

블록체인 과제 #3

이름: 조승현
학번: 201824590

이번 과제의 목표는 자동 입찰 기능 및 입찰 시간 연장 기능이 구현된 이더리움 스마트 컨트랙트의 구현입니다. 이를 위해 Solidity 에서 제공하는 함수 오버라이딩 기능을 활용해 아래 그림과 같이 초기 입찰값(_initBid)과 최대 입찰값(_maxBid)을 함수 인자로 입력 받는 bid() 함수를 추가로 정의했습니다. 또한, 각각의 bid() 함수 끝에 autoBidding() 함수를 **internal** 로 구현하였고, auto bidder 로 등록된 최대 10개의 EOA 계정들에 대해 자동 입찰 기능을 수행할 수 있도록 함수를 재구성했습니다. 결과의 가독성을 위해 자동 입찰 금액은 최대 입찰 금액보다 1 wei 높은 금액이 되도록 구현했습니다. auto bidder 목록은 address 배열을 선언하여 구현했으며, 부가적으로 각 EOA 계정에 대한 현재 입찰 금액(curBid), 최대 입찰 금액(maxBid)를 기록하는 mapping 을 구현했습니다. withdraw() 함수의 경우에도 auto bidder 여부를 나누어 pendingRefund 금액을 적절히 계산하였습니다. 구현에 관해서는 과제물로서 함께 첨부된 AutoBid.sol 파일을 참고할 수 있습니다.

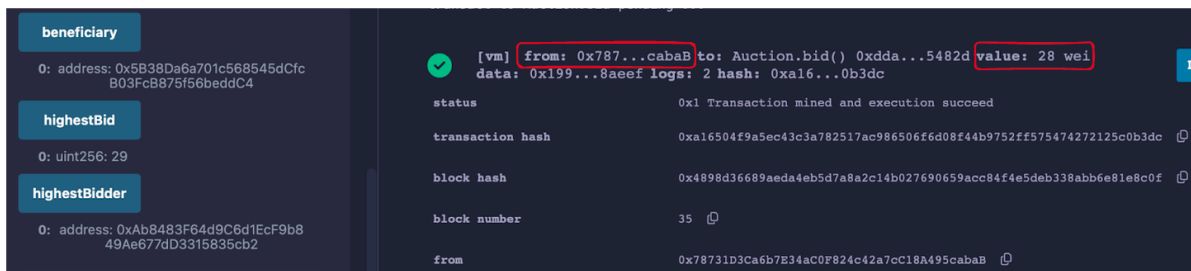


그림 1. 자동 입찰 (1)



그림 2. 자동 입찰 (2)

그림 1, 2는 하나의 Tx 로그를 나타낸 그림입니다. 우선 0xAb8...35cb2 계정이 (초기 입찰, 최대 입찰)을 (10 wei, 30 wei) 로 입찰하고 0x787...cabaB 계정이 28 wei 만큼 입찰한 결과를 보여줍니다.

그림 1의 트랜잭션 로그는 빨강 박스로 표시된 것처럼 0x787...cabaB 계정이 28 wei 만큼 bid() 함수 호출한 결과를 나타냅니다. 그림 2는 해당 로그의 내부 사항으로써, 0xAb8...35cb2 계정이 내부적으로 28 wei 보다 1 wei 높은 29 wei 를 입찰한 결과를 보여줍니다.

표 1은 그림1, 2의 기능 구현한 코드 중 일부인 autoBidding() 함수입니다. 자동 입찰은 누군가 입찰을 성공했을 때 트리거되기 때문에 각 bid() 함수의 로직 끝에 이를 추가하였습니다.

```

function autoBidding() internal {
    for (uint i = 0; i < autoBidder.length; i++) {
        if (highestBidder != autoBidder[i]) {
            if (maxBid[autoBidder[i]] >= highestBid + 1 wei) {
                pendingReturns[highestBidder] += highestBid;

                highestBidder = autoBidder[i];
                highestBid = highestBid + 1 wei;

                emit HighestBidIncreased(highestBidder, highestBid);
            }
        }
    }
}

```

표 1. autoBidding()

CALL	[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Auction.auctionEndTime() data: 0x4b4...49cba
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	Auction.auctionEndTime() 0xEf9f1ACE83dfbB8f559Da621f4aEA72C6EB10eBf
execution cost	2305 gas (Cost only applies when called by a contract)
input	0x4b4...49cba
decoded input	{}
decoded output	{ "0": "uint256: 1716573833"

그림 3. 기존 경매 종료 시간

CALL	[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Auction.auctionEndTime() data: 0x4b4...49cba
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	Auction.auctionEndTime() 0xEf9f1ACE83dfbB8f559Da621f4aEA72C6EB10eBf
execution cost	2305 gas (Cost only applies when called by a contract)
input	0x4b4...49cba
decoded input	{}
decoded output	{ "0": "uint256: 1716573863"

그림 4. 연장된 경매 종료 시간

그림 3, 4는 경매 종료 10초 전에 새로운 입찰에 성공했을 경우 경매 시간이 30초 증가하는 것을 보여줍니다. 표 2는 bid() 함수 내에 이를 구현한 코드 부분입니다.

```

function bid() public payable {
    // skip
    // (2) extending auctionEndTime
    if (auctionEndTime - now < 10) {
        // plus 30 seconds
        auctionEndTime += 30;
    }
}

```

표 2. 경매 종료 시간 연장 로직