

8 장 윈도 95 의 통신

윈도 95 의 통신 구조는 윈도 3.x 의 악명높았던 통신 구조에 비해 크게 개선되었다. 윈도 3.x 의 통신 드라이버가 악명높았던 이유는 바로 윈도 3.0 이 개발되던 당시의 모뎀 하드웨어 환경때문이라고도 할 수 있다. 윈도 3.0 이 처음 나왔을 때는 주로 2400bps 모뎀이 주로 사용되고 있었기 때문에, 이 때 개발된 윈도 3.0 의 통신구조는 당시 하드웨어 환경에서는 별다른 어려움없이 사용될 수 있었다. 하지만, 윈도 3.1 이 시장에 나왔을 때는 이미 9600bps 이상의 모뎀이 많이 나오고 있는 상황이었다. 따라서 2400bps 같은 속도가 느린 모뎀을 생각하고 만든 윈도의 통신드라이버는 여러 가지 문제를 일으킬 수밖에 없었다. 특히 고속 통신에서 문제가 발생했는데, 따라서 상용 통신프로그램을 판매하는 회사들은 부득이 윈도의 통신드라이버를 사용하지 않고, 자체적으로 제작한 통신드라이버를 사용하는 경우가 많았다.

윈도 95 의 통신구조는 윈도 3.1 에 비해 다음과 같은 점에서 개선되었다.

고속통신에서 통신포트로 들어오는 문자를 읽지 못하거나 통신포트에 써넣은 문자를 송신하지 못하는 경우가 거의 발생하지 않게되었다. 또한 윈도 95 의 통신 시스템은 선점형 멀티태스킹을 이용하기 때문에 응용프로그램에 대한 응답성이 훨씬 좋아졌고, 이에 따라 보다 많은 양의 자료를 처리하더라도 문제가 발생하지 않게 되었다. 윈도 3.1 에서는 파일을 송수신하는 작업같이 통신포트를 통한 입출력이 빈번하게 발생하는 작업이 지속되는 경우에 다른 응용프로그램의 수행속도가 현저히 떨어지게 된다. 또한 그런 작업 중에 다른 응용프로그램에서 시간이 많이 걸리는 작업을 계속 수행하고 있다면, 파일 송수신 부분에서 통신포트를 통한 자료의 입출력이 지연될 경우가 많다. 하지만 윈도 95 에서는 선점형 멀티태스킹 환경이기 때문에 이런 일이 잘 발생하지 않게 되었다.

그렇다면, 실제로 어떤 점들이 달라졌길래 이런 성능향상을 이루게 되었는지 알아보자.

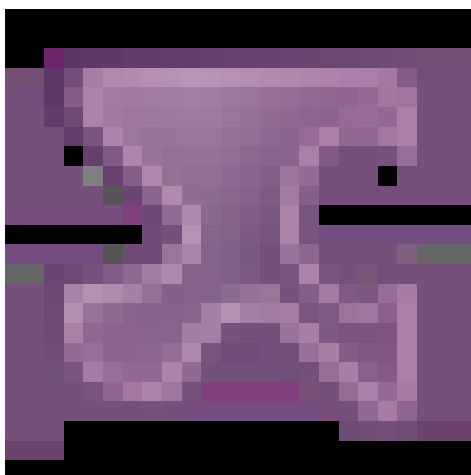
개선된 점

커널 개선

모뎀과 같은 직렬통신 장치는 통신 포트를 통해 하나의 문자가 들어올 때 마다 하나의 인터럽트가 발생하는데, 고속으로 다량의 자료가 통신포트를 통해 흘러들어오게 되면, 미처 처리하지 못하는 경우가 발생할 수 있다. 통신 장치를 통해 들어오는 정보를 고속으로 처리하려면, 이들 데이터에 대해 아주 빠르게 반응해서 인터럽트를 발생시켜 주어야 한다. 하지만, 윈도우 3.1의 통신 드라이버는 9600bps 정도의 속도가 거의 한계점이라고 할 수 있다. 윈도우 95의 통신 드라이버는 그 응답성을 중점적으로 개선해서 이제, CPU나 통신포트의 종류에 따른 속도 제한만이 있다고 보아도 될 것이다.

