Bitcoin Scripting Assignment Report CS 216:

Introduction to Blockchain Assignment 2: Bitcoin Scripting

Team Name: BlockForge

Team Members:

- Hardik Bansal (230001031)
- Shorya Kshettry (230003070)
- Yash Singh (230051019)

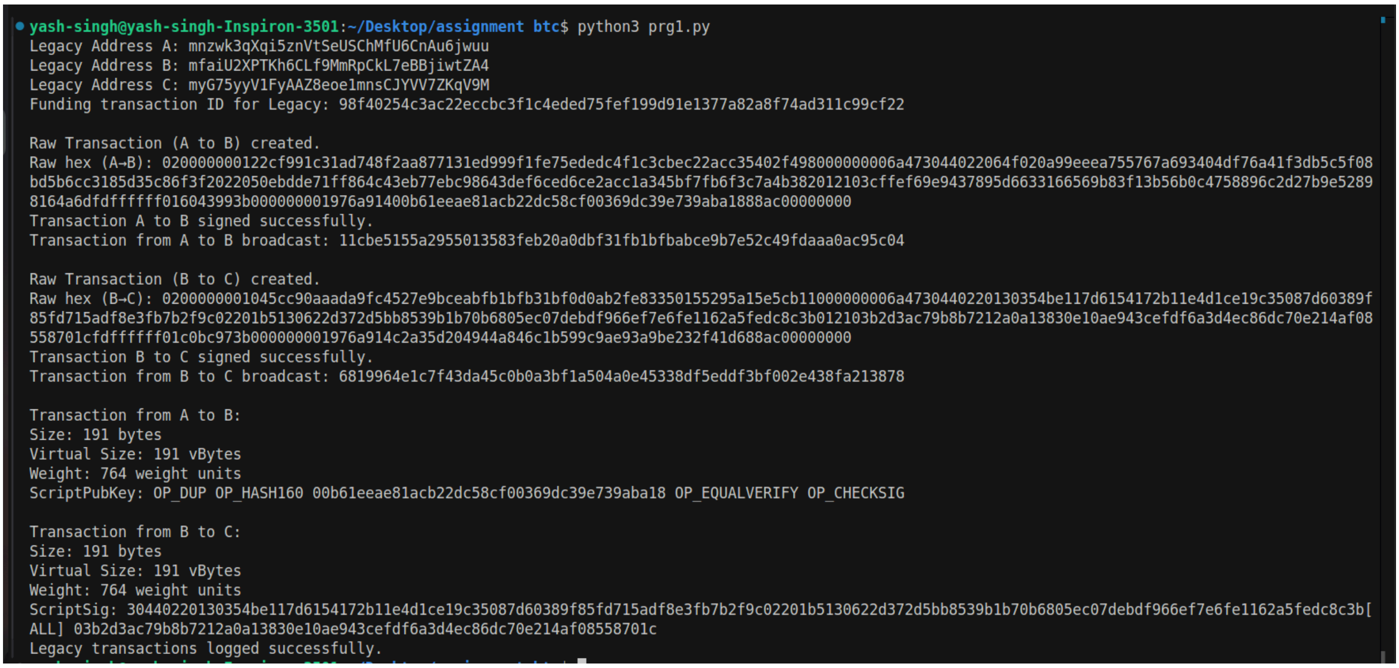
Introduction

The objective of this assignment is to understand the process of creating and validating Bitcoin transactions using both Legacy (P2PKH) and SegWit (P2SH-P2WPKH) address formats. This report outlines the implementation details, transaction analysis, and comparison of transaction sizes between these formats.

Legacy Address Transactions (P2PKH)

- Workflow
  - **Wallet Setup**: Connected to bitcoind via RPC and created a new wallet.
  - **Address Generation**: Generated three legacy addresses (A, B, C).
  - **Funding Address A**: Sent Bitcoin to address A using `sendtoaddress`.
  - **Transaction A → B**:
    - Created a raw transaction from A to B.
    - Decoded the transaction and extracted `ScriptPubKey` for B.
    - Signed the transaction and broadcasted it.
- **Transaction B → C**:
  - Used `listunspent` to obtain the txid from A to B.
  - Created and broadcasted a transaction from B to C.

- **Transaction B → C**:
    - Used `listunspent` to obtain the txid from A to B.
    - Created and broadcasted a transaction from B to C.



