的进料口。看它怎么从工资单里预扣，怎么用第三方报告堵住漏报，怎么把几亿人的账单自动对平。现在把镜头掉个头，走到出料口——这几万亿收上来，到底花哪去了。

因为收税从来不是目的。没人为了收税而收税。税是给支出**融资**的手段。你不看清楚钱往哪流，就永远只看懂了半台机器。

先看钱的去向：三堆半

联邦一年花掉的钱，粗粗分四块。量级我给近似，你抓结构，别抠小数点。

第一堆，也是最大的一堆：自动发的钱

叫 强制性支出 (mandatory spending)。名字唬人，意思特简单：**这些钱是法律写死的，到期自动发，国会每年不用投票**。谁符合条件谁领，法律说了算。

它主要是三样：养老金 (Social Security，退休了每月给你打钱)、老年人医保 (Medicare)、穷人医保 (Medicaid)。这一堆加起来，占了联邦支出的一大半还多。

大白话

还记得第 11 章那条 FICA 专款管道吗？就是从你工资里单独抠出来、贴着“社保/医保”标签的那笔钱。它喂的就是这一堆。进料口有条专门的管子，出料口有个专门的大池子，两头是对上的。

关键在“自动”两个字。这堆钱像开了定速巡航——法律不改，它就一直发，谁也按不停。人口越老，领的人越多，这堆就越大，而且没人踩刹车。记住这点，等下有反转。

第二堆：国防

军费。一大块，实打实的一大块。航母、军饷、装备、海外基地都在这。

第三堆：国会每年吵着分的那点钱

叫 非国防自由裁量支出 (discretionary spending)。“自由裁量”就是——**这部分国会每年投票现决定，想多给谁少给谁，当年说了算**。教育、科研、修路建桥、各个部门日常运转，都在这。

你以为这块很大？其实只占一小块。国会山上吵翻天、上电视骂“政府乱花钱”的，绝大多数就是在抢这一小块的分法。

反直觉的地方在这：大家骂得最凶的“乱花钱”，指的往往是这最小的一块自由裁量。真正的大头（养老、医保）是法律锁死的自动支出，吵也没用，动不了——除非去改法律，而改这种法律是政治核弹，没人敢碰。

剩下那半堆：利息

为过去借的钱付的利息。这块单独拎出来讲，因为它正在快速长大，是这章后半场的主角。

收不抵支：赤字和国债，一个是流量一个是存量

现在问题来了。要发的这几堆钱，加起来比收上来的税还多。怎么办？

先把两个总被人搞混的词分清楚。

赤字 (deficit)：**某一年，政府花的 > 收的，差出来的那块就是当年赤字**。今年收 5 万亿、花 7 万亿，赤字就是 2 万亿。美国几乎年年赤字——年年花得比收得多。

国债 (national debt)：**历年赤字一年一年累加起来的总欠款**。

用信用卡想最清楚：

- 赤字 = 你上个月透支了多少。是**流量 (flow)**，一段时间发生的量，带“每年”这个单位。
- 国债 = 你信用卡账单上累计欠了多少。是**存量 (stock)**，某一刻的总数，就是个数。

每个月透支一点（赤字），欠款总额就往上垒一点（国债）。哪怕某年你少透支了、赤字变小了，只要还在透支，总欠款照样在涨。这俩别混——一个是“这个月又超了多少”，一个是“总共还欠着多少”。

缺口怎么补：政府刷信用卡

当年差的那块钱，政府去哪儿弄？借。

怎么借？发 国债券 (Treasury bonds) ——本质就是**财政部打的借条**。政府印一批借条卖出去，白纸黑字写着：现在给我钱，将来连本带利还你。谁买？个人、银行、养老基金、外国政府、还有美联储，都买。

所以填缺口的动作，说穿了就是政府刷信用卡：**今天的支出，用未来的税收来还——要么等未来真收上税还，要么到期了再借一笔新债还旧债**。年年赤字，就是年年往外发借条，国债这个总欠款于是一年比一年高。

工程师该警觉的地方：利息在滚雪球

到这，任何写过带反馈系统的人都该竖起耳朵。因为借条是要付利息的。

把链路捋直：

1. 债欠得越多，每年要付的利息就越多。
2. 可政府本来就收不抵支，付利息的钱它也没有。
3. 于是只能再借钱来付利息。
4. 一借，总债又变大了。回到第 1 步。

