

Utreexo cmd to only validate the current chain tip 현재 chain tip을 검증하는 Utreexo 커맨드 라인 툴 만들기

Description:

A command line tool that will download blocks and utreexo proofs from peers to output a validated utreexo state at a block close to the tip.

The problem it solves:

Currently, a node does 2 things:

- 1: Perform the initial block download to catch the node's chainstate to the tip.
- 2: Receive, validate and propagate transactions/blocks once caught up.

The goals of 1 and 2 are different and for the best optimization, it's better to treat the two as two separate modes. This is the case today and the initial block download code is different from when the node is caught up.

However, currently it's not as easy to separate the two because of the large chainstate. But by using the utreexo accumulators, it's possible to keep the chainstate smaller than a kilobyte. This makes it possible to easily transfer the chainstate from one command line tool to the other.

Expected results:

Make a binary under the directory cmd/ in utreexod which validates and outputs a utreexo state both to the standard output and a file upon completing validation.

Bonus goal: embed utreexo roots into the binary and validate the blocks in parallel.

설명:

다른 노드로부터 블록 및 Utreexo proof를 다운로드하여 정해진 블록의 Utreexo state를 검증하여 Utreexo 를 출력하는 커맨드 라인 도구 만들기.

해결하는 문제:

현재 노드는 두 가지 역할을 한다:

1. 초기 블록 다운로드 (IBD)를 통해 노드의 체인 상태를 최신 상태로 동기화.
 2. 동기화가 완료된 이후 트랜잭션/블록을 수신, 검증, 그리고 전파.
- 1번과 2번의 목표는 다르며, 최적화를 위해 이 두 작업을 별도의 모드로 처리하는 것이 더 효율적이다. 현재 사용하는 IBD 코드도 이와같은 동기화된 노드의 코드와 다르게 다루고 있다.

하지만 **chainstate**가 너무 크기 때문에 이 두 작업을 각각 다른 커맨드라인 툴로 나누기 어려웠다. 이는 **Utreexo accumulator**를 사용하여 체인 상태를 1킬로바이트보다 작게 유지하여 구현이 가능해졌다.

결과물:

Utreexod의 **cmd/** 디렉토리에 바이너리를 생성하여, 블록 검증 완료 시 **Utreexo** 상태를 표준 출력 및 파일로 출력.

추가 목표: **Utreexo** 루트를 바이너리에 포함하고 블록을 병렬로 검증할 수 있도록 구현.