10 比特币分叉

fork分叉

分叉可能是多种原因造成的：

1. 同时挖到了区块 这种情况称之为state fork，由于对比特币当前状态出现了分歧而导致了分叉

   我们之前讲过的forking attack就是state fork的一种，也是发生了分歧，只不过这种分歧是人为造成的。也称为deliberate fork

2. 第二种情况就是升级软件。在一个去中心化的系统中，没有办法保证所有的节点都升级系统。升级系统就是修改比特币的协议，但有可能社区中有一些节点不同意修改协议，就会不升级。这个时候就会出现分叉，支持新协议的是一个分叉，不支持新协议的是一个分叉。这种情况称为protocol fork。通过对协议修改的内容的不同，可以进一步区分硬分叉hard fork和软分叉soft fork。

   1. hard fork：出现硬分叉的情况就是对比特币协议修改增加一些新的功能，这时候没有接受新协议的节点不会接受这些新的功能而认为他是非法的。这个时候就会出现两个分叉。举例来说，在实际情况中，大家对于区块的大小就存在分歧。有些人认为一个区块1M的字节大小太小了。1MB / (250B/transaction) = 4000 transaction

      4000 / (10mins * 60s/min) = 7 transactions/seconds

      这样的throughput太小了，而且限制了比特币交易的流通。

      假设有人把比特币区块的大小从1MB -> 4 MB，假设大部分的节点接受了这个新协议。这个大部分节点是按照算力来算多少的，而不是按照账户的数量来算多少。当系统中大部分的算力接受了新的协议，那么新的区块的链就会成为最长合法链。新的分叉也会接受老的版本的区块。但是旧的版本的节点不会接受新的区块，所以认为新的链就是非法的，从而沿着之前的链继续挖。但是因为大部分的算力都接受新的区块，所以新的分叉会变成最长合法链。分链之后就是各记各的账，但是账户的公私钥对还是一致的，所以会有问题。所以还会加一个chain id，来著名是哪个链。

   2. soft fork：比如说我把block size改小了，变成0.5M。假设大部分的节点认可了新协议，就会开始挖小的区块。但是旧版本协议的节点也会认小的区块，所以会把算力更强的节点挖出来的小区块形成的最长链也认为是合法的。所以这种分叉是临时的。但如果旧版本协议的节点一直不升级协议，可能会导致他挖出来的大区块都白挖了。实际中可能出现的情况就是新版本的协议给某些当前版本协议中的域一些新的含义。比如说铸币交易中的coinbase域，没有实际含义，但是可以作为nonce的额外的长度。实际当中会把coinbase域的前8个byte也作为额外的nonce（只有4个bytes）长度。但是coinbase剩下的bytes还没有用途，有人提议把它作为UTXO的根哈希值。思考一个场景：你手机上的比特币钱包（一个轻节点），想要查账户的余额，需要向全节点发出请求，因为轻节点自己本身并不维护这个链的结构，所以没有办法算出来余额。但问题是当请求发给全节点之后，全节点返回的余额轻节点怎么才能验证是不是对的呢？所以有人提议把UTXO中的内容也变成一个Merkle Tree，然后把根哈希写到coinbase里的某个位置。然后因为coinbase的内容会影响这个区块的哈希值的变化，所以我们就可以用merkle proof进行证明。这也是一种软分叉，因为旧节点会认可新节点。

      比特币历史上比较著名的软分叉就是P2SH。这个功能在最初的比特币协议里是没有的，而是后面通过软分叉的方式加进去的。

soft fork：只要系统中的大部分节点更新了协议，就不会出现永久分叉

hard fork：系统中的全部节点更新了协议，才不会出现永久行的分叉