这是一个标准的**正反馈**——输出反过来把输入推得更大，越滚越快，不自己收敛。

大白话

你太熟这个东西了，这就是**技术债的利息**。当年赶工留了个烂设计（借了债），你没空重构（不还本），于是每加一个新功能都得绕着它走、多花一份维护成本（付利息）。债不还本，光利息就越吃越多，最后利息本身就吃掉了你大半迭代预算——你忙一整个季度，全在还这份“利息”，没往前挪一步。

联邦这边一模一样。那第“半堆”利息支出正在快速膨胀，膨胀到什么程度算危险？**当每年光付利息就大到开始挤占别的支出、甚至逼近国防这种大项时，就是系统在报警**——你越来越多的收入不是拿去买服务、办事情，而是纯粹在给过去的账单交利息。

那为什么美国能一直这么借下去？

讲到这别急着喊末日。因为美国有一个别国没有的特权。

美元是全球储备货币，美债被全世界当成“最安全的资产”——真出事了大家反而往美债里躲。结果是：**全世界排着队抢着借钱给美国**。这种举债能力，普通国家做梦都没有；换个小国这么借法，早就没人接盘、利率飙上天了。

还有一个更该记住的判断姿势：看债务不能只看绝对数字，要看 债务 / GDP 比率——**欠的钱相对于整个经济体量有多大**。

大白话

一个人欠 100 万，是穷是富？光看 100 万没意义。他年入 20 万，这债压得喘不过气；他年入 500 万，这就是笔小账。GDP 就是这个国家的“年收入盘子”，债务 / GDP 才是真正说明问题的那个比值。绝对数字吓人，但要放到盘子里量。

可这也不是没边的。约束是实打实存在的两条：一条是刚说的**利息负担**——正反馈总有一天让利息占比大到咯人；另一条是政治上的 债务上限 (debt ceiling)，国会给总欠款设的一道法定天花板，每隔一阵就为要不要抬高它拉锯一次，闹到快违约再收场。

所以这里是一个张力，不是一个结论：**能持续，但不是无限**。别信“明天就崩”，也别信“随便借毫无代价”。

合上回路：税只是前半程

现在整台机器你算看全了。它根本不是一台“收税机”，它是一个**收入—支出—借贷三者联动**的闭环：

- **收入**：前十四章拆的一切——预扣、第三方报告、自动对账——都是为了把税高效收上来。
- **支出**：收上来的钱流进养老、医保、国防、各部门。
- **借贷**：支出长期大于收入，中间那道缺口，靠发国债这个**缓冲区 (buffer)** 垫着。

缓冲区是这个闭环最关键、也最容易被误读的部件。它让系统在收不抵支的年份照样能运转——今年不够，先从缓冲区里垫，明天再说。可缓冲区不是免费的：它靠的是市场愿意一直接你的借条。

而市场为什么愿意接？归根到底，是因为它相信这台机器**未来还收得上税**。信心的底座，正是前十四章那套高效、难逃、能自动对平的征收系统。

所以你看，绕了一整本书讲“怎么把税又快又稳地收上来”，最后服务的其实是这一个问题：**这个收入—支出—借贷的闭环，还能不能一直转下去**。税不是终点，是让这台机器活下去的燃料。收得上，缓冲区就有人垫；哪天市场不信它还收得上了，缓冲区就没人接了——那才是这台每年收上万亿的机器，真正的生死线。

不过在合上这本书之前，还有一件事这台机器没告诉你：它其实从不在海关停下。你以为出了国境就出了它的射程，可它盯的从来不是“住在美国境内的人”，而是“全世界拿美国护照的人”。下一章，我们跟着它跨过国界，看它怎么把前面那三招投射到整个地球。

第十六章 · 全世界的银行都在替它盯着你

前面十五章，我们一直默认一件事：这台机器管的是“住在美国的人”。你在美国挣钱、在美国花钱，它就盯着你。听上去天经地义。可这里藏着一个连很多美国人自己都没想明白的设定——**这台机器根本不在海关停下**。你飞出国境，以为出了它的射程，其实它一路跟着你，跟到东京、跟到伦敦、跟到你在异国他乡开的那个银行账户里。

这一章，我们跟着它跨过国界。你会发现还是前面那三招——第二章的**源头截流**、第四章的**第三方替你打小报告**、第五章的**对账工厂逐行比对**——只是这一次，投影的屏幕从一个国家换成了整个地球。

