

5장 파일 전송

컴퓨터 간에 파일을 송수신할 때는 미리 두 컴퓨터 사이에 어떤 약속 같은 것을 정해 두어야 한다. 이 약속에는 어떤 식으로 자료를 주고 받을 것인지가 들어 있는데, 예를 들어, 보내는 데이터 블록의 크기나 오류 검출 및 복구 등에 관한 것이다. 이 약속은 컴퓨터 양쪽에서 동일한 것을 사용해야 하는데, 이 약속을 파일 전송 규약, 또는 프로토콜이라고 한다. 요즘 가장 많이 사용되는 것이 바로 ZMODEM이란 전송규약인데, 대부분의 PC통신 서비스 업체에서 이 전송규약을 지원하며, 또한 대부분의 통신 에뮬레이터 역시 이 전송규약을 따른다. PC통신에서 자료를 올리거나(upload) 내려 받을 때(download) 항상, 어떤 전송규약, 또는 프로토콜을 선택하라고 나오는데, 이렇게 전송규약을 선택해야만 양쪽에서 같은 파일 전송 통신 규약을 사용해서 파일을 주고 받을 수 있게 되는 것이다.

그렇다면 프로토콜의 원래 의미는 무엇일까? '프로토콜'이란 단어를 사전에서 찾아보면 다음과 같다.

protocol

- 1.원안, 조약안, 의정서, 조서
2. (외교상의) 의전, 의례, 의식
3. (로마교황의 교칙등의) 정식문
4. 「전산」 프로토콜(컴퓨터 상호간의 대화에 필요한 통신규약)

데이터를 송수신할 때 송신된 데이터를 수신한 측에서 제대로 받았는지, 도중에 오류가 발생해서 데이터가 손상된 것은 아닌지를 판단해서 제대로 도착했으면 제대로 도착했으니 다음 데이터를 보내라고 하고, 제대로 도착하지 않았다면 손상된 데이터를 다시 송신해 줄 것을 요청하는 등의 일련의 과정들을 규약화 해 놓은 것이다. 따라서 파일을 주고 받을 때는 서로 같은 파일 전송 프로토콜을 사용해야 한다. PC통신에서 자료를 내려 받기 할 때, 먼저 어떤 파일 전송 프로토콜을 사용할 것인지 결정하도록 하는데, 만약 자신의 PC에는 ZMODEM 파일 송수신 프로그램 밖에 없는데 KERMIT같은 전혀 다른 파일 전송 프로토콜을 설정하면 파일 송수신에 실패하게 될 것이다.

파일 전송을 위해 제안된 여러가지 프로토콜들이 있는데, 그 기본적인 구현원리는

대체로 비슷하다. 현재 가장 일반적으로 사용되는 파일 전송 프로토콜은 ZMODEM 프로토콜이다.

파일 전송 통신 규약의 기본 원리는 다음과 같다.

데이터를 보내는 쪽과 받는 쪽은 먼저 주고받을 데이터 블록의 크기를 정해야 한다. 데이터 블록의 크기를 정한 다음에는 데이터의 시작과 끝을 알리는 신호를 정하고 이 신호에 맞게 데이터를 주고 받게 된다. 여기에 데이터 송수신 시에 끼어드는 노이즈로 인해 데이터가 변형되었는지를 검출하기 위해 적당한 오류 검출 코드등도 보내게 된다.

파일 전송 규약에서 사용되는 용어를 정리해 보았다.

용어	설명
내려받기(download)	원격지 시스템에서 파일을 수신한다.
올리기(upload)	원격지 시스템으로 파일을 송신한다.
송신자	다른 시스템으로 파일을 송신하는 시스템
수신자	다른 시스템으로부터 파일을 수신받는 시스템
SOH	Start of Header (ASCII 0x01)
EOT	End of Text(ASCII 0x04)
ACK	Acknowledgement (ASCII 0x06)
NAK	Negative Acknowledgement (ASCII 0x16)
CAN	Cancel (ASCII 0x18)

파일 전송 규약들

KERMIT

시분할 시스템으로 동작되는 중대형 컴퓨터들 사이의 연결에 많이 사용된다. 커밋 규약은 이기종 컴퓨터간의 통신을 위해 설계되었기 때문에 거의 모든 컴퓨터에서 구현되어 사용되고 있다. 전송 효율은 형편없기 때문에 아주 속도가 느리다.

XMODEM

이 파일 전송 규약은 1977년 워드 크리스텐슨이 개발했다. XMODEM은 비동기 직렬 통신 라인을 통해 8비트 데이터를 전송할 수 있도록 해준다.

XMODEM은 전송하려는 데이터 비트는 8비트, 정지비트는 1비트 패리티는 사용하지 않는다.

