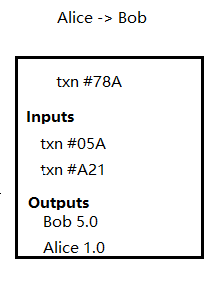
比特币源码研读系列2 交易和区块链的数据结构

上一篇我们讲到了比特币代码的执行初始化，本篇开始我们便来介绍下比特币中最重要的一个概念：Transaction，中文称交易，我们简写为tx。

在比特币中，Tx就是coin货币，花掉的每个coin都可以在某个tx中找到，挖矿得到的还是别人付款的coin也一定可以在某个tx中找到，tx记录着coin的所属权。已经得到公认的tx都是保存在block中，为了确保block之间的时间顺序，引入了blockchain区块链技术。所以可以说，所有的一切都是以tx为基础。

了解一些主要的数据结构有助于我们阅读源码，因此本文我们先来介绍下一些重要类的结构。

**一 tx的结构**

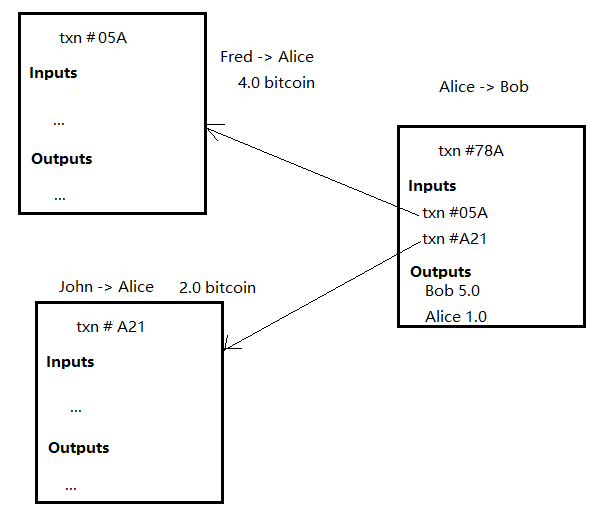
通过之前的文章我们大致也已经了解到了，tx主要结构如下：

其中inputs代表本次打算要花掉的coin，而outputs为收款人和找零。我们看看在代码中Tx的数据结构：

**CTransaction**

|  |
| --- |
| vector<CTxIn> vin |
| vector<CTxOut> vout |
| int32\_t nVersion |
| uint32\_t nLockTime(BIP113) |
| uint256 hash(memory only) |

Vin就是我们所说的inputs，它是vector数组，其中的每个元素CTxIn就是引用的每个coin。



我们看看CTxIn都包含有哪些信息：

|  |
| --- |
| **CTxIn** |
| COutPoint prevout; |
| CScript scriptSig; |
| uint32\_t nSequence; |
| CScriptWitness scriptWitness |

其中scriptSig就是签名后的字段，签名的意思就相当于信用卡刷卡时签字确认一样。从这个结构可以看出来比特币是对每个花掉的coin都进行签名，而不是对整个账单签名。

nSequence目前似乎并没有用，后面我们会看到nSequence通常是和nLockTime一起使用。scriptWitness是见证（witness）脚本，见证大致是比特币改进建议BIP141（Bitcoin Improvement Proposal）里引入的东西，大致就是用于定时触发BIP141更新用。

Prevout结构如下

|  |
| --- |
| **COutPoint** |
| uint256 hash;(doubled SHA256-hashed of previous to-be-used transaction) |
| uint32\_t n;(index) |

其中hash为引用的tx的hash值，n为coin在引入的tx中outputs中的位置。

我们继续看CTransaction的vout，它就是我们说的outputs，保存着收款人和收款金额，以及找零。同样vout也是个vector数组，我们来看看CTxOut结构

|  |
| --- |
| **CTxOut** |
| CAmount nValue;（ typedef int64\_t Camount） |
| CScript scriptPubKey; |

其中nValue就是收款金额，scriptPubKey是用收款人的公钥加密生成的，能解开就说明该coin是属于我的。为了避免其他人在没有解开scriptPubKey时就声称该coin属于自己并且引用到新的tx，因此当我们从网络中收到tx时我们还需要校验tx的vin中的tx，检查他花掉的coin是否是属于他的，这些在接下来的代码中都可以看到。

看完vout，tx的主要结构就看完了，下面的nVersion代表tx的版本号，nLockTime代表锁定时间，nLockTime值是tx可以被打包进block的最早时间，通常tx在创建时都会有个大约5min左右的冷却期，在这个时间段内是不能参与生成block的。

