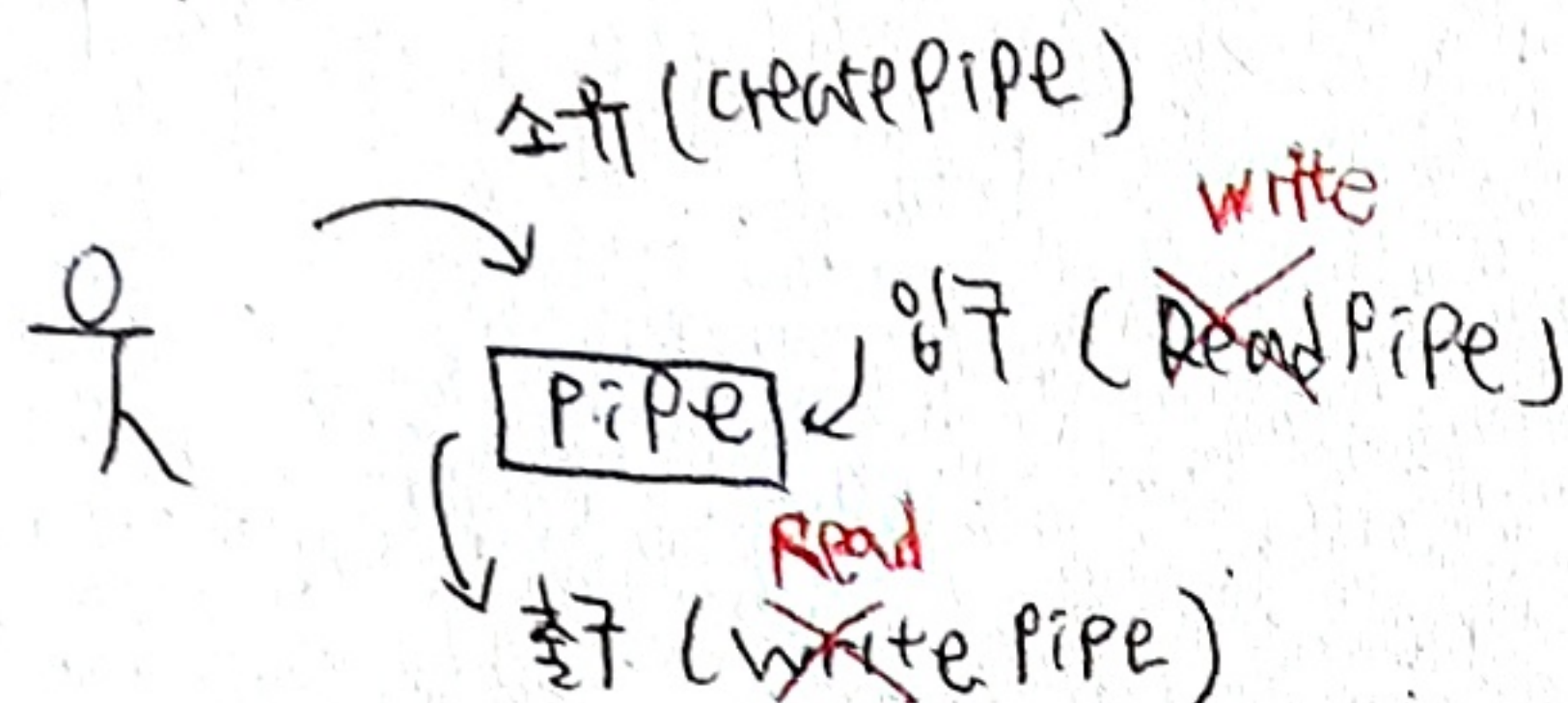


# <파이프 방식의 IPC>

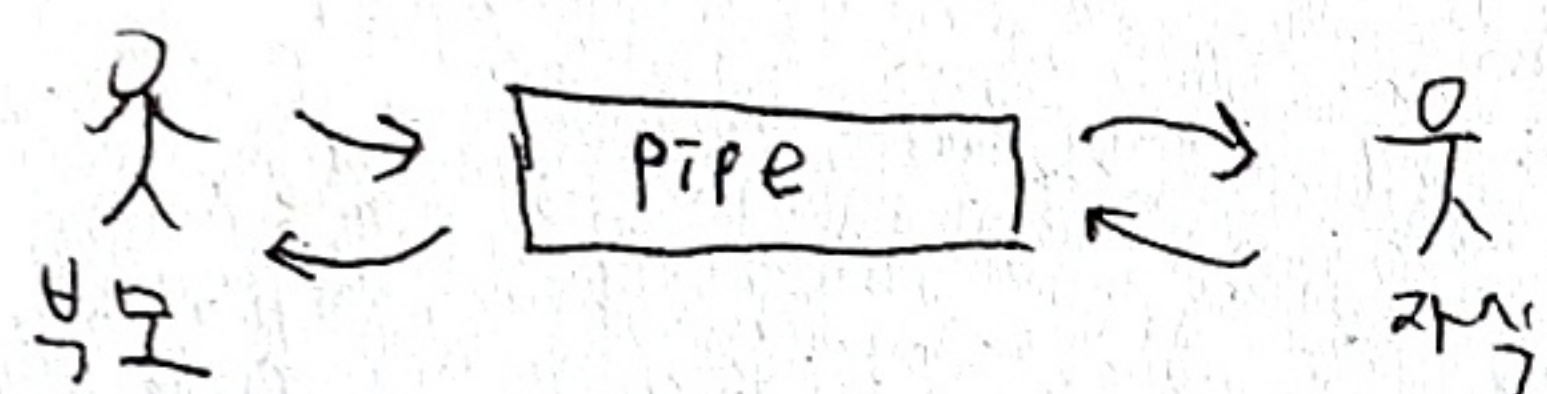