XMODEM의 기본 구조는 패킷이다. 오류 검출 방법에 따라 132, 133개의 문자들이 하나의 패킷을 이룬다. 128바이트의 데이터와 4바이트, 5바이트의 추가 정보로 구성된 데이터 패킷을 송수신한다. 다음 그림은 XMODEM 패킷의 구조이다.

SOH (1바이트)	일련번호 (1바이트)	일련번호의 1의 보수 (1바이트)	DATA (128바이트)	Error Checksum (1바이트)
---------------	----------------	-----------------------	------------------	--------------------------

첫번째 바이트가 SOH이다. 그 다음 바이트는 이 패킷의 일련번호이다. 이 일련번호는 1부터 시작해서 255까지 연속적으로 붙는다. 각 패킷마다 1씩 증가하는데 255다음은 0이다. 세번째 바이트는 두번째 바이트의 일련번호에 대한 1의 보수이다. 이 값은 다음과 같이 계산할 수 있다.

$$\text{보수} = (255 \& \text{일련번호}) \wedge 255$$

그 다음에는 128바이트의 데이터가 들어간다. 데이터 바로 다음에는 이 데이터에 대한 오류 검사값이다. XMODEM은 두가지 방식의 오류 검사를 지원하는데 하나는 체크섬방식이고 다른 하나는 CRC방식이다. 체크섬이 사용되는 경우에는 오류 검사를 위해 1바이트가 사용되며 이렇게 되면 하나의 XMODEM 패킷의 크기는 132바이트이다. 반면 CRC가 사용되는 경우 오류검사를 위해 두바이트가 사용되며 이 때는 패킷의 크기가 133바이트가 된다.

체크 섬 계산식은 다음과 같다.

$$128\text{바이트 데이터 각 바이트들의 총합} / 256 = \text{체크섬 값}$$

XMODEM으로 파일 전송을 할 때 그 과정을 하나씩 따라가 보자.

데이터 전송을 시작하면, 데이터를 받을 수신자가 자신이 데이터를 받을 준비가 되어 있다는 것을 알리기 위해 NAK신호를 보낸다. 10초안에 송신자에서 데이터가 오지 않으면 다시 NAK를 보낸다. 이것이 10번 이상 반복되면 수신을 중단한다. 송신자에서 데이터가 송신되어 수신자가 송신된 데이터를 받으면 그 체크섬값을 검사해서 데이터가 변형되지 않고 제대로 도착했는지 확인한다. 오류검사에서 이상이 없으면 수신자는 ACK를 송신자에게 보낸다. 송신자는 ACK신호를 받으면 다음 데이터를 보내고 만약 ACK대신 NAK신호가 오면 데이터가 전달도중 손상을 입은 것이니까 이 신호를 재전송해 준다. 데이터의 전송이 모두 끝나면 EOT신호를 보내고 수신자가 ACK 신호를 보내면 전송은 마치게 된다.

파일 송수신 중 전송을 중단하기 위해서는 CAN 제어 문자를 사용한다.

XMODEM은 구현이 간단하고 사용이 간편한 반면 오류 검출 기능이 약하고 한번에 한 파일 밖에 전송하지 못하며 파일 이름을 보낼 수 없기 때문에 파일을 수신하는 쪽에서 파일 이름을 지정해 주어야 하는 불편이 있다.

YMODEM

YMODEM은 1984년에 나온 프로토콜로 XMODEM에 비해 몇가지 장점을 가진다. YMODEM은 XMODEM과 달리 데이터 블록의 크기가 128바이트로 지정되어 있는 것이 아니라 1KB도 사용이 가능하며 동시에 여러개의 파일을 보낼 수도 있다. 데이터 블록의 크기가 128바이트와 1KB 두가지가 존재하는데, 이는 상황에 따라 파일 전송에 효율을 기할 수 있게 하기 위해서 이다. 당연한 이야기겠지만 1KB의 데이터 블록을 갖는 YMODEM 패킷은 전송 속도는 증가하지만 오류가 발생할 확률이 그만큼 증가하게 되며 128바이트 데이터블록을 갖는 YMODEM 패킷은 전송 속도는 1KB에 비해 떨어지지만 데이터의 안정성은 낫다. 따라서 회선의 상태가 비교적 좋은 곳에서는 1KB 데이터 블록을 갖는 YMODEM 패킷을 사용하는 것이 좋겠고 그렇지 않은 경우에는 128바이트 짜리 데이터 블록을 갖는 YMODEM 패킷을 사용하는 것이 낫다.