개선된 윈도우 95의 통신 구조

윈도우 3.1에서는 COMM.DRV라는 통신 드라이버가 있었는데, API 함수를 이용해서 통신 프로그램을 작성하면, 바로 이 통신 드라이버를 통해 실제 직렬포트와 상호작용을 할 수 있었다. 이와 같은 구조에서는 직렬 포트나 모뎀에 새로운 기능이 추가되는 경우, 이 COMM.DRV를 완전히 교체해 주어야만 한다. 다음 그림은 윈도우 3.1상의 통신 구조이다.



하지만, 윈도우 95에서는 VCOMM이라는 새로운 계층이 생겼는데, 이 VCOMM 통신 드라이버는 직렬 포트 드라이버와 모뎀 드라이버 사이의 완충 역할을 수행한다. 윈도우 3.1이 COMM.DRV 하나로 직접 하드웨어와 연결되었던데 비해, 윈도우 95는 모뎀 드라이버, VCOMM과 직렬포트 드라이버 세 부분으로 구성되어 있다. 다음 그림은 윈도우 95상의 통신 구조이다.



새로운 통신장치와 하드웨어 지원

가장 두드러진 개선점은 바로, 윈도우 95에 와서는 16550A UART FIFO를 지원하게 되었다는 점이다. 16550A UART에는 인터럽트 지연시간으로 인해 넘침(overflow)현상을 막기 위해 16바이트의 FIFO 버퍼가 마련되어 있다. 윈도우 3.1의 통신 드라이버는 바로 이 16550A

UART 를 완전히 지원하지 못해 몇몇 통신 응용프로그램 개발자들은 윈도우 통신 드라이버 사용을 포기하고, 자체 개발한 고속 통신 드라이버를 사용하기도 했었다.

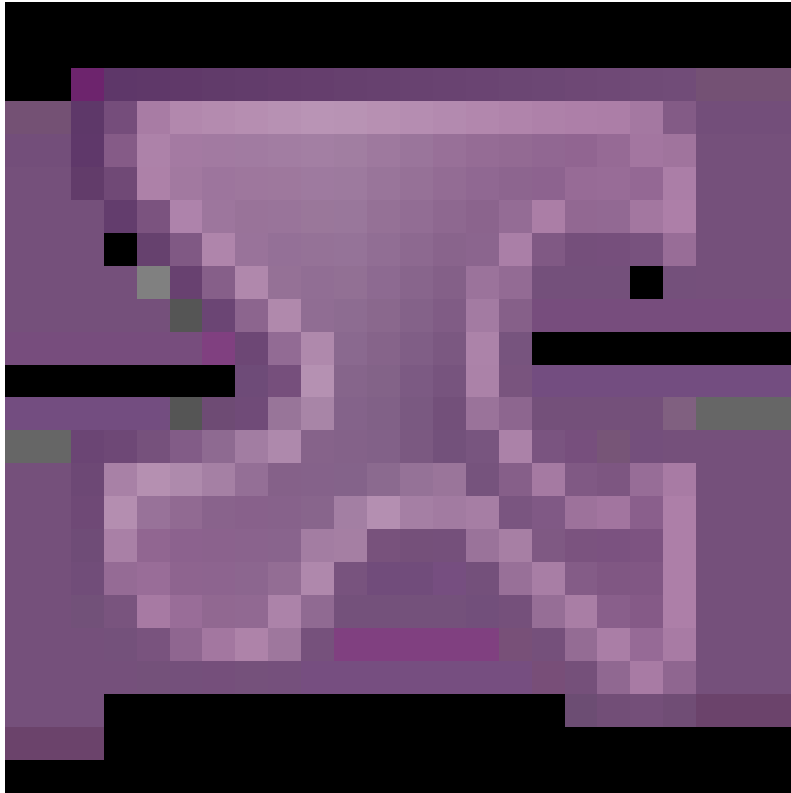
또한 윈도우 3.1 에서는 논리적으로 직렬통신 포트 9 개와 병렬 통신 포트 4 개가 지원되었다. 하지만 윈도우 95 의 통신 API 는 직렬 통신 포트와 병렬통신 포트 모두 128 개까지 논리적으로 지원가능하게 되었다.

TAPI(Telephony API)

윈도즈의 TAPI 는 마이크로소프트의 WOSA(Windows Open Services Architecture) 중의 한부분이다.