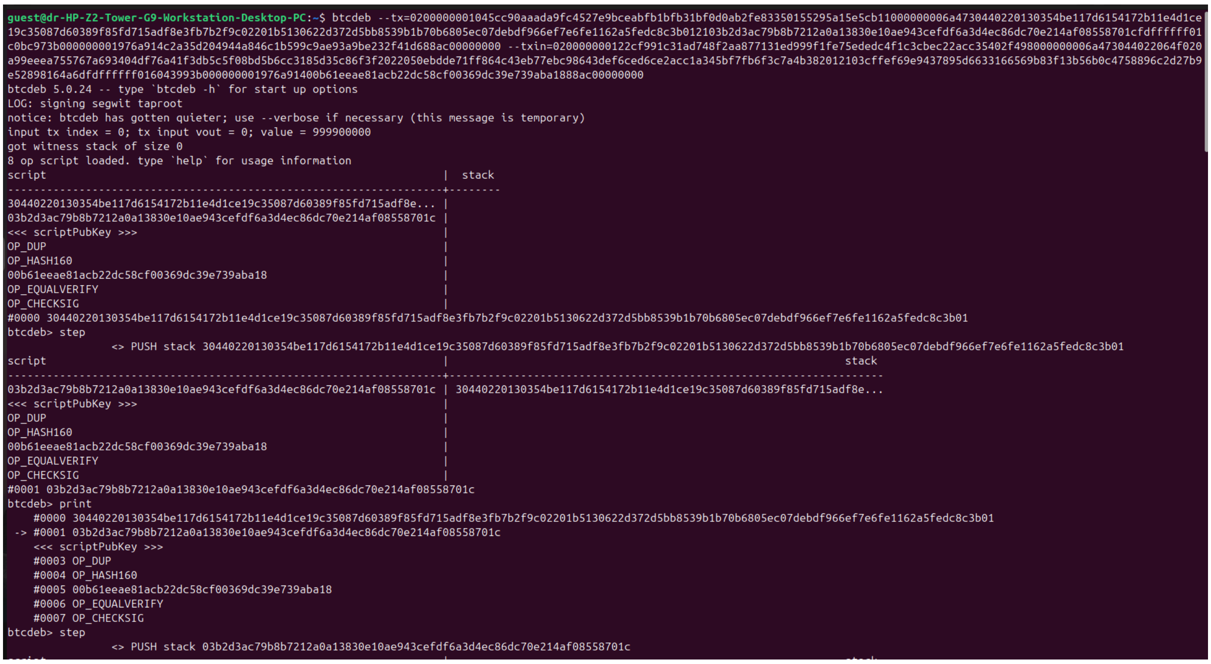
```
yash-singh@yash-singh-Inspiron-3501:~/Desktop/assignment btc$ python3 prg1.py
Legacy Address A: mnzwk3qXqi5znVtSeUSChMfU6CnAu6jwuu
Legacy Address B: mfaiU2XPTKh6CLf9MmRpCkL7eBBjiwtZA4
Legacy Address C: myG75yyV1FyAAZ8eoe1mnsCJYVV7ZKqV9M
Funding transaction ID for Legacy: 98f40254c3ac22eccbc3f1c4eded75fef199d91e1377a82a8f74ad311c99cf22

Raw Transaction (A to B) created.
Raw hex (A→B): 020000000122cf991c31ad748f2aa877131ed999f1fe75ededc4f1c3cbec22acc35402f498000000006a473044022064f020a99eeea755767a693404df76a41f3db5c5f08
bd5b6cc3185d35c86f3f2022050ebdde71ff864c43eb77ebc98643def6ced6ce2acc1a345bf7fb6f3c7a4b382012103cffef69e9437895d6633166569b83f13b56b0c4758896c2d27b9e5289
8164a6dfdffffff016043993b000000001976a91400b61eeae81acb22dc58cf00369dc39e739aba1888ac00000000
Transaction A to B signed successfully.
Transaction from A to B broadcast: 11cbe5155a2955013583feb20a0dbf31fb1bfbabce9b7e52c49fdaaa0ac95c04

Raw Transaction (B to C) created.
Raw hex (B→C): 0200000001045cc90aaada9fc4527e9bceabfb1bfb31bf0d0ab2fe83350155295a15e5cb11000000006a4730440220130354be117d6154172b11e4d1ce19c35087d60389f
85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b012103b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08
558701cfdffffff01c0bc973b000000001976a914c2a35d204944a846c1b599c9ae93a9be232f41d688ac00000000
Transaction B to C signed successfully.
Transaction from B to C broadcast: 6819964e1c7f43da45c0b0a3bf1a504a0e45338df5eddf3bf002e438fa213878

Transaction from A to B:
Size: 191 bytes
Virtual Size: 191 vBytes
Weight: 764 weight units
ScriptPubKey: OP_DUP OP_HASH160 00b61eeae81acb22dc58cf00369dc39e739aba18 OP_EQUALVERIFY OP_CHECKSIG

Transaction from B to C:
Size: 191 bytes
Virtual Size: 191 vBytes
Weight: 764 weight units
ScriptSig: 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b[
ALL] 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
Legacy transactions logged successfully.
```

- ## Challenge and Response Script Analysis:

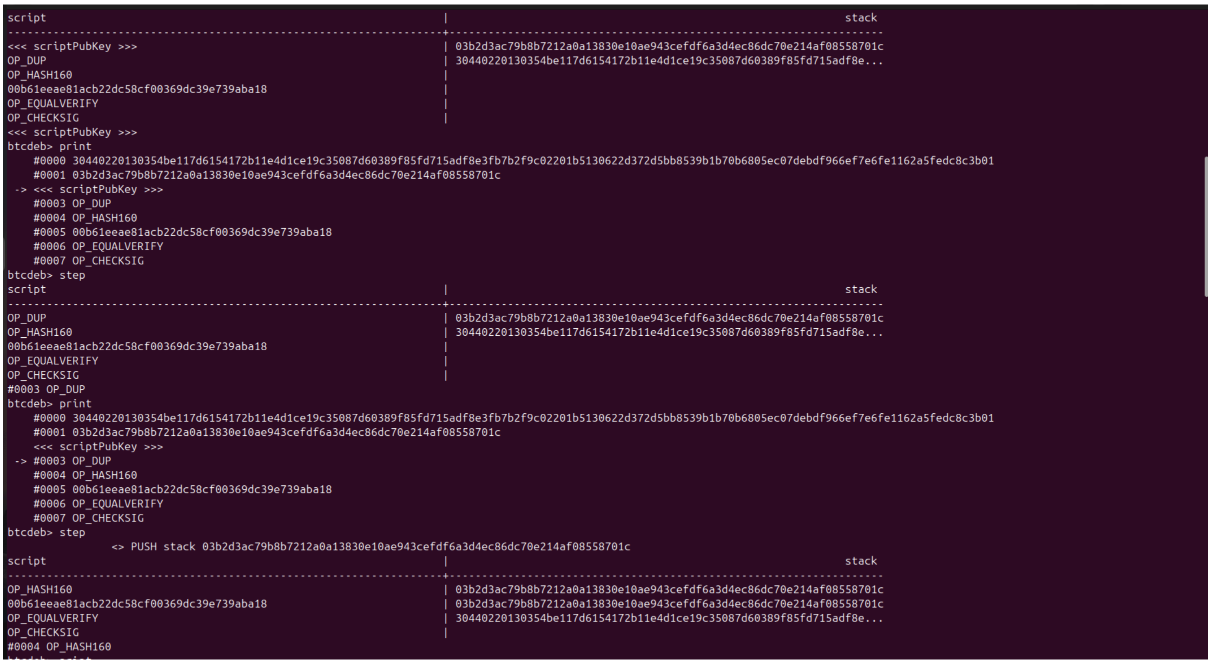
  Each transaction uses P2PKH scripts, which involve a challenge (locking script) and a response (unlocking script).

    - **Locking Script (Challenge):** Ensures only the recipient can spend the output.
    - **Unlocking Script (Response):** Provides a valid signature and public key.
    - **Validation Process:**
        - Bitcoin nodes execute the unlocking script first, pushing the provided signature and public key onto the stack.
        - Then, the locking script is executed to verify ownership by checking that the provided public key hash matches and that the signature is valid.