第一件事：它认的是护照，不是邮编

先问一个最朴素的问题：一个国家凭什么向你收税？全世界几乎所有国家的答案都一样——凭你**住在这儿**。这叫 居住地征税 (residence-based taxation)：你住哪国，就给哪国交税；你搬走了、不住了，这国的税务局跟你就基本两清了。

美国是极少数的例外。它用的是 国籍征税 (citizenship-based taxation)。

大白话

说人话：只要你是美国公民，不管你住在地球哪个角落，哪怕你出生后一天都没在美国待过、一辈子住在东京，你每年都得向美国税务局申报，并且可能要就你在全世界挣的全部收入，向美国交税。全世界还认真这么干的国家，基本上只剩美国和一个叫厄立特里亚的小国。

用工程师的话说，这台机器的**寻址方式变了**。别国系统的节点集合是“住在本国境内的人”，寻址跟着邮编走——你搬出这个邮编，就被移出了节点表。美国这台机器的节点集合是“全世界拿美国护照的人”，寻址**跟着护照走，不跟着邮编走**。你人在地球背面，护照没扔，就还在表里。

两道减压阀：别把人两头收干

这里立刻有个问题：一个住东京的美国人，收入已经被日本收过一遍税了，美国再收一遍，那不成了两头收、把人扒两层皮吗？

为了不把人逼死，机器上装了两道减压阀。

第一道叫 海外劳动收入豁免（Foreign Earned Income Exclusion）。

大白话

说人话：你在海外挣的工资，有一大笔（近年大约十几万美元上下，每年微调，别咬死数字）可以直接不算进美国的应税收入里。等于机器说：“你在国外挣的辛苦钱，头十几万我不看。”

第二道叫 外国税收抵免（Foreign Tax Credit）。

大白话

说人话：你已经交给日本的税，可以拿来一块钱顶一块钱地抵掉你欠美国的税。你在高税率国家交得多，抵完往往就没美国的份了。

这两道阀一开，绝大多数海外美国人算到最后，其实补给美国的税没多少，甚至是零。但——这里是全章你最该记住的一句话：

“要不要交税”和“要不要申报”，是两码事。 税可能一分不用补，可申报义务一分都跑不掉。你每年照样得把全世界的收入、账户、投资，一份不落地报给美国。

顺带提一个稳的数字：美国人在海外的银行账户，只要合计余额超过约**一万美元**，每年就得单独交一份申报（这份表叫 FBAR，海外银行账户报告），跟报税分开走。忘了报、报晚了，罚起来极狠。

这套义务有多烦？烦到从 2010 年往后，每年都有**创纪录数量的美国人主动跑去大使馆放弃国籍**——不是不爱国，是实在受不了这台机器跨洋追着他们要表格。

第二件事：FATCA——把“打小报告”投到全球银行

问题来了：一个美国公民藏在瑞士的银行账户里，美国怎么知道？它又没法派人去翻瑞士银行的账本。

2010 年，美国立了一部法，把第四章那招“第三方替你作证”直接放大到全世界的银行头上。这法叫 FATCA（海外账户税收合规法案）。

大白话

说人话：美国对全世界的外国银行下了一道命令——你们自己去把客户里的“美国人”揪出来，然后把这些账户的信息，报给美国税务局。等于逼着外国银行，替美国站岗打小报告。

凭什么外国银行要听美国的？靠一根大棒。**不配合的外国银行，凡是从美国流向它的钱（利息、股息这类美国来源收入），美国先扣走 30%。**这个 30% 是稳的、写死在法里的。

你体会一下这句威胁的分量。美国等于对全世界的银行说：“把美国客户报上来。否则凡是流经你的、来自美国的钱，我先 skim 30% 走。”一家做全球生意的大银行，血管里天天流着美国市场的资金，被卡这一刀根本扛不住——只能就范。

落地靠什么？靠美国和**一百多个国家**签的 政府间协议（IGA）。很多国家干脆让本国银行先把美国人的账户报给**本国政府**，再由本国政府打包转交美国。相当于各国政府亲自当了美国收税机器的一段传送带。

看明白这是什么了吗？还是第四、五章那两招——第三方替你作证、机器逐行比对——只不过现在替你作证的“第三方”，是你海外的开户银行。**你在世界任何角落开的账户，都在替美国盯着你。**

副作用也来了：有些外国银行嫌美国客户太烫手、合规成本太高，索性挂个牌子——不收美国人开户。你拿着美国护照，反而在别国办不了张银行卡。

第三件事：CRS，和那个漂亮的反转

FATCA 太好使了，好使到别的国家看红了眼：“凭什么只有美国能逼全世界的银行交自己人的账？我们也想要。”