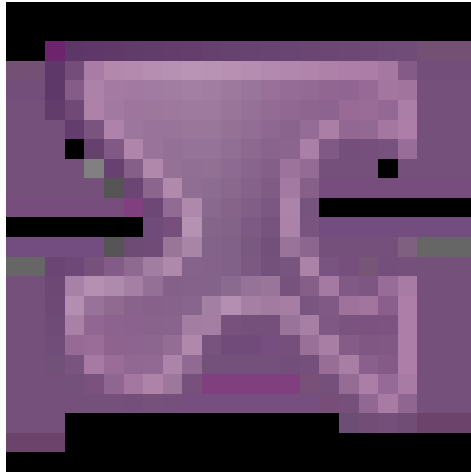
WOSA 란 기업 컴퓨팅 환경에 단일하고 개방된 사용자 인터페이스를 제공하기 위해 마이크로소프트에서 계획하고 있는 것이다. WOSA 인터페이스를 이용한 윈도우 기반의 응용프로그램은 사용자에게는, 현재 사용하고 있는 네트워크의 종류가 어떤 것인지, 컴퓨터는 어떤 것인지 등에 대한 지식없이도 원하는 정보에 쉽게 접근할 수 있게 해준다. 컴퓨터 환경이나 정보 자원에 변화가 생기더라도 WOSA 를 이용한 응용프로그램은 잘 동작한다. WOSA 는 일반 응용 프로그램 서비스, 통신 서비스, 버티컬 마켓 서비스 세부분으로 이루어 진다.

TAPI 도 다른 WOSA 서비스와 마찬가지로 개발자들이 사용하는 응용 프로그램 인터페이스(Application Programming Interface:API)와 하드웨어 개발자가 제공하는 서비스 제공자 인터페이스(Service Provider Interface:SPI)로 구성된다.

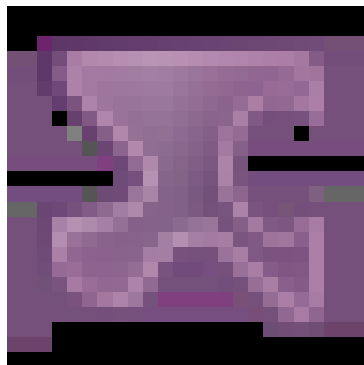


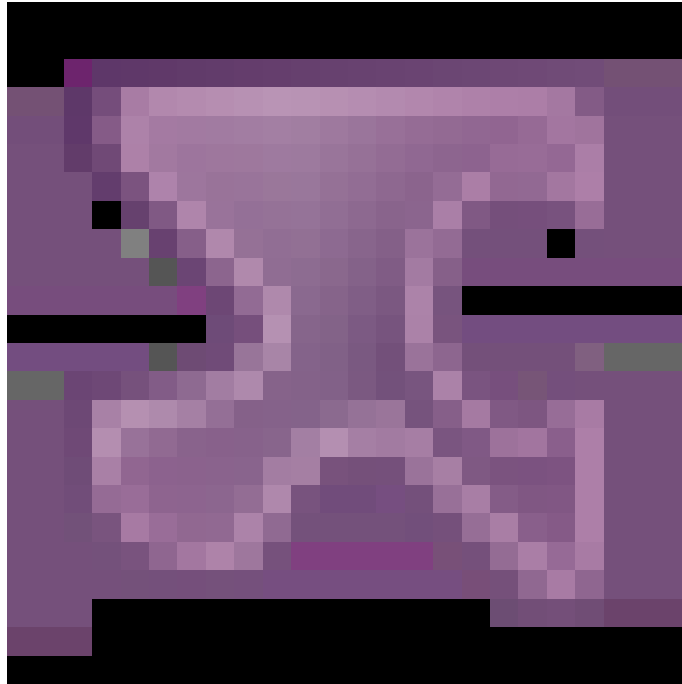
TAPI는 접속확인, 응답, 전화 끊기 등 기본 기능과, PBX, ISDN 같은 데서 볼 수 있는 추가적인 기능들도 모두 포함되어 있다.