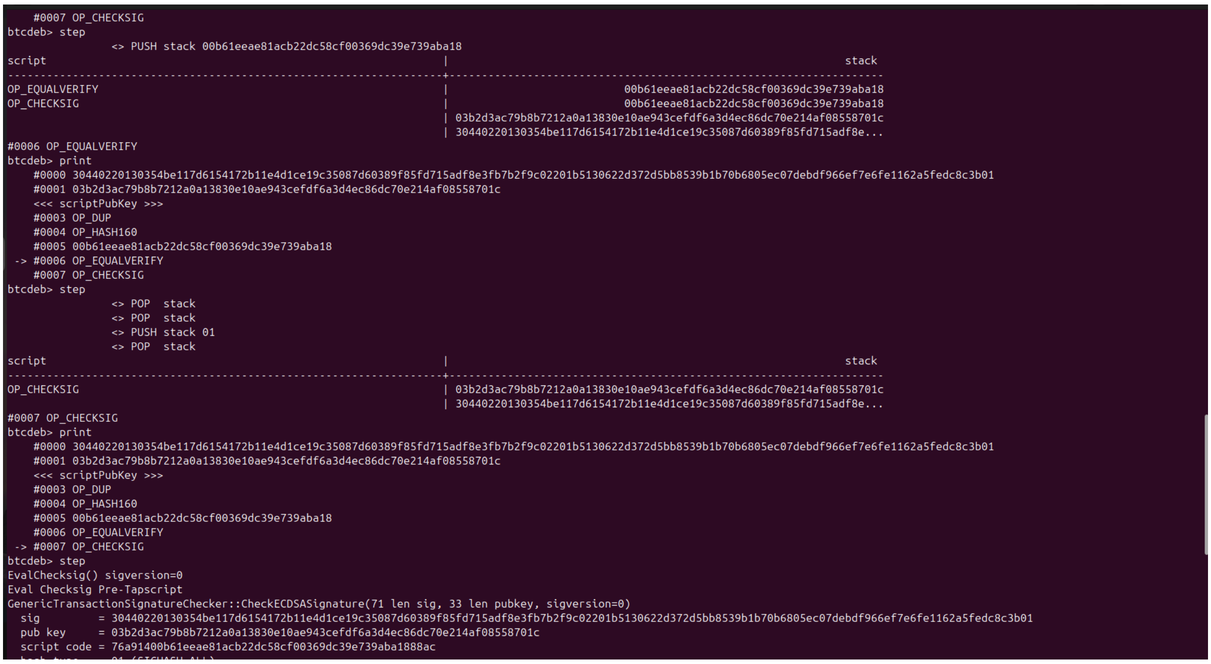
- ### Debugger Script(Legacy Transaction A->B and B->C)



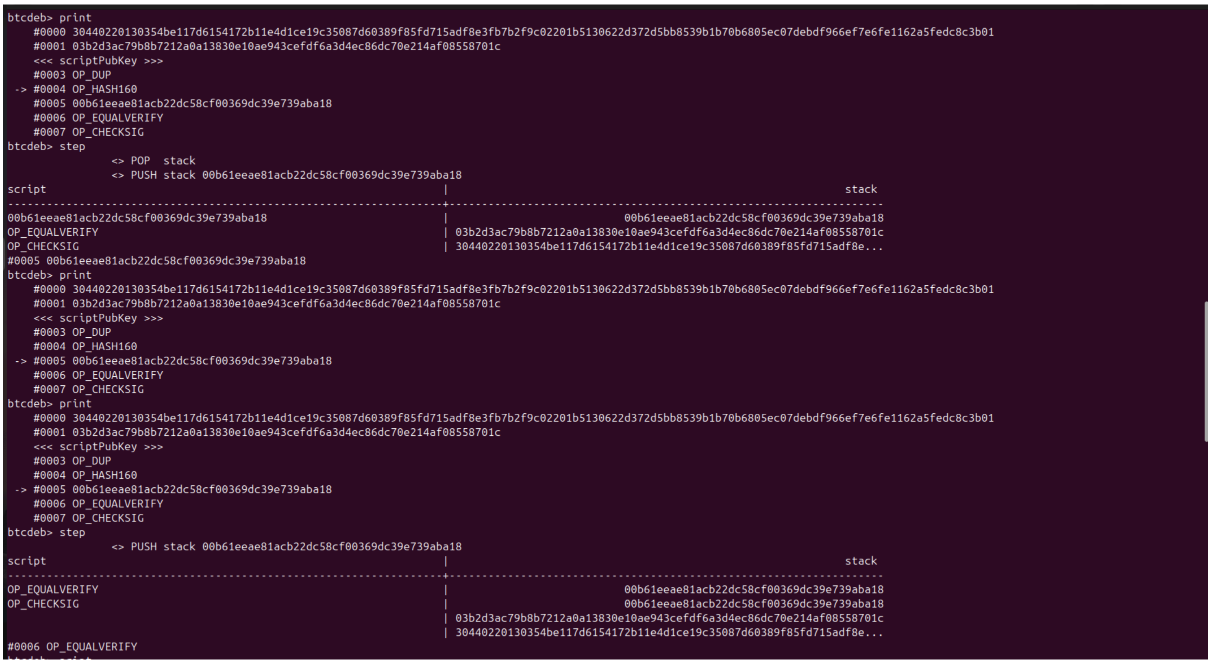
```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb --tx=0200000001045cc90aaada9fc4527e9bceabfb1bfb31bf0d0ab2fe83350155295a15e5cb11000000006a4730440220130354be117d6154172b11e4d1ce
19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b012103b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701cfdffffff01
c0bc973b000000001976a914c2a35d204944a846c1b599c9ae93a9be232f41d688ac00000000 --txin=020000000122cf991c31ad748f2aa877131ed999f1fe75ededc4f1c3cbec22acc35402f498000000006a473044022064f020
a99eeea755767a693404df76a41f3db5c5f08bd5b6cc3185d35c86f3f2022050ebdde71ff864c43eb77ebc98643def6ced6ce2acc1a345bf7fb6f3c7a4b382012103cffef69e9437895d6633166569b83f13b56b0c4758896c2d27b9
e52898164a6dfdffffff016043993b000000001976a91400b61eeae81acb22dc58cf00369dc39e739aba1888ac00000000
btcdeb 5.0.24 -- type `btcdeb -h` for start up options
LOG: signing segwit taproot
notice: btcdeb has gotten quieter; use --verbose if necessary (this message is temporary)
input tx index = 0; tx input vout = 0; value = 999900000
got witness stack of size 0
8 op script loaded. type `help` for usage information
script                                                             |  stack 
-------------------------------------------------------------------+--------
30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e... | 
03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c | 
<<< scriptPubKey >>>                                               | 
OP_DUP                                                             | 
OP_HASH160                                                         | 
00b61eeae81acb22dc58cf00369dc39e739aba18                           | 
OP_EQUALVERIFY                                                     | 
OP_CHECKSIG                                                        | 
#0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
btcdeb> step
		<> PUSH stack 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
script                                                             |                                                             stack 
-------------------------------------------------------------------+-------------------------------------------------------------------
03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
<<< scriptPubKey >>>                                               | 
OP_DUP                                                             | 
OP_HASH160                                                         | 
00b61eeae81acb22dc58cf00369dc39e739aba18                           | 
OP_EQUALVERIFY                                                     | 
OP_CHECKSIG                                                        | 
#0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
 -> #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
btcdeb> step
		<> PUSH stack 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
```