于是 2014 年前后，OECD（经济合作与发展组织，一个富国俱乐部）搞出了一套东西，叫 CRS（共同申报准则）。

大白话

说人话：CRS 就是把 FATCA 抄了一遍，但抄成了一个“人人平等、互相交换”的多边版本——“给所有人用的 FATCA”。一百多个国家和地区加入，各国的金融机构把“税务上算别国居民”的账户报给本国税务局，然后各国税务局之间**自动**交换这些信息（这套自动交换叫 AEOI）。

注意一个细节：CRS 认人用的是**税务居住地**——你税务上算哪国人，就报给哪国。这跟别国“居住地征税”的逻辑是一路的。全世界照着美国的样子，织了一张互相通报的大网。

然后，本章的高潮来了。

全世界照着美国的 FATCA，抄出了一个人人对等的 CRS——**唯独美国自己，没加入 CRS。**

美国的理由是：“我已经有 FATCA 了，不用你这套。”可关键在于，FATCA 基本是**单向**的：外国银行拼了命地把美国人报给美国，美国这边却基本不对等回报——美国的银行很少把外国人的账户信息反馈给对方国家，因为美国国会一直没通过那条“对等交出去”的立法。

信息只进不出。这就催生了一个黑色幽默的结果。

因为美国既置身 CRS 那张大网之外，自己又“只收不发”，它反而变成了全世界数一数二的**避税天堂、保密保险库**。别国的有钱人，把钱停在美国的银行、或者停进特拉华、南达科他、内华达这些对信托特别友好的州，他们本国的税务局几乎**看不见**这笔钱——因为美国不往外报。

发明了这场全球征税军备竞赛的国家，自己成了那个信息进得来、出不去的金库。

用工程视角收口，这个反转精确得可怕：CRS 是全世界把 FATCA 逆向成了一个“**人人对等广播**”的对称协议，每个节点又收又发；美国用的却是“**中心辐射**”式——它当那个只收不发的中心节点，所有边都指向它，没有一条从它指出去。

同一套机制，**协议的对称性一改，天堂和地狱就换了位置**。对外国人隐私是地狱的东西（银行必须上报），换个方向，就成了美国自己的避税天堂。

还是那三招，只是投到了全世界

回头看这一整章，你会发现没有一个新原理。全是老朋友：

- **源头截流**（第二章）——FATCA 那把 30% 的扣款大棒，就是在钱流经银行时先截一刀，逼它就范。
- **第三方替你作证**（第四章）——只不过这次的“第三方”，是你海外的开户银行，是别国的政府。
- **对账工厂逐行比对**（第五章）——IRS 收到全球报上来的账户信息，照样拿去跟你的申报逐行对，对不上照样发通知。

这台机器最狠的地方，不是它多复杂，而是它的三个核心零件**可以无限复制、随处投影**。美国境内怎么盯你，全世界就怎么盯你。它把国境这道墙，当成了不存在。

可是——一台连全球都罩得住、看起来滴水不漏的机器，为什么美国人自己报个税还能被逼疯？为什么它的核心跑着六十年前的代码？既然这么精巧，它到底哪儿还坏着？下一章，也是最后一章，我们把镜头拉到最远，退后一步，看这台奇迹般的机器身上，那些谁都看得见、却谁都修不动的毛病。

第十七章 · 这台机器的 bug 与演化

你已经跟着这台机器走了十六章。你看过钱怎么在到手之前就被截走，看过第三方怎么替你作证，看过审计的探照灯怎么让人自己算完账就不敢赌。现在我们退到最远的地方，把整台机器完整地看一眼——不是看它多精巧，而是看它**哪儿坏了**。

因为一台跑了一百多年、每年处理上亿份申报、还一天都不能停的系统，不可能是干净的。它一定带着满身的补丁、老旧的零件和被人塞进去的私货。这一章，我们就来拆它的三个 bug。拆完，你就能回答这本书从头到尾那个问题了。

Bug 一：报税为什么这么痛苦？答案不是技术，是游说

第九章我留了个钩子：为什么美国不像很多国家那样，直接把税表给你填好、你看一眼确认就完事？我说那是最后一章的事。现在到了。

先说清楚一件事：**技术上，这件事早就能做到了**。

你想想，第四章讲过，IRS 手里已经攥着你几乎所有的收入信息——你的 W-2（工资单）、1099（利息、股息、零工收入），雇主和银行早在你动笔之前就替你申报过一遍了。既然它都知道了，为什么不能直接算好、印在表上给你？