윈도 95가 나오면서 TAPI라는 것이 처음 소개될 때, 잘못 알려진 것 중 하나가 바로, TAPI를 이용하면, 통신장치를 공유할 수 있다는 것이었다. 얼핏 듣기에는 어떤 한 프로그램이 TAPI를 이용해서 통신장치를 사용하고 있을 때, 다른 프로그램이 TAPI를 이용한다면, 동시에 이 통신자원에 접근할 수 있다는 말처럼 들렸기 때문이다. 사실, 그럴 수는 없는 일 아닌가? 실제로 두 프로그램에서 동시에 모뎀같은 통신자원에 접근할 수 있는지 살펴보자. 윈도 95의 보조프로그램 중 하이퍼터미널과 전화걸기 프로그램은 모두 TAPI를 이용하는 응용프로그램이다.

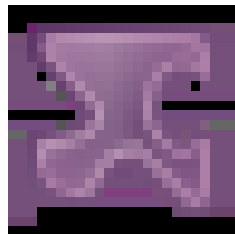


먼저 하이퍼터미널 프로그램을 띄어두고, 전화걸기 프로그램을 실행시켜보자. 전화는 잘 걸린다. 그렇다면, 정말 동시 모뎀 자원을 사용할 수 있는 것일까? 결론부터 말하면 이건 속임수에 불과하다. 하이퍼터미널은 기동되면서 바로 통신자원을 잡고 계속 사용하는 것이 아니다. 전화를 걸어 연결되거나 또는 전화걸기를 취소하고 직접 터미널에 글자를 입력하면 그 때서야 통신자원을 사용하게 되는 것이다.



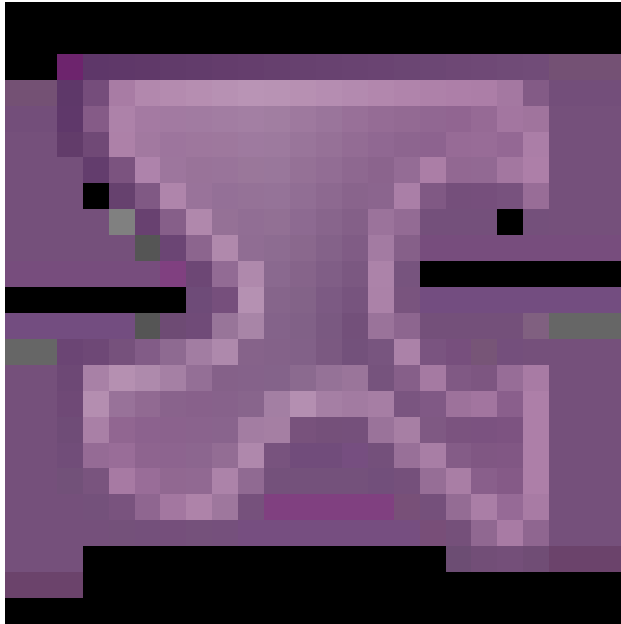


이 상태에서 전화걸기를 실행시켜 전화를 걸어보면 다음과 같은 오류 메시지를 보게 될 것이다.

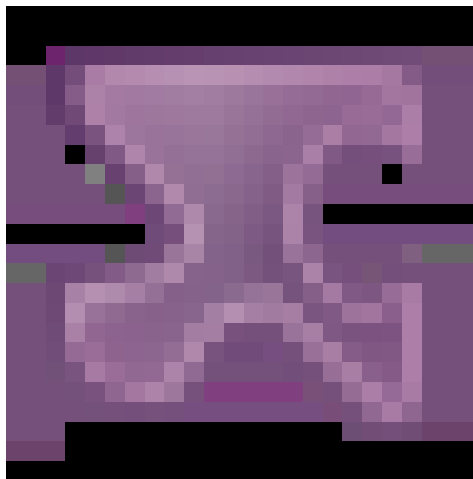


그렇다면, 개발자 입장에서 보면 편리해 진지 모르겠지만, 사용자 입장에서 보면 별로 달라진 것도 없는 것일까? 그렇지 않다. 사용자 입장에서 보면 가장 편리해진 점은 바로 모뎀 설치와 모뎀 환경 설정에 대한 것이다. TAPI 를 사용하지 않는 경우라면, 모든 전화관련 응용 프로그램은 각각 모뎀이 연결된 통신포트와 모뎀 속도, 패리티 비트, 정지비트, 데이터비트 따위의 값들을 일일이 설정해 주어야 했다. 하지만 TAPI 를 사용하는 응용프로그램은 이런 작업들이 대부분 처음 시스템을 설치할 때 자동으로 수행되면, 모뎀이 장착된 모뎀 포트도 자동으로 검색한다. 그리고, 모뎀에 대한 설정도 제어판에 있는 모뎀을 이용해서 설정해 두면, 모든 TAPI 를 이용하는 응용프로그램들에 영향을 미치게 된다. 한가지 주의할 점은 TAPI 를 사용하는 경우에 대해서만 그렇다는 것이다. 그러니까 기존의 3.1 용의 통신프로그램들이나, 윈도우 95 용으로 작성되었더라도 TAPI 를 사용하지 않은 통신프로그램들은 여전히 각각 설정해 주어야 한다.