①

이름없는 파이프	이름있는 파이프
단방향, 부모자식 프로세스	양방향, 제한 X

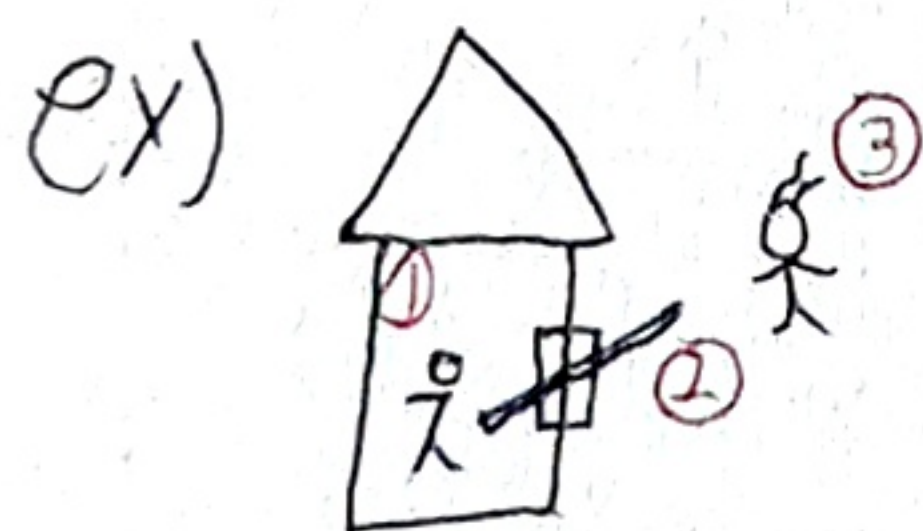
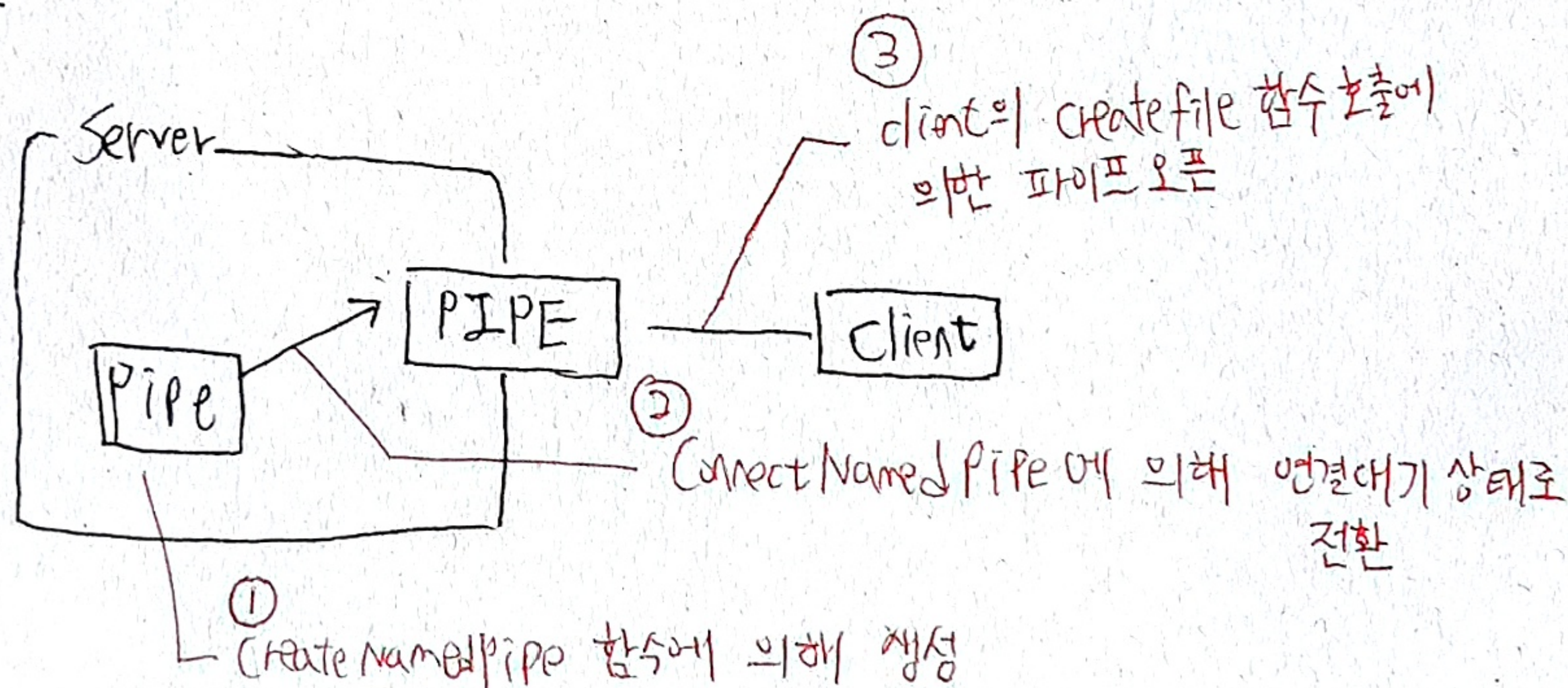
② 이름없는 파이프



