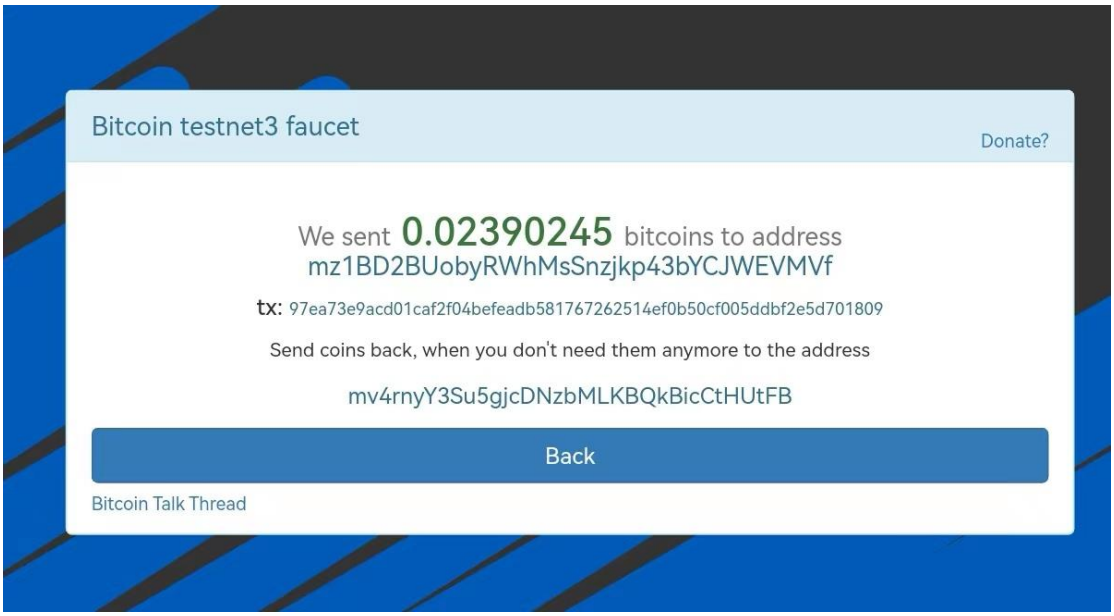
区块链第一次实验报告

2013536 汤清云

【实验过程】

1. 使用 keygen.py 生成一个 testnet 私钥和地址，在 faucet（https://coinfaucet.eu/en/btctestnet/）粘贴得到的地址，得到一些 testnet BTC：



Private key:

cQf8sE9fhMtF6gBCv8Kb5Q7cPgcDvUm4VDGc3TDRKKiEbRekKdap

Address:

mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf

交易哈希值：

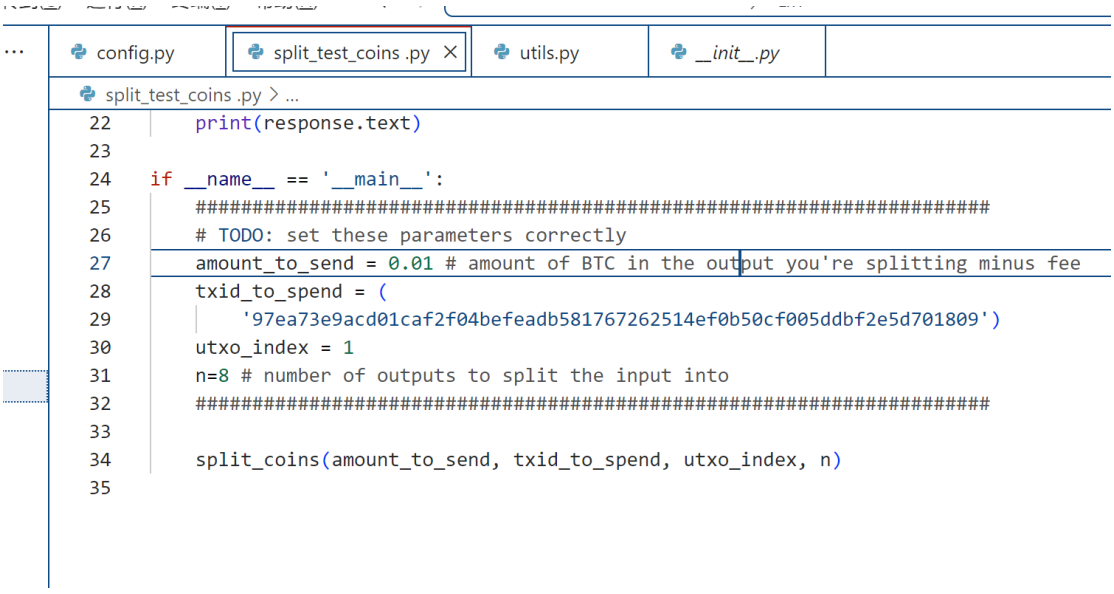
97ea73e9acd01caf2f04befeadb581767262514ef0b50cf005ddbf2e5d701809