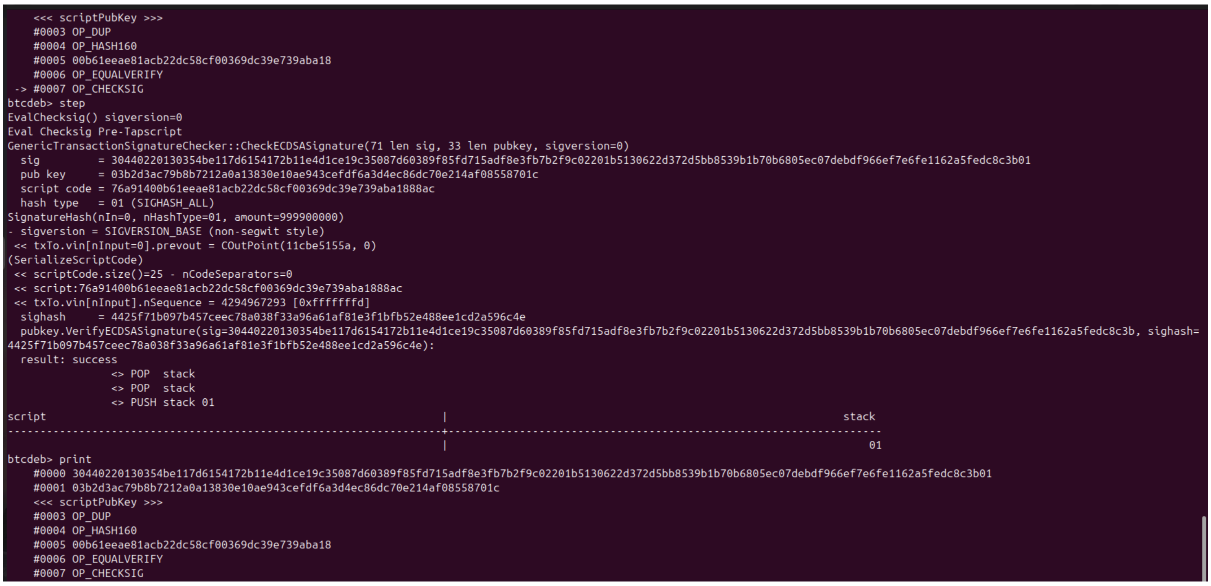
```
script                                                             |                                                             stack 
-------------------------------------------------------------------+-------------------------------------------------------------------
<<< scriptPubKey >>>                                               | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
OP_DUP                                                             | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
OP_HASH160                                                         | 
00b61eeae81acb22dc58cf00369dc39e739aba18                           | 
OP_EQUALVERIFY                                                     | 
OP_CHECKSIG                                                        | 
<<< scriptPubKey >>>
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
 -> <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
btcdeb> step
script                                                             |                                                             stack 
-------------------------------------------------------------------+-------------------------------------------------------------------
OP_DUP                                                             | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
OP_HASH160                                                         | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
00b61eeae81acb22dc58cf00369dc39e739aba18                           | 
OP_EQUALVERIFY                                                     | 
OP_CHECKSIG                                                        | 
#0003 OP_DUP
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
 -> #0003 OP_DUP
    #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
btcdeb> step
		<> PUSH stack 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
script                                                             |                                                             stack 
-------------------------------------------------------------------+-------------------------------------------------------------------
OP_HASH160                                                         | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
00b61eeae81acb22dc58cf00369dc39e739aba18                           | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
OP_EQUALVERIFY                                                     | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
OP_CHECKSIG                                                        | 
#0004 OP_HASH160
```



```
    #0007 OP_CHECKSIG
btcdeb> step
                <> PUSH stack 00b61eeae81acb22dc58cf00369dc39e739aba18
script                                                             |                                                             stack
-------------------------------------------------------------------+------------------------------------------------------------------
OP_EQUALVERIFY                                                     |                           00b61eeae81acb22dc58cf00369dc39e739aba18
OP_CHECKSIG                                                        |                           00b61eeae81acb22dc58cf00369dc39e739aba18
                                                                   | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
                                                                   | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
#0006 OP_EQUALVERIFY
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
 -> #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
btcdeb> step
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
                <> POP  stack
script                                                             |                                                             stack
-------------------------------------------------------------------+------------------------------------------------------------------
OP_CHECKSIG                                                        | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
                                                                   | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
#0007 OP_CHECKSIG
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
 -> #0007 OP_CHECKSIG
btcdeb> step
EvalChecksig() sigversion=0
Eval Checksig Pre-Tapscript
GenericTransactionSignatureChecker::CheckECDSASignature(71 len sig, 33 len pubkey, sigversion=0)
  sig         = 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
  pub key     = 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
  script code = 76a91400b61eeae81acb22dc58cf00369dc39e739aba1888ac
```