핸들은 상속이 가능하므로 부모간의 통신이 가능!  
자식 Process 생성시



③ 이름 있는 파이프



## <프로세스 환경변수>

Set Environment Variable  
Get

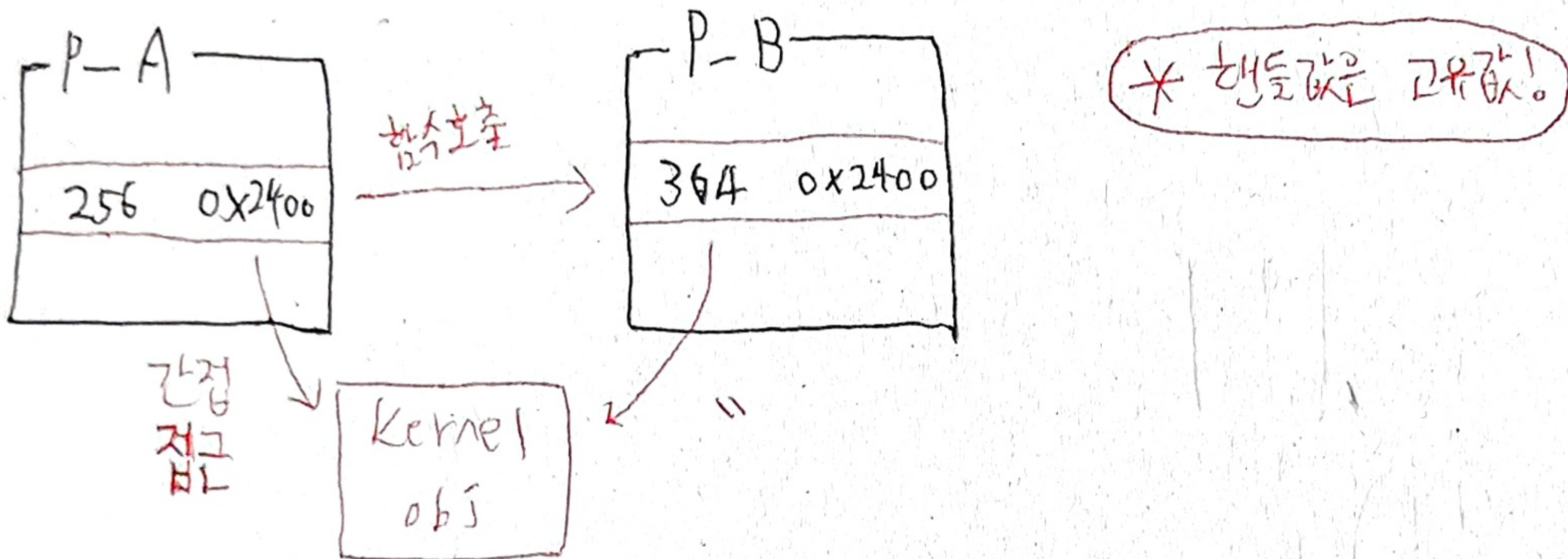
key	Value

=> 자식 Process를 보면 부모 Process에서 설정해줬던 Value 값이 출력됨 = 상속이 가능하다.



