

终于大功告成!

由于命令很长, 在这里向大家推荐一个更容易操作的启动方法。读者可以借助笔者的 GitHub 工程, 直接下载, 在工程内部有现成的创世块文件和启动脚本, 操作起来更简单, 步骤如下。

步骤 01: 下载 GitHub 工程, 命令行如下。

```
mkdir ~/ethdev
cd ~/ethdev
git clone https://github.com/yekai1003/rungeth
```

步骤 02: 进入该目录, 进行初始化, 命令行如下。

```
cd rungeth
geth init genesis.json --datadir ./data
```

步骤 03: 执行脚本 “./rungeth.sh”, 启动 Geth, 结果如下。

```
root:rungeth yk$ ./rungeth.sh
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.6-stable/darwin-amd64/go1.13.1
at block: 0 (Thu, 01 Jan 1970 08:00:00 CST)
  datadir: /Users/yk/ethdev/yekai1003/rungeth/data
  modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0
personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
```

至此, 搞定了节点, 可以准备智能合约的开发了。

## 4.1.2 初识Solidity

由于目前支持智能合约的区块链平台很多, 智能合约的开发语言也有多种选择, 不过, 以太坊毕竟是第一个诞生智能合约的区块链平台, 之后产生的很多区块链平台也多是参考以太坊平台的虚拟机技术, 借鉴以太坊的智能合约开发环境。以太坊平台智能合约开发语言主要采用 Solidity。因此 Solidity 也是目前多数主流区块链平台所采用的智能合约开发语言。

Solidity 是一门面向对象、为实现智能合约而创建的高级编程语言, 这门语言受到了 C++、Python 和 Javascript 等语言的影响。在以太坊黄皮书披露的技术细节中, 提到了以太坊虚拟机是一款图灵完备 (Turing Completeness) 的虚拟机器, Solidity 自然也就是一款图灵完备的高级开发语言。它的内部可以支持变量定义、容器、自定义类型、函数、循环、继承等高级语言的通用技术, 在后面的内容中我们将逐渐展开介绍。

按照惯例, 学习一门语言的第一个程序是介绍 “hello-world”, 这就当是和 Solidity 的初识吧。

```
pragma solidity^0.6.0;
```