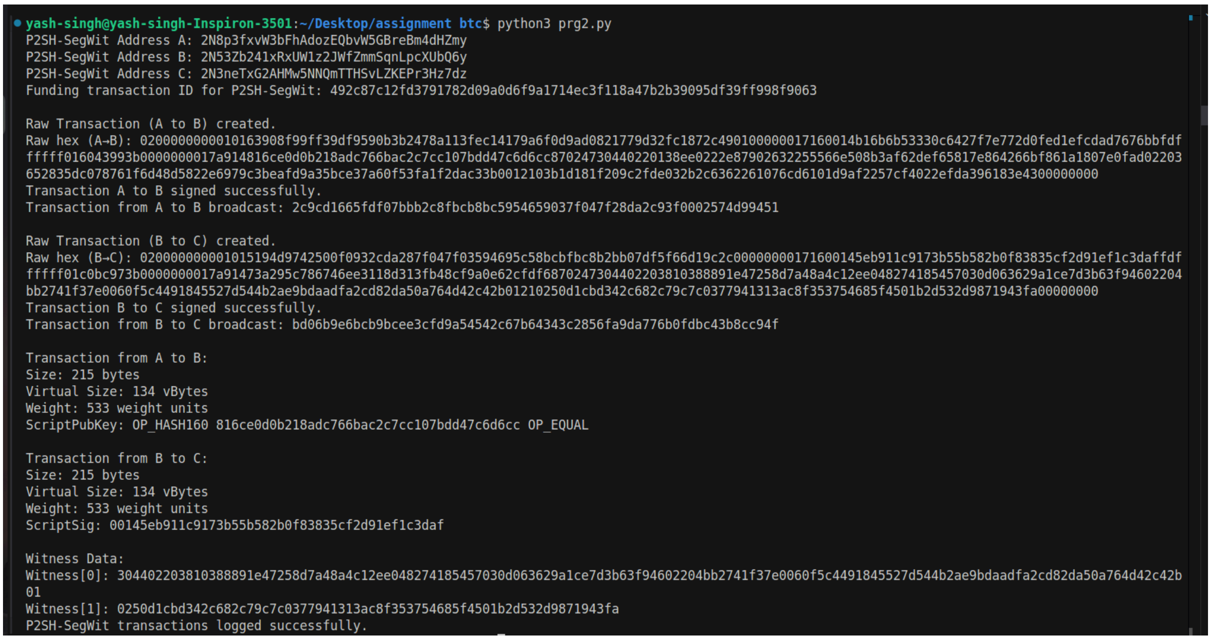
```
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
    #0003 OP_DUP
 -> #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
btcdeb> step
                <> POP  stack
                <> PUSH stack 00b61eeae81acb22dc58cf00369dc39e739aba18
script                                                             |                                                             stack
-------------------------------------------------------------------+------------------------------------------------------------------
00b61eeae81acb22dc58cf00369dc39e739aba18                           |                           00b61eeae81acb22dc58cf00369dc39e739aba18
OP_EQUALVERIFY                                                     | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
OP_CHECKSIG                                                        | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
#0005 00b61eeae81acb22dc58cf00369dc39e739aba18
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
 -> #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
 -> #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
btcdeb> step
                <> PUSH stack 00b61eeae81acb22dc58cf00369dc39e739aba18
script                                                             |                                                             stack
-------------------------------------------------------------------+------------------------------------------------------------------
OP_EQUALVERIFY                                                     |                           00b61eeae81acb22dc58cf00369dc39e739aba18
OP_CHECKSIG                                                        |                           00b61eeae81acb22dc58cf00369dc39e739aba18
                                                                   | 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
                                                                   | 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e...
#0006 OP_EQUALVERIFY
```



```
    <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
 -> #0007 OP_CHECKSIG
btcdeb> step
EvalChecksig() sigversion=0
Eval Checksig Pre-Tapscript
GenericTransactionSignatureChecker::CheckECDSASignature(71 len sig, 33 len pubkey, sigversion=0)
  sig         = 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
  pub key     = 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
  script code = 76a91400b61eeae81acb22dc58cf00369dc39e739aba1888ac
  hash type   = 01 (SIGHASH_ALL)
SignatureHash(nIn=0, nHashType=01, amount=999900000)
- sigversion = SIGVERSION_BASE (non-segwit style)
 << txTo.vin[nInput=0].prevout = COutPoint(11cbe5155a, 0)
(SerializeScriptCode)
 << scriptCode.size()=25 - nCodeSeparators=0
 << script:76a91400b61eeae81acb22dc58cf00369dc39e739aba1888ac
 << txTo.vin[nInput].nSequence = 4294967293 [0xfffffffd]
  sighash     = 4425f71b097b457ceec78a038f33a96a61af81e3f1bfb52e488ee1cd2a596c4e
  pubkey.VerifyECDSASignature(sig=30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b, sighash=
4425f71b097b457ceec78a038f33a96a61af81e3f1bfb52e488ee1cd2a596c4e):
  result: success
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
script                                                              |                                                            stack
--------------------------------------------------------------------+------------------------------------------------------------------
                                                                    |                                                                01
btcdeb> print
    #0000 30440220130354be117d6154172b11e4d1ce19c35087d60389f85fd715adf8e3fb7b2f9c02201b5130622d372d5bb8539b1b70b6805ec07debdf966ef7e6fe1162a5fedc8c3b01
    #0001 03b2d3ac79b8b7212a0a13830e10ae943cefdf6a3d4ec86dc70e214af08558701c
    <<< scriptPubKey >>>
    #0003 OP_DUP
    #0004 OP_HASH160
    #0005 00b61eeae81acb22dc58cf00369dc39e739aba18
    #0006 OP_EQUALVERIFY
    #0007 OP_CHECKSIG
```

SegWit Address Transactions (P2SH-P2WPKH)

- **Workflow**
  - **Wallet Setup**: Created a new wallet and generated three SegWit addresses (A', B', C').
  - **Funding Address A'**: Sent Bitcoin to A' using `sendtoaddress`.
  - **Transaction A' → B'**:
    - Created a raw transaction and extracted the `ScriptPubKey`.
    - Signed and broadcasted the transaction.
  - **Transaction B' → C'**:
    - Used `listunspent` to obtain the txid from A' to B'.
    - Created and broadcasted a transaction from B' to C'.



```
yash-singh@yash-singh-Inspiron-3501:~/Desktop/assignment btc$ python3 prg2.py
P2SH-SegWit Address A: 2N8p3fxvW3bFhAdozEQbvW5GBreBm4dHZmy
P2SH-SegWit Address B: 2N53Zb241xRxUW1z2JWfZmmSqnLpcXUbQ6y
P2SH-SegWit Address C: 2N3neTxG2AHMw5NNQmTTHSvLZKEPr3Hz7dz
Funding transaction ID for P2SH-SegWit: 492c87c12fd3791782d09a0d6f9a1714ec3f118a47b2b39095df39ff998f9063

Raw Transaction (A to B) created.
Raw hex (A→B): 0200000000010163908f99ff39df9590b3b2478a113fec14179a6f0d9ad0821779d32fc1872c490100000017160014b16b6b53330c6427f7e772d0fed1efcdad7676bbfdf
fffff016043993b0000000017a914816ce0d0b218adc766bac2c7cc107bdd47c6d6cc87024730440220138ee0222e87902632255566e508b3af62def65817e864266bf861a1807e0fad02203
652835dc078761f6d48d5822e6979c3beafd9a35bce37a60f53fa1f2dac33b0012103b1d181f209c2fde032b2c6362261076cd6101d9af2257cf4022efda396183e4300000000
Transaction A to B signed successfully.
Transaction from A to B broadcast: 2c9cd1665fdf07bbb2c8fbcb8bc5954659037f047f28da2c93f0002574d99451

Raw Transaction (B to C) created.
Raw hex (B→C): 020000000001015194d9742500f0932cda287f047f03594695c58bcbfbc8b2bb07df5f66d19c2c00000000171600145eb911c9173b55b582b0f83835cf2d91ef1c3daffdf
fffff01c0bc973b0000000017a91473a295c786746ee3118d313fb48cf9a0e62cfdf6870247304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7d3b63f94602204
bb2741f37e0060f5c4491845527d544b2ae9bdaadfa2cd82da50a764d42c42b01210250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa00000000
Transaction B to C signed successfully.
Transaction from B to C broadcast: bd06b9e6bcb9bcee3cfd9a54542c67b64343c2856fa9da776b0fdbc43b8cc94f

Transaction from A to B:
Size: 215 bytes
Virtual Size: 134 vBytes
Weight: 533 weight units
ScriptPubKey: OP_HASH160 816ce0d0b218adc766bac2c7cc107bdd47c6d6cc OP_EQUAL

Transaction from B to C:
Size: 215 bytes
Virtual Size: 134 vBytes
Weight: 533 weight units
ScriptSig: 00145eb911c9173b55b582b0f83835cf2d91ef1c3daf

Witness Data:
Witness[0]: 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7d3b63f94602204bb2741f37e0060f5c4491845527d544b2ae9bdaadfa2cd82da50a764d42c42b
01
Witness[1]: 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
P2SH-SegWit transactions logged successfully.
```

