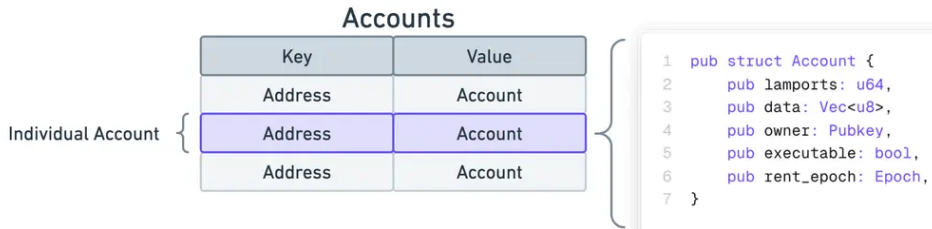


Contents

1. 강의 개요
2. 어카운트가 뭐라고요?
3. 그러면 프로그램은?
4. 솔라나 프로그램의 구조: 데이터와 컴퓨테이션
5. 어카운트 권한: Singer and Writeable
6. Program Derived Address (PDA)
7. Cross Program Invocation (CPI)
8. Labs: Escrow Program

어카운트가 뭐라고요?

- 솔라나 네트워크에 저장되어있는 모든 데이터는 "어카운트"에 있음
- 아래의 "어카운트" 테이블이 있는, 누구나 액세스 할 수 있는 데이터베이스
- 운영체제의 입장에서 "파일"과 가장 비슷한 개념



어카운트가 뭐라고요?

```
pub struct Account {  
    /// lamports in the account  
    pub lamports: u64,  
    /// data held in this account  
    pub data: Vec<u8>,  
    /// the program that owns this account. If executable, the program that  
    loads this account.  
    pub owner: Pubkey,  
    /// this account's data contains a loaded program (and is now read-only)  
    pub executable: bool,  
    /// the epoch at which this account will next owe rent  
    pub rent_epoch: Epoch,  
}
```

어카운트가 뭐라고요?

- 10MiB 까지 저장 가능 함 (both executable program code or program state)
- 크기에 따라 렌트 디파짓 (deposit in SOL) 요구
- 모든 어카운트는 지정된 프로그램 "owner" 를 가지고 있음
- 오직 명시된 프로그램만:
 - 데이터의 변경
 - SOL 잔고를 사용 할 수 있음 (debit)
- 하지만 SOL 잔고를 올리는 것은 권한이 필요 없음 (credit)

Example: An Account Containing Data (Token Account)

Account: DxGWhWxnAc4B6D3VdmoAupyRt5rHEM35ASTeydETT344

Public Key: DxGWhWxnAc4B6D3VdmoAupyRt5rHEM35ASTeydETT344

Balance: 0.00203928 SOL

Owner: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA

Executable: false

Rent Epoch: 18446744073709551615

Length: 165 (0xa5) bytes

```
0000: 1d 8c cf 87 ac 01 47 ba e7 56 eb 96 3a 2e f6 24 .....G..V...$
0010: 4c 96 91 56 9a 8e c0 8f 00 20 a2 eb 8f bd b5 a1 L..V.....
0020: 29 c1 15 54 7f 3e ab a6 79 89 fa 1c ce e5 9c 9e )..T.>..y.....
0030: e5 6e 8a 1a 37 92 7b b2 e1 20 e5 0a 4f 9a 0a 03 .n..7.{...0...
0040: 80 6e b1 9e 32 03 00 00 00 00 00 00 00 00 00 00 .n..2.....
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0060: 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 .....
0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00a0: 00 00 00 00 00 .....
```


그러면 프로그램은?

솔라나 프로그램의 구조: 데이터와 컴퓨테이션

- 솔라나 프로그램(smart contracts)은
 - 실행가능한 코드를 저장하고 있는
 - stateless 한 어카운트
 - 실행 가능한 함수들은 instruction 개념으로 정리 되어 있음
- 실행을 위해 요구되는 상태(state)는 프로그램들을 통해 생성되고 관리 됨
- 데이터 -> 아카운트
- 컴퓨테이션 -> instruction

Native (Built-in) Programs

솔라나 네트워크의 기본적인 작동을 위한 특수 프로그램들

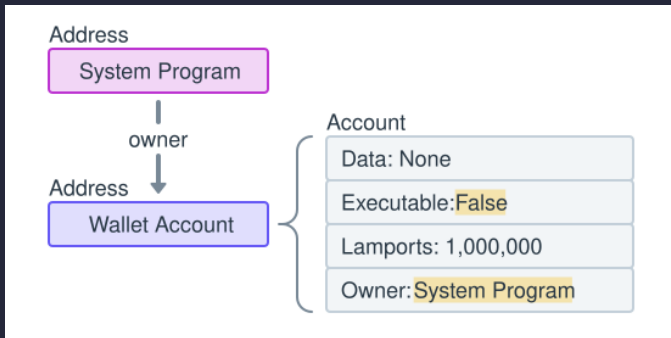
주요 프로그램으로:

- The System Program
- BPF Loader

Native (Built-in) Programs

The System Program:

- 네트워크의 모든 어카운트들의 생성
- 어카운트들의 저장 공간 할당
- 어카운트들의 owner 프로그램 할당
- 네트워크의 모든 "Wallet" 어카운트들은 간단히 말해 시스템 프로그램 어카운트:
 - "Wallet" 어카운트들의 SOL 발란스는 어카운트의 Lamports 발란스
 - 오직 시스템 프로그램 어카운트만 트랜잭션 fee payer로 사용될 수 있음



Native (Built-in) Programs

The BPF Loader:

- 네트워크에 존재하는 모든 프로그램의 owner
- 프로그램 deployment 와 upgrade 를 처리함

Example: An Account Containing Program (Token Program)

Program ID: ATokenGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL

[illegible]

어카운트 권한

Signer and Writeable

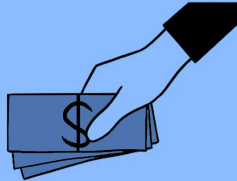
- Accounts that are writable and signers
- Accounts that are read-only and signers
- Accounts that are writable and not signers
- Accounts that are read-only and not signers

Program Derived Address (PDA)

- Deterministic Account Addresses:
 - seeds (미리 정의된 입력) 와 주소간의 1:1 관계를 정의할 수 있음
- 프로그램이 자신의 프로그램 ID에서 파생된 PDA에 대해 "서명"할 수 있도록 함
- seeds 로부터 Hashmap 과 유사한 구조를 생성할 수 있음

Cross Program Invocation (CPI)

- CPI는 프로그램 instruction 내부에서 다른 프로그램의 instruction 을 호출하는 것을 말함
- 프로그램은 자신의 프로그램 ID에서 파생된 PDA를 사용하여 "서명" 가능함
- 피호출자 프로그램은 다른 프로그램에 추가적인 CPI를 수행할 수 있으며, 최대 4단계 까지 지원



Escrow

[ˈe-,skrō]

A legal concept describing a financial agreement whereby an asset or money is held by a third party on behalf of two other parties that are in the process of completing a transaction.



Labs: Escrow Program

Account Types

```
pub struct EscrowState {  
    pub manager: Pubkey,  
    pub maker_fee_bps: u16,  
    pub taker_fee_bps: u16,  
    pub bump: u8,  
}
```


Example: Escrow Program

Account Types

```
pub struct Offer {  
    pub id: u64,  
    pub maker: Pubkey,  
    pub token_a_mint: Pubkey,  
    pub token_b_mint: Pubkey,  
    pub token_b_wanted_amount: u64,  
    pub bump: u8,  
}
```

Example: Escrow Program

Account Types

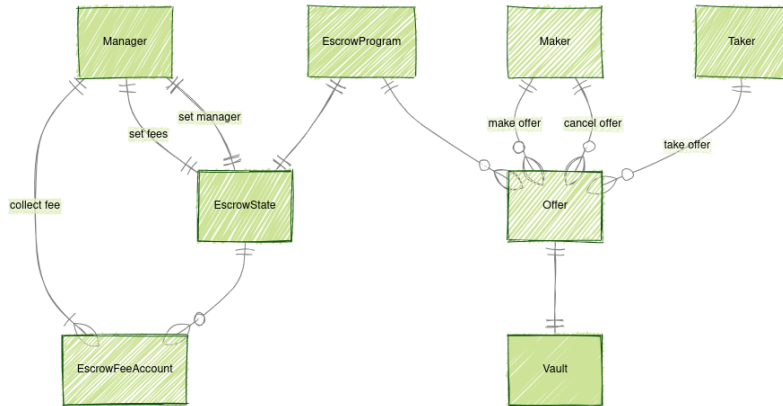
```
let offer_seeds = &[  
  b"offer"  
  maker_pubkey.as_ref(),  
  &offer_id.to_le_bytes(),  
];
```

Labs: Escrow Program

Instructions

```
pub enum EscrowInstruction {  
    // Manager-facing instructions  
    Initialize(InitializeArgs), // Initialize the program state (Escrow State)  
    SetFees(SetFeesArgs),       // Set programwide fees (maker fee and taker  
    fee) in the program state  
    SetManager,                 // Set programwide manager in the program state  
    CollectFee(CollectFeeArgs),  // Collect tokens in escrow fee account  
  
    // User-facing instructions  
    MakeOffer(MakeOfferArgs),   // Make escrow offer  
    TakeOffer,                  // Take escrow offer  
    CancelOffer,                // Cancel escrow offer  
}
```

Labs: Escrow Program



References:

- <https://solana.com/developers/guides/getstarted/intro-to-native-rust>
- <https://solana.com/docs/core/programs>
- <https://solana.com/docs/core/transactions>
- <https://solana.com/docs/core/pda#facts>
 - <https://solanacookbook.com/kr/core-concepts/pdas.html#facts>
- <https://solana.com/docs/core/cpi>
- <https://www.helius.dev/blog/a-guide-to-testing-solana-programs>