Linearize

Construct a linear, no-fork, best version of the Bitcoin blockchain.

Step 1: Download hash list

```
$ ./linearize-hashes.py linearize.cfg > hashlist.txt
```

Required configuration file settings for linearize-hashes: * RPC: `datadir` (Required if `rpcuser` and `rpcpassword` are not specified) * RPC: `rpcuser`, `rpcpassword` (Required if `datadir` is not specified)

Optional config file setting for linearize-hashes: * RPC: `host` (Default: `127.0.0.1`) * RPC: `port` (Default: `8332`) * Blockchain: `min_height`, `max_height` * `rev_hash_bytes`: If true, the written block hash list will be byte-reversed. (In other words, the hash returned by getblockhash will have its bytes reversed.) False by default. Intended for generation of standalone hash lists but safe to use with linearize-data.py, which will output the same data no matter which byte format is chosen.

The `linearize-hashes` script requires a connection, local or remote, to a JSON-RPC server. Running `bitcoind` or `bitcoin-qt -server` will be sufficient.

Step 2: Copy local block data

```
$ ./linearize-data.py linearize.cfg
```

Required configuration file settings: * `output_file`: The file that will contain the final blockchain. or * `output`: Output directory for linearized `blocks/blkNNNNN.dat` output.

Optional config file setting for linearize-data: * `debug_output`: Some printouts may not always be desired. If true, such output will be printed. * `file_timestamp`: Set each file's last-accessed and last-modified times, respectively, to the current time and to the timestamp of the most recent block written to the script's blockchain. * `genesis`: The hash of the genesis block in the blockchain. * `input`: bitcoind blocks/ directory containing blkNNNNN.dat * `hashlist`: text file containing list of block hashes created by linearize-hashes.py. * `max_out_sz`: Maximum size for files created by the `output_file` option. (Default: `1000*1000*1000 bytes`) * `netmagic`: Network magic number. * `out_of_order_cache_sz`: If out-of-order blocks are being read, the block can be written to a cache so that the blockchain doesn't have to be sought again. This option specifies the cache size. (Default: `100*1000*1000 bytes`) * `rev_hash_bytes`: If true, the block hash list written by linearize-hashes.py will be byte-reversed when read by linearize-data.py. See the linearize-hashes entry for more information. * `split_timestamp`: Split blockchain files when a new month is first seen, in addition to reaching a maximum file size (`max_out_sz`).