TAPI를 사용하는 응용프로그램들에서 모뎀 환경설정을 해 보면, 제어판의 모뎀 항목의 설정 때와 같은 대화상자를 만날 수 있는데, 이것은 모뎀 환경 설정 대화상자가 TAPI에서 제공되기 때문이다.



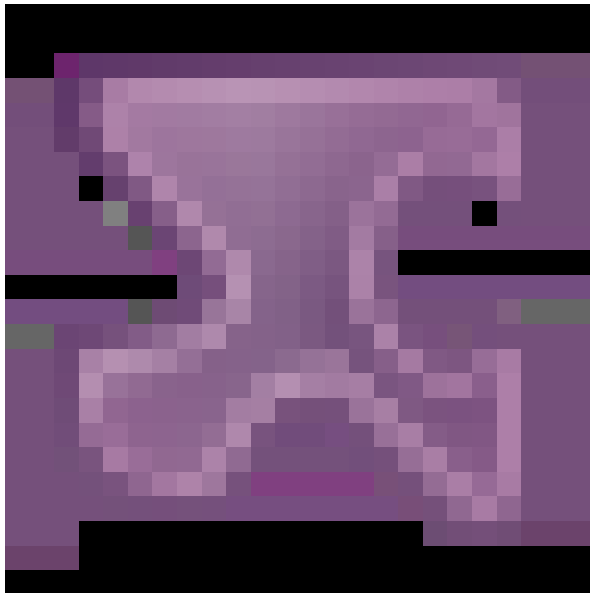
모뎀이 설치되어 있지 않은 경우, 제어판의 모뎀을 더블클릭해 보면,



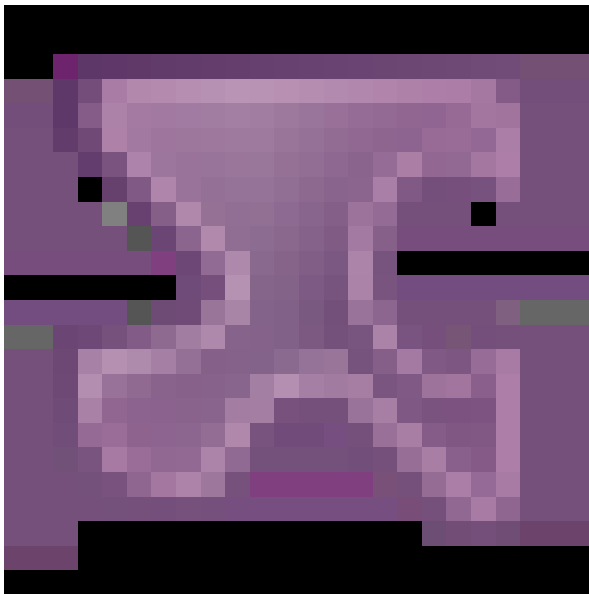
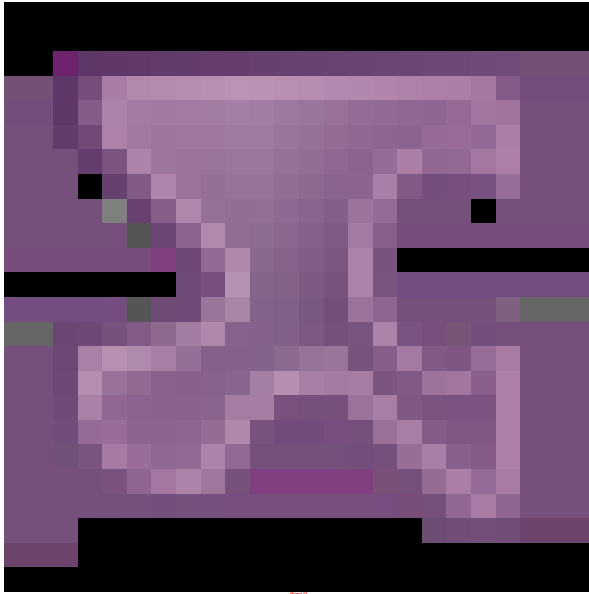
위와 같은 대화상자가 나타난다.  을 눌러보면 자동으로 모뎀을 찾는 다음과 같은 대화상자가 나타난다.