很多国家就是这么干的。这套做法有个名字，叫 return-free filing——直译是“免申报报税”。

大白话

说人话：政府把它已经知道的你的收入和该交的税全算好，寄给你或者放到网上。对绝大多数只有工资收入的人来说，你要做的就是看一眼数字对不对，点个“确认”。整个报税过程一分钟。

日本、英国、北欧都在用类似的东西。这不是什么黑科技，就是“数据我都有了，那我替你填”这么个再自然不过的自动化。

那美国为什么没有？为什么每年四月，几亿人还要在一堆表格、软件和会计费里挣扎？

答案很难看：**因为有一整个行业，靠你的痛苦活着。**

报税软件公司——Intuit (TurboTax 的东家)、H&R Block 这些——几十年来持续往华盛顿砸游说的钱，就为了拦住一件事：别让 IRS 自己给你提供免费的、预填好的报税服务。道理简单到残忍：**如果政府免费帮你把税报了，谁还花钱买 TurboTax？**

这里有个工程师一听就会气笑的操作。2000 年代初，这些公司跟 IRS 谈了一个叫 Free File 的协议。

大白话

表面上：软件公司答应给低收入人群提供免费报税版本。听起来像做好事。

暗地里的交换条件：作为回报，IRS 承诺自己不去开发一个官方的免费报税产品。

你品品这个交易。IRS 用“我保证不自己做”，换来了行业“我给穷人做点免费的”。结果呢？后来有调查发现，这些公司还想方设法把免费版藏起来——比如给免费报税页面加代码，让它不被谷歌搜到，好把用户导流到收费产品上去。

这就是这个 bug 最精髓的地方。用工程师的话说：

一个本来可以自动化的手动流程，被靠这个流程吃饭的供应商，用游说硬生生锁死在了手动状态。

报税越复杂，卖软件的越赚钱。所以他们没有任何动力让它变简单——恰恰相反，他们要花钱确保它**永远别变简单**。系统的一个 bug，被既得利益者精心养护成了自己的护城河。

近几年 IRS 终于自己下场，试点了一个官方免费报税工具，叫 Direct File（政府自建的、直接在 IRS 网站上免费报税，不经过任何第三方软件）。你猜怎么着？话音刚落，同一批公司的游说火力就猛烈反扑上来了。这场仗到今天还在打。

所以下次你被报税折磨，记住：折磨你的不是技术难度，是有人花钱把简单的按钮拆掉了。

Bug 二：全世界最贵的一坨技术债

第二个 bug，工程师会更有共鸣，因为你八成正维护着一个类似的东西。

这台每年抽走几万亿、处理上亿份申报的机器，它的核心系统有多老？老到你不敢信。

IRS 的心脏叫 IMF，Individual Master File，个人主文件。

大白话

说人话：这是一个大数据库，装着每一个纳税人的账户——你交了多少、欠多少、退多少，全在里面。它是整台机器的账本。

这个账本，是 1960 年代建起来的，跑在 COBOL 和汇编语言上。

大白话

COBOL 是上世纪五六十年代的老编程语言。今天几乎没有年轻程序员会写它，很多懂它的人早就退休甚至去世了。汇编就更底层、更古老。

一个撑着现代国家财政命脉的关键系统，用的是比登月还早的技术，代码里的逻辑比大多数在世的工程师年纪都大。这不是段子，这是 IRS 被审计报告反复点名的现实——它长期是整个美国联邦政府里最古老的 IT 系统之一。

为什么不重写？谁都知道该重写。IRS 的现代化项目立了一次又一次，拖延、超支、再拖延，几十年了还没真正换掉那个内核。

原因你已经猜到了，因为它**不能停机**。

这是全世界最贵的一坨技术债。谁都知道该重写，可这机器一天都停不得——每年几万亿要从它身上流过，几亿人的退税等着它算。你不能对一个还在收着几万亿的系统说“我们下线三个月，重构一下”。

这就是“不能停机的遗留系统”最教科书的样子：坏，但活着；旧，但要命；所有人都想动它，所有人都不敢动它。

而且技术债从来不只是技术问题。IRS 长期预算不足，钱不够，系统就更新不动，人也招不够。别忘了第六章讲的审计威慑——那台探照灯是靠人和算力撑起来的。**预算一削，审计能力就下降；审计一松，作弊的期望收益就回正，遵从率迟早跟着滑坡。**老旧的系统和被砍的预算，是从底座上啃食着这台机器的威慑力。