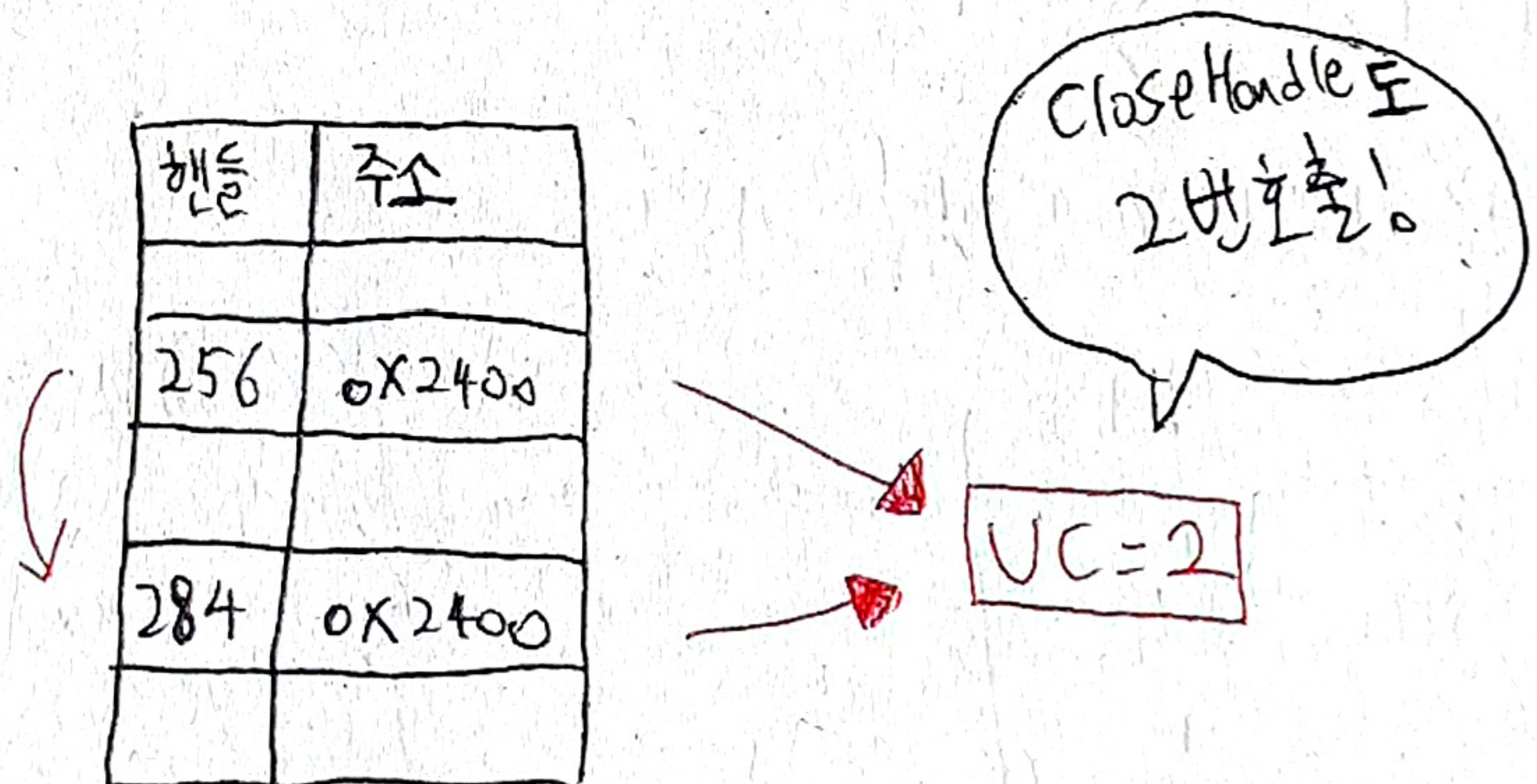
# < Pseudo 핸들과 핸들의 중복 >

Duplicate Handle ( A 핸들, 256, B 핸들, &val, -- );  
 ↳ Handle val = 364



## ① 자기 자신에게 복사

Duplicate Handle ( P-A 핸들, 256, P-A 핸들, &val );