2. 将上一步得到的交易哈希值在 www.blockchain.com/btc-testnet 网址中查询交易，可以得到结果如下

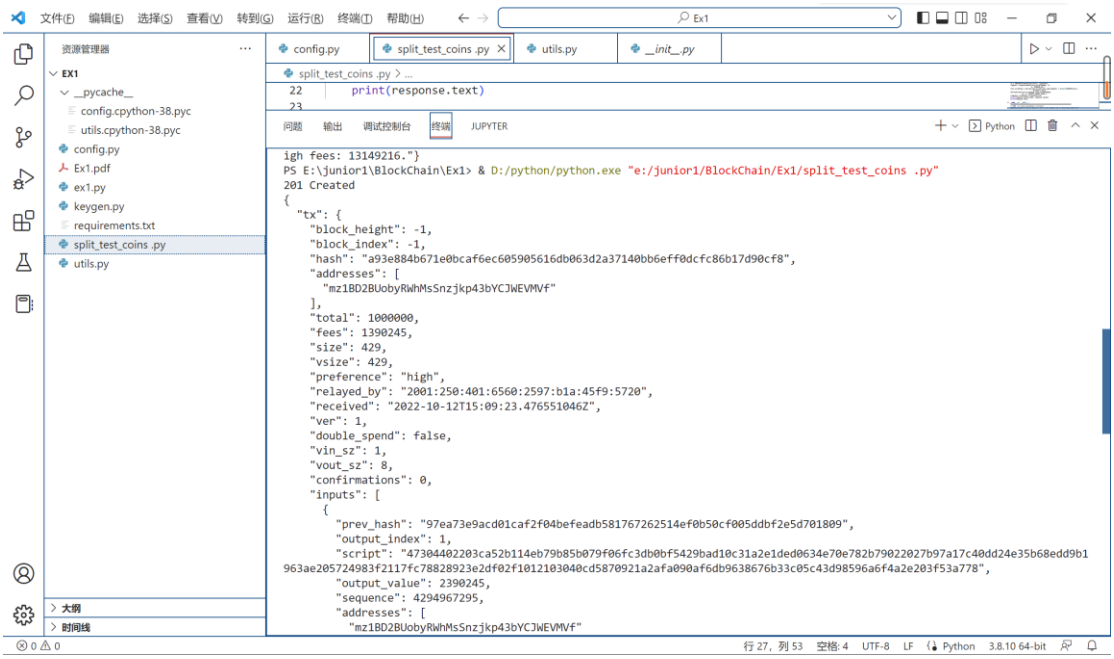
3. 修改 config.py 文件，将 my_private_key 修改为第一步中得到的私钥。

```
5
6   SelectParams('testnet')
7
8   # TODO: Fill this in with your private key.
9   my_private_key = CBitcoinSecret(
10      'cQf8sE9fhMtF6gBCv8Kb5Q7cPgcDvUm4VDGc3TDRKKiEbRekKdap')
11  my_public_key = my_private_key.pub
12  my_address = P2PKHBitcoinAddress.from_pubkey(my_public_key)
13
14  faucet_address = CBitcoinAddress('mv4rnyY3Su5gjcDNzbMLKBQkBicCtHUtFB')
15
```

4. 修改 split_test_coins.py 文件，txid 为第二步中得到的哈希值；将第一步中得到的比特币中的 0.01 平均分为 8 份（n），使用第二个输出（utxo_index=1）