Analysis of Challenge and Response Scripts:

Bitcoin Script Execution

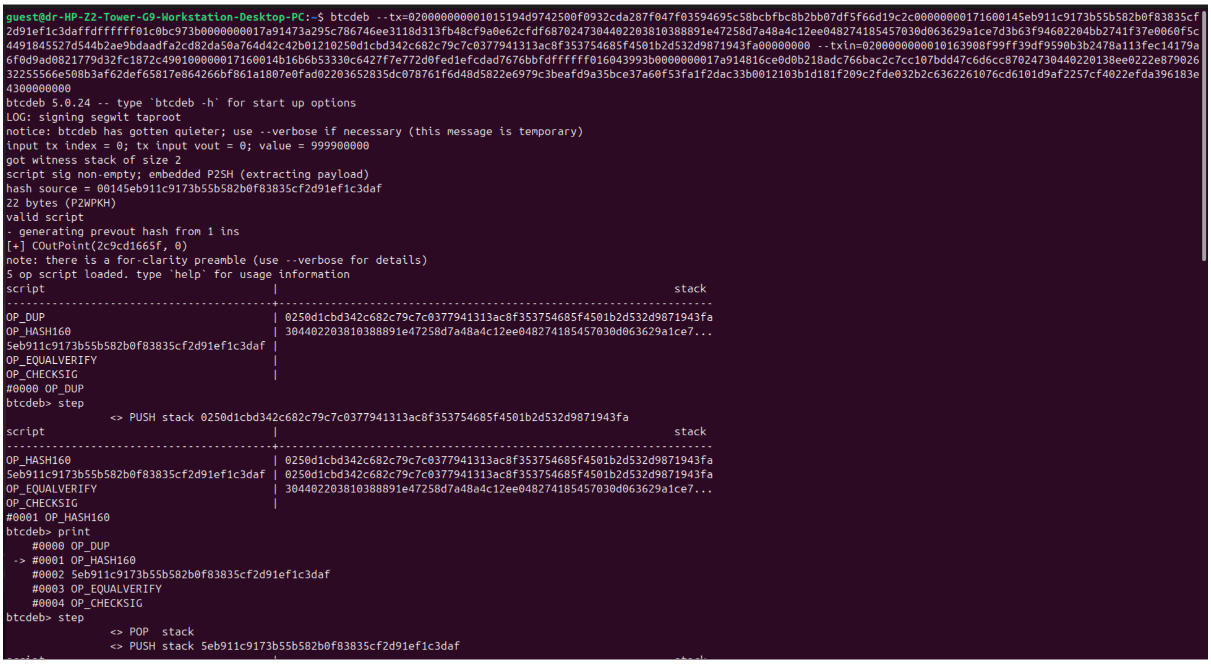
Bitcoin transactions use a challenge-response mechanism through scripting.

1. **Locking Script (scriptPubKey):** This script defines the conditions required to unlock the UTXO. A typical Pay-to-Public-Key-Hash (P2PKH) script follows the structure:
   - OP_DUP: Duplicates the public key on the stack.
   - OP_HASH160: Computes the hash of the public key.
   - OP_EQUALVERIFY: Verifies that the provided public key hash matches the expected hash.
   - OP_CHECKSIG: Validates the signature against the public key.
2. **Unlocking Script (scriptSig):** This script provides the necessary proof to meet the locking script conditions:
   - The digital signature is generated using the sender's private key and proves ownership of the UTXO.

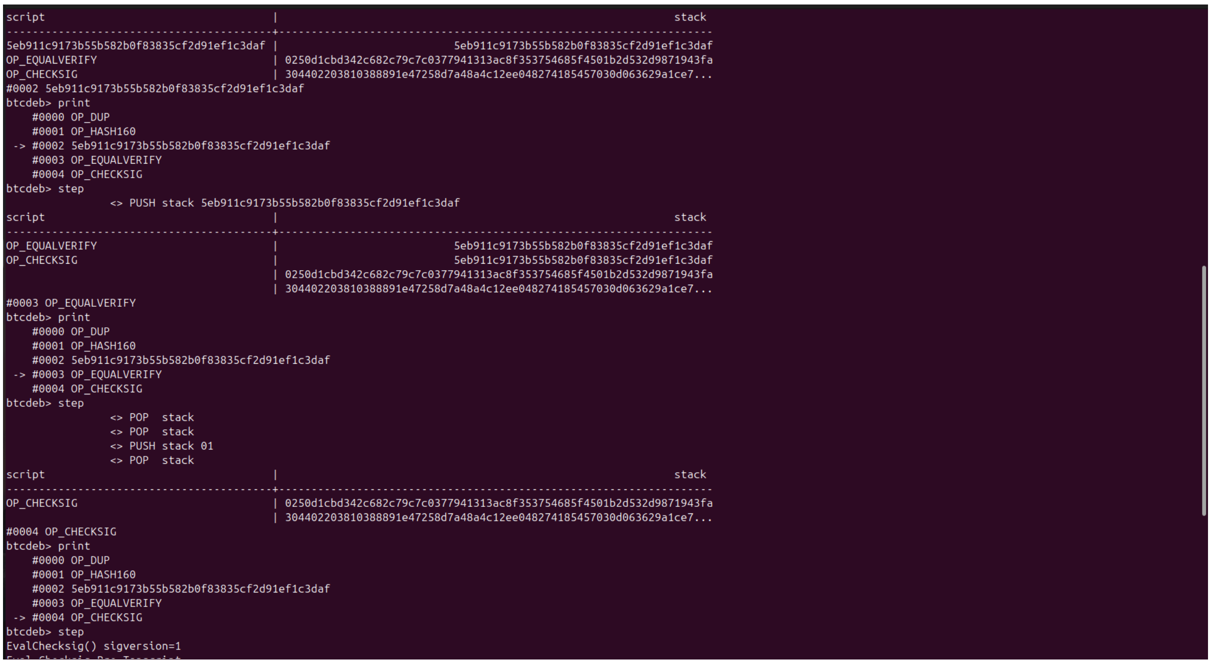
Transaction Validation Process:

3. The unlocking script (scriptSig) executes first, placing the signature and public key onto the stack.
4. The locking script (scriptPubKey) runs, verifying the public key’s hash and the authenticity of the digital signature.
5. If all conditions hold, the transaction is deemed valid and accepted into the blockchain.

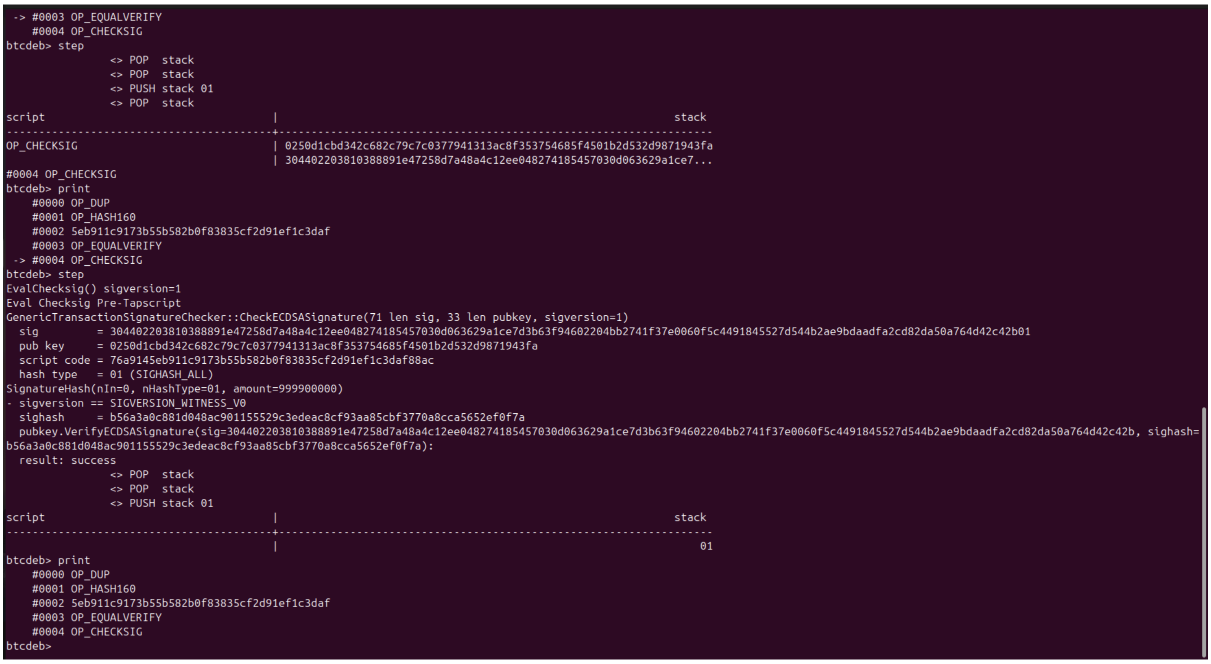
- **Debugger Script(Legacy Transaction A->B and B->C)**



```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb --tx=020000000001015194d9742500f0932cda287f047f03594695c58bcbfbc8b2bb07df5f66d19c2c00000000171600145eb911c9173b55b582b0f83835cf
2d91ef1c3daffdffffff01c0bc973b0000000017a91473a295c786746ee3118d313fb48cf9a0e62cfdf6870247304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7d3b63f94602204bb2741f37e0060f5c
4491845527d544b2ae9bdaadfa2cd82da50a764d42c42b01210250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa00000000 --txin=0200000000010163908f99ff39df9590b3b2478a113fec14179a
6f0d9ad0821779d32fc1872c490100000017160014b16b6b53330c6427f7e772d0fed1efcdad7676bbfdffffff016043993b0000000017a914816ce0d0b218adc766bac2c7cc107bdd47c6d6cc87024730440220138ee0222e879026
32255566e508b3af62def65817e864266bf861a1807e0fad02203652835dc078761f6d48d5822e6979c3beafd9a35bce37a60f53fa1f2dac33b0012103b1d181f209c2fde032b2c6362261076cd6101d9af2257cf4022efda396183e
4300000000
btcdeb 5.0.24 -- type `btcdeb -h` for start up options
LOG: signing segwit taproot
notice: btcdeb has gotten quieter; use --verbose if necessary (this message is temporary)
input tx index = 0; tx input vout = 0; value = 999900000
got witness stack of size 2
script sig non-empty; embedded P2SH (extracting payload)
hash source = 00145eb911c9173b55b582b0f83835cf2d91ef1c3daf
22 bytes (P2WPKH)
valid script
- generating prevout hash from 1 ins
[+] COutPoint(2c9cd1665f, 0)
note: there is a for-clarity preamble (use --verbose for details)
5 op script loaded. type `help` for usage information
script                                   |                                                             stack 
-----------------------------------------+-------------------------------------------------------------------
OP_DUP                                   | 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
OP_HASH160                               | 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7...
5eb911c9173b55b582b0f83835cf2d91ef1c3daf |
OP_EQUALVERIFY                           |
OP_CHECKSIG                              |
#0000 OP_DUP
btcdeb> step
                <> PUSH stack 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
script                                   |                                                             stack 
-----------------------------------------+-------------------------------------------------------------------
OP_HASH160                               | 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
5eb911c9173b55b582b0f83835cf2d91ef1c3daf | 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
OP_EQUALVERIFY                           | 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7...
OP_CHECKSIG                              |
#0001 OP_HASH160
btcdeb> print
    #0000 OP_DUP
 -> #0001 OP_HASH160
    #0002 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
    #0003 OP_EQUALVERIFY
    #0004 OP_CHECKSIG
btcdeb> step
                <> POP  stack
                <> PUSH stack 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
```



```
script                                   |                                                             stack 
-----------------------------------------+-------------------------------------------------------------------
5eb911c9173b55b582b0f83835cf2d91ef1c3daf |                           5eb911c9173b55b582b0f83835cf2d91ef1c3daf
OP_EQUALVERIFY                           | 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
OP_CHECKSIG                              | 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7...
#0002 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
btcdeb> print
    #0000 OP_DUP
    #0001 OP_HASH160
 -> #0002 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
    #0003 OP_EQUALVERIFY
    #0004 OP_CHECKSIG
btcdeb> step
                <> PUSH stack 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
script                                   |                                                             stack 
-----------------------------------------+-------------------------------------------------------------------
OP_EQUALVERIFY                           |                           5eb911c9173b55b582b0f83835cf2d91ef1c3daf
OP_CHECKSIG                              |                           5eb911c9173b55b582b0f83835cf2d91ef1c3daf
                                         | 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
                                         | 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7...
#0003 OP_EQUALVERIFY
btcdeb> print
    #0000 OP_DUP
    #0001 OP_HASH160
    #0002 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
 -> #0003 OP_EQUALVERIFY
    #0004 OP_CHECKSIG
btcdeb> step
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
                <> POP  stack
script                                   |                                                             stack 
-----------------------------------------+-------------------------------------------------------------------
OP_CHECKSIG                              | 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
                                         | 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7...
#0004 OP_CHECKSIG
btcdeb> print
    #0000 OP_DUP
    #0001 OP_HASH160
    #0002 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
    #0003 OP_EQUALVERIFY
 -> #0004 OP_CHECKSIG
btcdeb> step
EvalChecksig() sigversion=1
```



```
 -> #0003 OP_EQUALVERIFY
    #0004 OP_CHECKSIG
btcdeb> step
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
                <> POP  stack
script                                   |                                                             stack 
-----------------------------------------+-------------------------------------------------------------------
OP_CHECKSIG                              | 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
                                         | 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7...
#0004 OP_CHECKSIG
btcdeb> print
    #0000 OP_DUP
    #0001 OP_HASH160
    #0002 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
    #0003 OP_EQUALVERIFY
 -> #0004 OP_CHECKSIG
btcdeb> step
EvalChecksig() sigversion=1
Eval Checksig Pre-Tapscript
GenericTransactionSignatureChecker::CheckECDSASignature(71 len sig, 33 len pubkey, sigversion=1)
  sig         = 304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7d3b63f94602204bb2741f37e0060f5c4491845527d544b2ae9bdaadfa2cd82da50a764d42c42b01
  pub key     = 0250d1cbd342c682c79c7c0377941313ac8f353754685f4501b2d532d9871943fa
  script code = 76a9145eb911c9173b55b582b0f83835cf2d91ef1c3daf88ac
  hash type   = 01 (SIGHASH_ALL)
SignatureHash(nIn=0, nHashType=01, amount=999900000)
- sigversion == SIGVERSION_WITNESS_V0
  sighash     = b56a3a0c881d048ac901155529c3edeac8cf93aa85cbf3770a8cca5652ef0f7a
  pubkey.VerifyECDSASignature(sig=304402203810388891e47258d7a48a4c12ee048274185457030d063629a1ce7d3b63f94602204bb2741f37e0060f5c4491845527d544b2ae9bdaadfa2cd82da50a764d42c42b, sighash=
b56a3a0c881d048ac901155529c3edeac8cf93aa85cbf3770a8cca5652ef0f7a):
  result: success
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
script                                   |                                                             stack 
-----------------------------------------+-------------------------------------------------------------------
                                         |                                                                 01
btcdeb> print
    #0000 OP_DUP
    #0001 OP_HASH160
    #0002 5eb911c9173b55b582b0f83835cf2d91ef1c3daf
    #0003 OP_EQUALVERIFY
    #0004 OP_CHECKSIG
btcdeb>
```