모뎀을 통신포트에서 찾으면, 이제 모뎀을 설정하게 된다.



모뎀 등록정보에서는 모뎀 자체의 등록정보와 전화걸기에 대한 등록정보등을 설정할 수 있다.



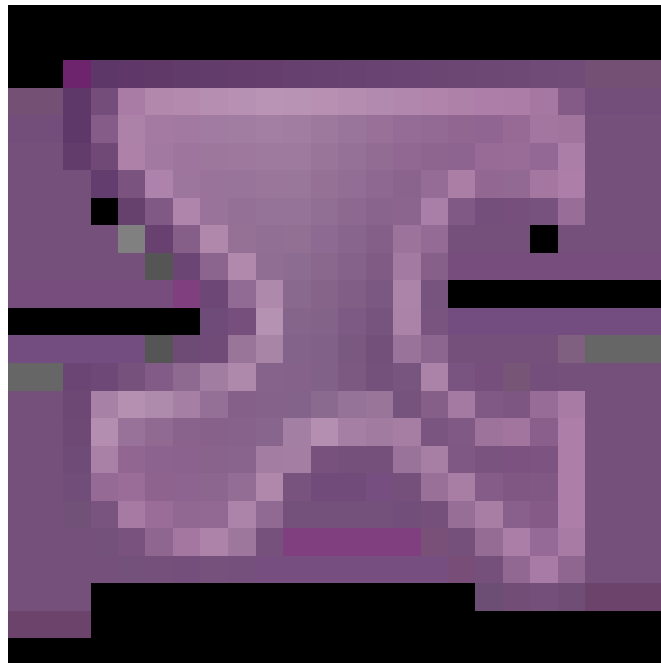
모뎀 등록 정보를 통해 연결된 통신포트와 최대 속도등을 볼 수 있다. 물론 그 설정값도 바꿀 수 있다.

전화 걸기 등록 정보에서는 위치 정보와 톤방식과 펄스방식등을 설정할 수 있다.

위치 지정이란, 노트북 컴퓨터처럼 여기저기 들고 다니는 경우에 유용하게 사용할 수 있는데, 하이퍼터미널 같은 프로그램에서 전화번호를 등록해 두었는데, 이 번호가 모두 서울 전화번호를 기준으로 등록되어 있다고 하자. 이럴 경우, 내가 지역번호가 032 인 지방에 내려가

이 노트북으로 전화를 걸면 어떻게 될까? 지역번호 02 를 누르지 않기 때문에 엉뚱한 곳에 걸리거나 또는 "지금 거신 전화는 국번이 없거나...."라는 말을 듣게 될 것이다. 이 때, 이 전화 걸기 등록정보에서 사용자 위치를 하나 등록해 두고, 예를 들어, "인천 사무소"라고 등록하자. 지역번호에는 032 를 입력해 둔다. 이렇게 한 다음 사용자 위치를 "기본 위치"에서 "인천 사무소"라고 바꾼 다음, 하이퍼터미널로 전화를 걸어보라. 신기하게도 지역번호 02 가 자동으로 들어가게 되는 것을 볼 수 있을 것이다.

톤 방식이란 요즘 쓰이는 전자식 전화기에서 사용하는 방식이고, 펄스 방식이란 옛날에 다이얼을 돌리던 시절에 쓰던 방식을 말한다. 펄스 방식으로 설정하고 전화를 걸면, 옛날 그 추억의 다이얼 소리를 들을 수도 있다.



모뎀 등록 정보에서 "진단" 탭을 눌러보면, 현재 모뎀이 어느 통신포트에 연결되어 있는지 볼 수 있다. 또한 [추가 정보] 버튼을 눌러, 이 통신포트에 접속되어 있는 모뎀이 정상적으로 동작하는지도 알 수 있다.