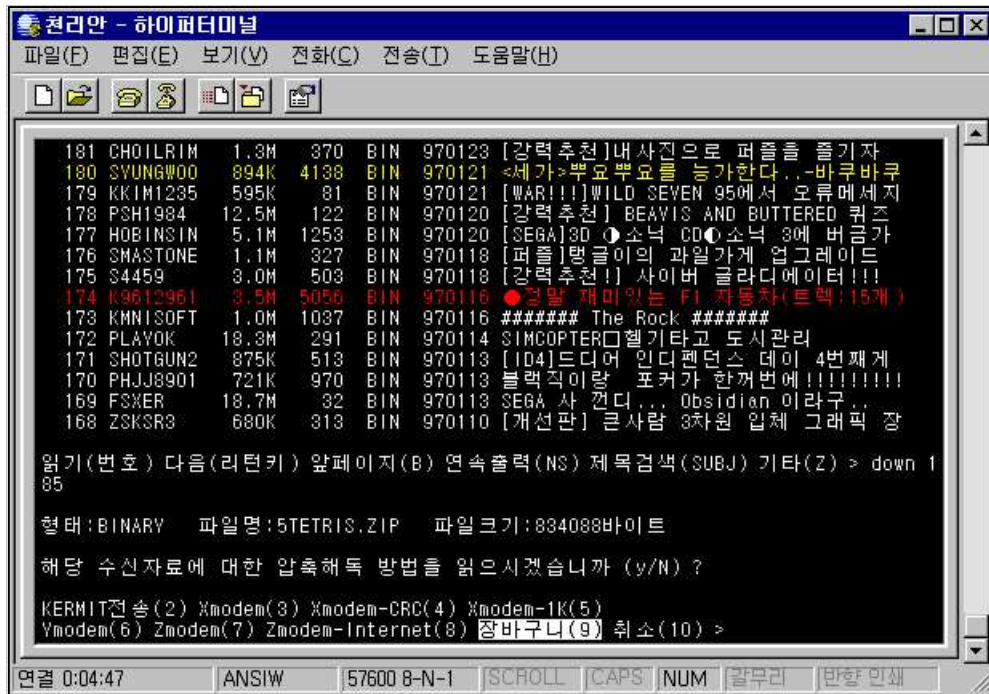
SOH (1바이트)	일련번호 (1바이트)	일련번호의 1의 보수 (1바이트)	DATA (128바이트)	CRC-16 (2바이트)
---------------	----------------	-----------------------	------------------	------------------

YMODEM은 파일 정보를 수신자에게 보내기 위해 실제로 파일의 내용을 보내기 전에 파일에 대한 정보를 보내기 위해 128바이트 크기의 패킷을 보낸다. 이 패킷에는 소문자로 구성된 파일이름과 그 크기 파일이 만들어진 날짜가 전달된다. 파일이름은 ASCIIZ문자열이고 디렉토리가 포함될 경우에는 \대신 /가 들어gå야하며, 파일의 크기는 10진수 문자열로 전송된다.

YMODEM은 XMODEM의 단점을 일부 보완하기는 했지만 여전히 7비트 데이터를 보낼 수 없다는 것과 패킷교환망에서 사용하는 제어문자를 처리하지 않기 때문에 패킷망에서는 사용할 수 없다는 단점이 있다.

ZMODEM

PC통신망에서 아마도 지금 가장 널리 쓰이는 파일 전송 규약이 바로 ZMODEM일 것이다. PC통신망에서 자료를 올리거나 내려 받을 때 무심코 선택한 Z-MODEM이 바로 이것이다.



1986년 패킷 스위치 통신망을 위한 새로운 파일 전송 프로그램이 개발되었는데 이 파일 전송 프로그램은 패킷 통신망에서 효율적으로 사용될 수 있었다. 이 결과물이 바로 ZMODEM이다. ZMODEM이란 이름은 아마도 XMODEM, YMODEM의 연장선 상에 있는 프로토콜이라는 상상을 하게 한다. 하지만 사실 ZMODEM프로토콜은 완전히 새로운 프로토콜이다.

ZMODEM의 중요한 특징들을 살펴보면 다음과 같다.

1. 모든 상호작용은 16비트, 32비트 CRC로 보호된다.
2. 데이터를 하나의 연속된 패킷 스트림으로, 인증을 기다리지 않고 보낸다. 수신자가 오류를 보고하기 전에는 보내기를 멈추지 않는다. 이렇게 연속적인 스트림으로 데이터를 보내면 패킷 스위치 망이나 버퍼가 되는 모뎀에서는 전송속도가 아주 크게 향상된다.
3. 오류 복구를 위해 각 패킷에 고유번호를 붙인다. 수신자는 항상 어느 데이터 블록이 파일로 저장되기로 되었는지를 알고 있다. 이러한 고유번호 붙이기는 파일 전송 복구에도 도움이 된다. 사용자가 전송을 중단한 그 지점에서 다시 전송을 계속할 수 있다.

이런 특징들 때문에 오늘날 ZMODEM은 가장 널리 쓰이는 프로토콜이 되었다.