**二 tx的状态**

|  |
| --- |
| **Status** |
| enum Status { |
| Confirmed, |
| /// Normal (sent/received) transactions |
| OpenUntilDate, |
| OpenUntilBlock, |
| Offline, |
| Unconfirmed, |
| Confirming, |
| Conflicted, |
| Abandoned, |
| /// Generated (mined) transactions |
| Immature, |
| MaturesWarning, |
| NotAccepted |
| }; |

TransactionStatus类记录着tx的状态，

confirmed指的是已经在block中且不可更改的tx，

OpenUntilDate指的是tx的冷却期结束时间点，

OpenUntilBlock同样指的是tx的冷却期结束block number，

Offline表示尚未广播出去，

Unconfirmed尚未加入到block中，

Confirming已经在block中，但是成熟度不够，成熟度的意思是是否可更改，在比特币中通常是认为如果tx所在block前面有超过6个block则认为该block是成熟的不可更改的，这是为了防止double-pay的情况。

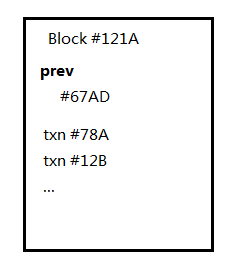
Conflicted和之前的tx有冲突，

Abandoned表示该tx已丢弃，

Immature，MaturesWarning和NotAccepted应该指的都是coinbase的状态。

**三 Block结构CBlock**

|  |
| --- |
| 前面我们已经介绍过block大致如下图所示：  现在我们看看真实的结构是什么样的：  **class CBlock : public CBlockHeader** |
| vector<CTransactionRef> vtx;(network and disk) |
| mutable bool fChecked;(memory only) |

可以看出来CBlock结构很简单，就是一个数组，里面保存着tx的引用值CTransactionRef，那blockheader有哪些数据呢？

|  |
| --- |
| **class CBlockHeader** |
| int32\_t nVersion; |
| uint256 hashPrevBlock; |
| uint256 hashMerkleRoot; |
| uint32\_t nTime; |
| uint32\_t nBits; |
| uint32\_t nNonce; |

原来block的重要信息都在blockheader中，其中nVersion代表版本号，hashPrevBlock是指向前一个block，hashMerkleRoot是block中的tx的merkle树根(merkle树是用于简单验证某个tx是否在该block中而使用的一种二叉树结构，后面我们会专门来讲，这里只需要知道如果去掉merkle也不会影响block，只是查找时慢一点而已)，nTime指的是block生成的unix time，nBits和nNonce都是用于PoW（proof of work，也就是前面提到的不断计算block的id找到一个满足条件的blockId）。

到此我们可以发现block中的所有tx都是在vtx数组中。

**四 区块链blockchain结构**

Blockchain就是一个block的链条，我们看看它的结构

|  |
| --- |
| **class CChain** |
| vector<CBlockIndex\*> vChain; |

可以看到果然如我们所想的那样，我们看到CBlockIndex有哪些信息：

|  |
| --- |
| **class CBlockIndex** |
| const uint256\* phashBlock; |
| CBlockIndex\* pprev; |
| CBlockIndex\* pskip; |
| int nHeight;//! height of the entry in the chain. The genesis block has height 0 |
| int nFile;//! Which # file this block is stored in (blk?????.dat) |
| unsigned int nDataPos; |
| unsigned int nUndoPos; |
| arith\_uint256 nChainWork; |
| unsigned int nTx; |
| unsigned int nChainTx; |
| uint32\_t nStatus; |
| int32\_t nVersion; |
| uint256 hashMerkleRoot; |
| uint32\_t nTime; |
| uint32\_t nBits; |
| uint32\_t nNonce; |
| int32\_t nSequenceId; |
| unsigned int nTimeMax; |

从这里就可以看到每个block的重要信息都保存在这里了，这里我就解释几个重要的字段了：phashBlock是block的hash值，

pprev指向前一个block，

nHeight代表在blockchain的高度或者也称之为深度，其中创世块的深度为0，

nFile表示block的具体信息保存在哪个数据文件里，nDataPos和nUndoPos应该都与文件中的位置有关。

其他一些字段都是辅助字段，遇到时我们再讲。

好了，本篇我们就介绍到这里了，下一篇我们真的要来介绍下tx是如何创建的…