Bug 三：税法是几十年补丁叠出来的一坨

第三个 bug，是前两个的总根子。

美国税法为什么那么厚、那么绕、那么多莫名其妙的特例？因为它从来没被“重构”过。它是一台运行中的机器，每一次改动都是**热更新**——在不关机的前提下往上打补丁。

大白话

热更新：系统不停机，直接在运行状态下打补丁改代码。好处是不中断服务，坏处是你没法推倒重来，只能在旧结构上层层叠加，越叠越乱。

你翻翻历史就懂了。1986 年里根搞过一次大税改，口号是简化，砍掉一堆漏洞，把税率档位削平——那是难得的一次“想把代码理干净”。可理干净之后呢？后面几十年，每一届政府、每一个利益集团，又往里塞新的抵免、新的减免、新的特例。到 2017 年的 TCJA（那次大减税法案）又是一轮大改动，加进去一堆新规则、新例外。

每一个抵免、每一个漏洞的堵与开，都是给运行中的机器打的一个补丁。补丁摞补丁，没人敢推倒重来——因为推倒重来意味着停机，意味着触碰无数条已经把自己利益刻进法条里的势力。于是税法就长成了今天这样：极度复杂，充满了利益集团亲手凿进去的特例。

第十二章我还留了个钩子：跨国公司怎么把利润“搬”到低税率的地方去避税？现在也能收了。那种利润转移，恰恰长在**各国规则的接缝里**——每个国家的税法都是各自打补丁打出来的，接缝对不齐，中间就露出了缝隙，聪明的会计师专门往缝里钻。

各国当然也在打补丁堵缝，比如这些年尝试搞的“全球最低企业税”，想给这场逃逸设个地板。但你堵一个缝，对方就找下一个缝——这是一场没有终点的**军备竞赛**：一方修漏洞，一方找漏洞，永远慢半拍，永远停不下来。

回望：三根柱子，一个奇迹，一台修不动的机器

好，机器拆完了。现在回到最开始那个问题——这本书从第一页就在问的：

一个不派人上门收钱、主要靠你自己填表申报的国家，凭什么能从三亿人和几百万家公司手里，合法、准时、几乎不出错地，抽走每年几万亿美元？

读到这里你应该已经能自己回答了。答案不是美国人觉悟高、不是大家爱国到主动纳税。答案是**三个冷静的工程设计**，三根柱子，我们再点一次题：

1. **源头预扣**——钱还没到你手里，就在雇主那一环被截走了。你没机会赖，因为你根本没碰到过那笔钱（第二、三章）。
2. **信息申报交叉核对**——第三方（雇主、银行、券商）早替你把收入报给了 IRS，机器再逐行比对。系统不需要“相信”你，它有别人替你作的证，一对账就知道你有没有说谎（第四、五章）。
3. **审计威慑**——万一你真想赌一把，那台随机探照灯和重罚，让作弊的期望收益是负的。你自己算完账，就不赌了（第六章）。

预扣让你没机会赖，信息申报让系统不必信你，审计威慑让你自己劝退自己。三根柱子一搭，“自愿”两个字就成立了——它是被设计出来的自愿，不是天生的自愿。这，就是这本书标题里那张“三亿人的自愿账单”真正的答案。

但今天我们也看清了它的另一面。这台机器高效、精巧，同时又臃肿、老旧、被利益集团扭曲。它一边是人类组织能力的一个奇迹——让三亿人自愿掏钱、还几乎不出错；一边是一台谁都修不动的遗留系统——报税痛苦被游说锁死，内核跑着六十年前的代码，税法是几十年补丁叠成的一坨。

最后，给你留一点工程师才尝得出的余味。

你有没有发现，这三个 bug 其实不是税务独有的？**任何一个足够大、运行足够久、又一天都不能停机的系统，最终都会长成这个样子。**精巧和腐坏并存，奇迹和技术债同居，所有人都明知道哪儿坏了，却只能在它还带病运行、还在收着几万亿的前提下，继续小心翼翼地打下一个补丁。

你维护过的那个老系统是这样，你所在的这个社会的很多制度也是这样。税务机器的特别之处，只是它把这条规律，用几万亿美元和三亿人的年复一年，演给你看得格外清楚而已。

下次四月你坐在电脑前，对着那份表格叹气的时候——希望你叹的不再只是“真麻烦”，而是“哦，我看懂这台机器了”。

三亿人的自愿账单 · 王建硕