Analysis and Comparison

Comparison of P2PKH (Legacy) and P2SH-P2WPKH (SegWit) Transactions

Size Comparison

The transaction data shows significant size differences between the two transaction types:

- **P2PKH (Legacy) Transactions**:
    - Virtual Size: 191 vBytes
    - Weight: 764 weight units
- **P2SH-P2WPKH (SegWit) Transactions**:
    - Virtual Size: 134 vBytes
    - Weight: 533 weight units

This represents a reduction of 57 vBytes in virtual size (30% smaller) and 231 weight units (30% smaller) when using P2SH-P2WPKH compared to legacy P2PKH transactions.

Script Structure Comparison

P2PKH (Legacy) Structure

- **ScriptPubKey (Challenge)**: `OP_DUP OP_HASH160 OP_EQUALVERIFY OP_CHECKSIG`
- **ScriptSig (Response)**: (Contains the signature and public key)

The entire signature and public key data must be included in the transaction itself, contributing to the larger size.

P2SH-P2WPKH (SegWit) Structure

- **ScriptPubKey (Challenge)**: `OP_HASH160 OP_EQUAL`
- **ScriptSig (Response)**: (Typically `0014{20-byte-key-hash}`)
- **Witness Data** (separate from ScriptSig):
  - Witness: (Signature)
  - Witness: (Public Key)

The key difference is that signature and public key data are moved to the witness data structure, which is counted differently for fee calculations.

Benefits of SegWit Transactions

1. **Reduced Transaction Size**: SegWit transactions are approximately 30% smaller in virtual size, resulting in lower transaction fees.

2. **Transaction Malleability Fix**: By moving signature data (witness data) outside the transaction hash calculation, SegWit solves the transaction malleability problem that previously affected Bitcoin.

3. **Increased Block Capacity**: While maintaining the 1MB block size limit for backward compatibility, SegWit effectively increases the block capacity by giving witness data a "discount" in the weight calculation.

4. **Scalability Improvements**: The smaller transaction size allows more transactions to fit in each block, effectively increasing Bitcoin's throughput without changing the base block size.

5. **Fee Efficiency**: The reduced virtual size directly translates to lower fees for the same transaction, making Bitcoin more economical to use, especially during periods of network congestion.

The primary reason SegWit transactions are smaller is the separation of witness data (signatures) from the transaction data used to calculate the transaction ID, and the different accounting method used for this witness data in the block weight calculation.