运行结果：

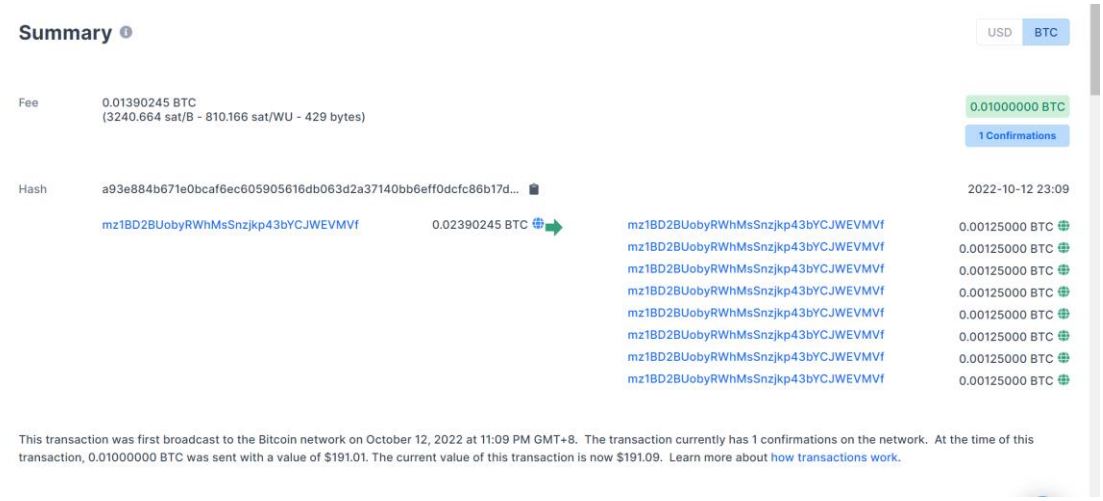


201 Created
{
"tx": {
"block_height": -1,
"block_index": -1,
"hash":
"a93e884b671e0bcaf6ec605905616db063d2a37140bb6eff0dcfc86b17d90cf8",
"addresses": [
"mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf"
],
"total": 1000000,
"fees": 1390245,
"size": 429,
"vsize": 429,
"preference": "high",
"relayed_by": "2001:250:401:6560:2597:b1a:45f9:5720",
"received": "2022-10-12T15:09:23.476551046Z",
"ver": 1,
"double_spend": false,
"vin_sz": 1,
"vout_sz": 8,
"confirmations": 0,
"inputs": [
{
"prev_hash":
"97ea73e9acd01caf2f04befeadb581767262514ef0b50cf005ddbf2e5d701809",

"output_index": 1,
"script":
"47304402203ca52b114eb79b85b079f06fc3db0bf5429bad10c31a2e1ded0634e70e782b79022027b97a17c40dd24e35b68edd9b1963ae205724983f2117fc78828923e2df02f1012103040cd5870921a2afa090af6db9638676b33c05c43d98596a6f4a2e203f53a778",
"output_value": 2390245,
"sequence": 4294967295,
"addresses": [
"mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf"
],
"script_type": "pay-to-pubkey-hash",
"age": 2350553
}
],
"outputs": [
{
"value": 125000,
"script":
"76a914cac8b0c1c960ee6b2771613dcb3ea962f7575f8c88ac",
"addresses": [
"mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf"
],
"script_type": "pay-to-pubkey-hash"
},
{
"value": 125000,
"script":
"76a914cac8b0c1c960ee6b2771613dcb3ea962f7575f8c88ac",
"addresses": [
"mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf"
],
"script_type": "pay-to-pubkey-hash"
},
{
"value": 125000,
"script":
"76a914cac8b0c1c960ee6b2771613dcb3ea962f7575f8c88ac",
"addresses": [
"mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf"
],
"script_type": "pay-to-pubkey-hash"
},
{

```
            "value": 125000,
            "script":
"76a914cac8b0c1c960ee6b2771613dcb3ea962f7575f8c88ac",
            "addresses": [
                "mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf"
            ],
            "script_type": "pay-to-pubkey-hash"
        }
    ]
  }
}
```


按照该哈希值进行交易查询，结果如下：



分币成功交易截图：

Hash	a93e884b671e0bcaf6ec605905616db063d2a37140bb6eff0dcfc86b17d90cf8
Status	Confirmed
Received Time	2022-10-12 23:09
Size	429 bytes
Weight	1,716
Included in Block	2350569
Confirmations	7
Total Input	0.02390245 BTC
Total Output	0.01000000 BTC
Fees	0.01390245 BTC
Fee per byte	3240.664 sat/B
Fee per vbyte	N/A
Fee per weight unit	810.166 sat/WU
Value when transacted	$191.01