## ② 자기 프로세스에게 부모 프로세스의 핸들 전달

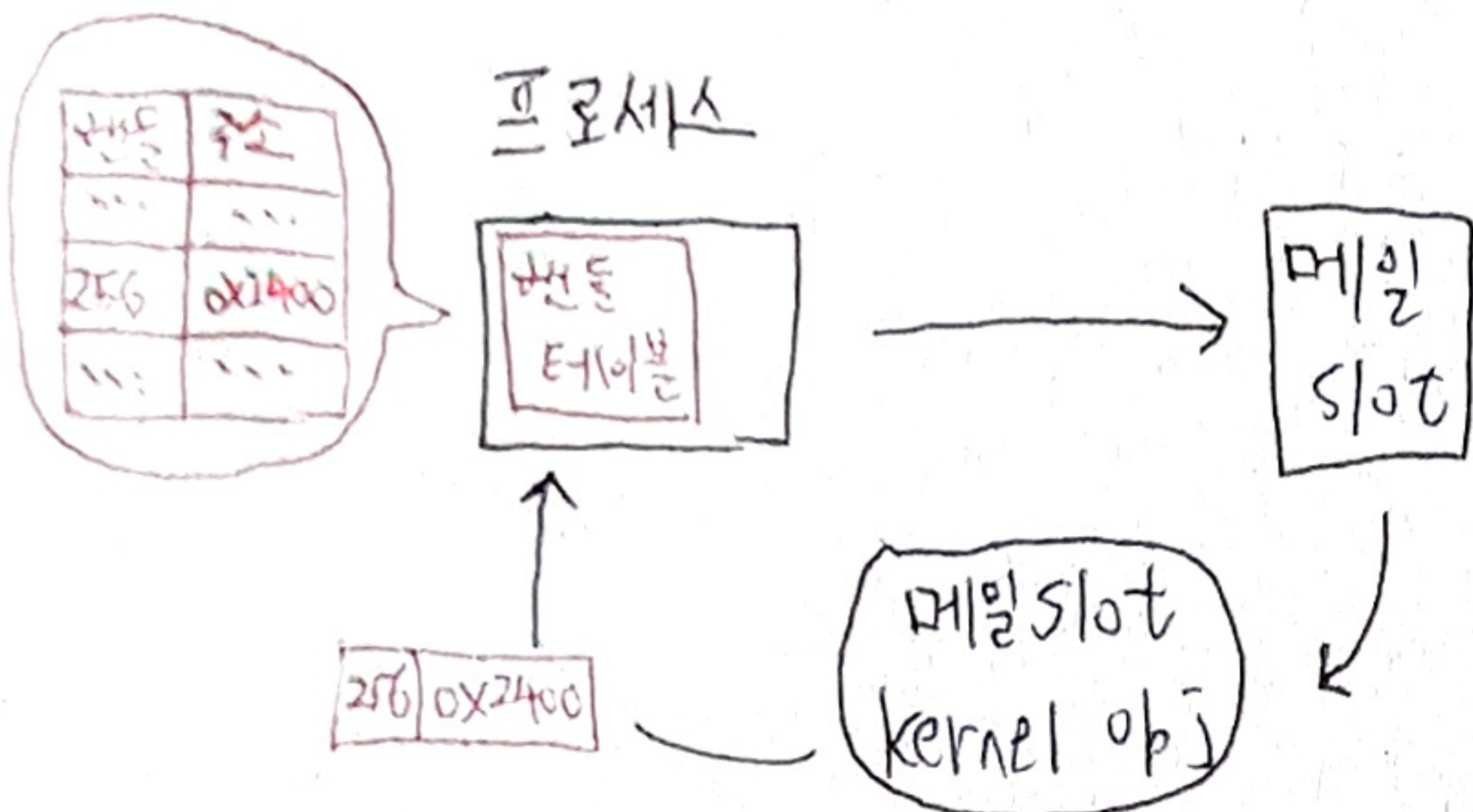
부모 프로세스가 종료하기를 기다리기 원한다면 ... => Get Current Handle ( ) 사용!



# 8장 핸들 테이블과 오브젝트 핸들의 상속

- 1/13 이충현

## < 핸들 테이블의 이해 >



## < 핸들 테이블의 상속 >

= 핸들의 상속여부가 Y 인 것만 가능!

부모	Process	HT	자식	Process	HT
127	0x1200	Y	127	0x1200	Y
168	0x1680	N			
256	0x2400	Y	256	0x2400	Y

= Bool binheritedHandles 함수가 HT 상속여부 결정

0x1200  
kernel obj  
VC = 2

0x1680  
kernel obj  
VC = 1

0x2400  
kernel obj  
VC = 2

ex)

부모 HT

127 0x1700 Y

CreateProcess()

자식 HT

127 0x1700 Y

0x1700  
kernel obj  
VC = 2

Mailslot → Receiver Process