5. 修改 ex1.py 文件，具体修改如下：

```
7
8   def P2PKH_scriptPubKey(address): #用来获取地址对应的公钥脚本
9       ###############################################################################
0       # TODO: Complete the standard scriptPubKey implementation for a
1       # 完成a的标准scriptPubKey实现
2       # PayToPublicKeyHash transaction 付款给对方的Public Key Hash
3       return [OP_DUP, OP_HASH160, address, OP_EQUALVERIFY, OP_CHECKSIG]
4       #复制栈顶元素（即压入公钥）
5       #将栈顶元素取出并哈希后重新压入
6       #将公钥的哈希值压入栈
7       #弹出栈顶的两个元素，比较它们是否相等
8       #用公钥PubKey检查一下签名Sig是否正确
9       ###############################################################################
0
21
22  def P2PKH_scriptSig(txin, txout, txin_scriptPubKey):
23      signature = create_OP_CHECKSIG_signature(txin, txout, txin_scriptPubKey,
24                                               my_private_key)
25      ###############################################################################
26      # TODO: Complete this script to unlock the BTC that was sent to you
27      # in the PayToPublicKeyHash transaction. You may need to use variables
28      # that are globally defined.
29      # 完成此脚本以解锁paytoppublickeyhash事务中发送给您的BTC。您可能需要使用全局定义的变量。
30      return [signature,my_public_key]
31
32      ###############################################################################
33
```

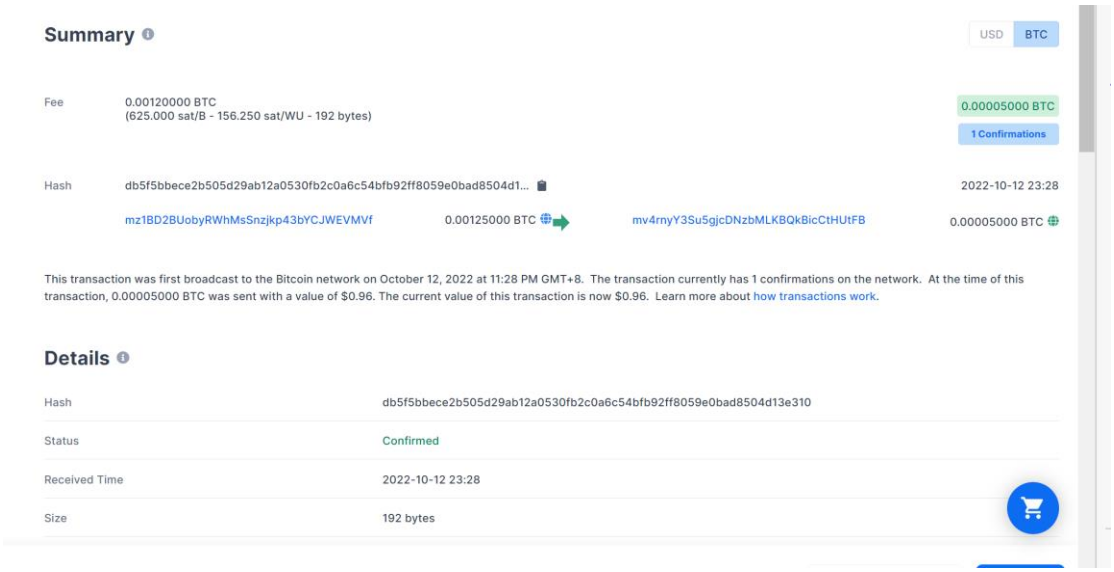
6. 取出第一份进行试验（故设置 utxo_index=0），运行如下：

```
}
PS E:\junior1\BlockChain\Ex1> & D:/python/python.exe e:/junior1/BlockChain/Ex1/ex1.py
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "db5f5bbece2b505d29ab12a0530fb2c0a6c54bfb92ff8059e0bad8504d13e310",
    "addresses": [
      "mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf",
      "mv4rnyY3Su5gjcDNzbMLKBQkBicCtHUtFB"
    ],
    "total": 5000,
    "fees": 120000,
    "size": 192,
    "vsize": 192,
    "preference": "high",
    "relayed_by": "2001:250:401:6560:2597:b1a:45f9:5720",
        "addresses": [
          "mv4rnyY3Su5gjcDNzbMLKBQkBicCtHUtFB"
        ],
        "script_type": "pay-to-pubkey-hash"
      }
    ]
  }
}
PS E:\junior1\BlockChain\Ex1>
```

201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,

```
        "hash":
"db5f5bbece2b505d29ab12a0530fb2c0a6c54bfb92ff8059e0bad8504d13e3
10",
        "addresses": [
          "mz1BD2BUobyRWhMsSnzjkp43bYCJWEVMVf",
          "mv4rnyY3Su5gjcDNzbMLKBQkBicCtHUtFB"
        ],
        "total": 5000,
        ]
    }
}
```


7. 查询该交易结果：



发币成功截图：