6 장 터미널 에뮬레이터

통신 프로그램을 흔히들 에뮬레이터(emulator)라고 부른다. 왜 통신 프로그램을 통신 에뮬레이터라고 부를까 하는 의문을 가져 보았을 것이다. 에뮬레이터를 사전에서 찾아보면

emulator : 경쟁자, 모방자

「전산」에뮬레이션을 하는 장치, 프로그램

여기에서 다시, 에뮬레이션(emulation)의 뜻을 찾아보자.

emulation : 경쟁, 겨룸, 대항

「전산」다른 컴퓨터의 기계어 명령 대로 실행 할 수 있는 기능

정리해 보면

전산 분야에서 사용되는 용어로서, 다른 컴퓨터의 기계어 명령대로 실행 할 수 있는 기능을 하는 장치, 프로그램

이란 뜻이다.

에뮬레이터란 말의 유래는 하나의 주 컴퓨터에 수십, 수백개의 단말기(terminal)를 연결해서 사용하던 시절로 거슬러 올라간다. 각 단말기는 요즘의 PC 와는 달리 혼자서는 아무 일도 할 수 없는 말 그대로 바보 단말기(dummy terminal)였다. 단말기에서 어떤 명령을 입력하

면 이 입력한 명령이 주 컴퓨터로 전달되고 명령을 전달 받은 주 컴퓨터는 명령에 해당하는 작업을 수행한다. 그 다음 그 작업 결과를 다시 단말기로 보내주고, 단말기는 주 컴퓨터가 보낸 결과를 화면에 뿌려주는 역할만을 하는 방식이었다. 통신 프로그램이 바로 이 단말기를 흉내내는 프로그램이란 뜻으로 터미널 에뮬레이터란 이름이 붙었던 것이다. 통신 프로그램의 기능이라는 것을 간단하게 정리한다면, 통신포트로 들어온 데이타를 화면에 뿌려주고, 사용자가 입력한 문자를 통신포트를 통해 보내준다. 이 뿐이다. 다른 것은 아무 것도 없다. 이 기능만 구현해 주면 PC 통신 업체에 당장이라도 접속할 수 있다. 전화걸기, 전화끊기 같은 기능은 어떻하냐고? 그런 기능은 모두 모뎀에서 수행해 주는 기능이다. 프로그래머는 사용자가 입력하는 문자를 직렬포트에 써주면 직렬포트는 이 데이타를 모뎀에 8 비트 스트림으로 나누어 보내고 모뎀은 이 비트 스트림들을 해석해서 모뎀 명령어이면 해당 명령을 수행하고 아니면 무시하는 것이다. 즉, "ATDT 01410"라고 입력하고 엔터를 치면 "ATDT 01410\r"이 직렬포트를 통해 모뎀으로 전달되고, 이 데이타를 받은 모뎀은 01410 으로 톤 방식으로(T) 전화를 걸라(ATD)는 명령이로구나 하고 해석하고 전화를 건다. 명령을 해석해서 전화는 거는 것은 모뎀 내부에서 벌어지는 일이므로 사실 에뮬레이터에서 해 주어햐 하는 일은 단지 포트에 들어온 데이타를 빠뜨리지 않고 읽어오고, 입력한 데이타를 통신포트를 통해 모뎀에 전달해 주는 일 뿐이다. 이렇게 글자만 그대로 전송하는 터미널을 TTY(TeleTYpe)터미널이라 부른다. 그러니까 앞에서 말한 기능을 구현한 통신 프로그램을 작성했다면 이것은 TTY 터미널 에뮬레이터라고 부를 수 있겠다. 하지만 TTY 터미널 에뮬레이터로 PC 통신에 접속해 보면 다른 이야기 같은 에뮬레이터로 접속했을 때와는 많이 다르다는 느낌을 받을 것이다 . PC 통신 회사들이 운영하는 서비스들은 대부분 다른 방식의 터미널을 지원하기 때문이다.

사실 처음 통신용 프로그램은 현재처럼 PC 통신에 접속하기 위해 만들어진 것이라기 보다는 PC 를 대형 기종의 컴퓨터의 단말기로도 사용할 수 있도록 만들어 졌다고 볼 수 있다 . 그런데 그 바보 단말기에도 아주 단순한 기능이 있었는데 바로 문자를 표시할 때, 반전시켜서 표시한다거나 깜박거리게 또는 밑줄 등 여러가지 다양한 모양을 글자를 출력할 수 있게 해 주는 기능이었다. 이를 위해 주 컴퓨터와 단말기 사이에 정해놓은 제어코드 규약이 있는데, 지금처럼 개인용 컴퓨터가 발달하기 이전에는 주컴퓨터를 팔아먹는 회사에서 단말기까지 일괄적으로 판매하는 경우가 많았으므로 다른 회사의 단말기를 사용하지 못하도록 자기 회사의 주 컴퓨터에서 정한 제어코드규약 만 따르는 단말기들을 개발해서 판매를 했다. 주 컴퓨터 판매 회사들은 앞다투어 자기들만의 터미널들을 발표하기 시작했다. 따라서 당연한 일이지만 미국 표준 기구(ANSI:American National Standard Institute)에서는 X3.34 사양에 의거 터미널 표준을 도입하려고 했다. 이 터미널 표준을 ANSI 터미널이라고 한다. 하지만 이전의 터미널들도 여전히 사용되고 있다. 여기에서 우리는 ANSI 터미널을 주로 살펴보도록 하겠다.

ANSI 터미널

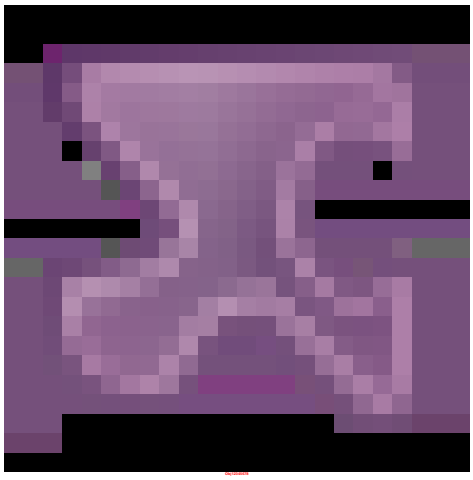
ANSI 터미널은 DEC 사의 VT-100 터미널과 거의 비슷하다. ANSI 터미널 에뮬레이터로 국내 PC 통신망에 접속해 보면 TTY 터미널과는 달리 깨끗한 화면을 볼 수 있을 것이다.

터미널 제어코드는 보통 다음과 같은 형식으로 되어 있다.

ESC[코드

여기서 ESC 는 물론 ASCII 의 Esc 문자, 즉 ASCII 27 (0x1B)이다. 두개 이상의 인자가 필요한 제어코드(예를 들어 커서를 임의의 위치로 옮기는 경우 row 값과 column 값이 필요하다)는 두 값은 세미콜론(;)으로 구분한다.

그렇다면 여기서 한가지 짚고 넘어가야 할 것이 있다. ANSI 에서는 터미널의 화면 좌표를 어떤 방식으로 잡을까? 화면 왼쪽 위가 0,0 이며, 가로 80, 세로 25 가 그 크기이다. 또한 화면 좌표는 row 와 column 을 이용해서 표시하는데, 일반적으로 많이 사용하는 x-y 좌표로 생각하면, row 는 y 좌표에 대응되고, column 은 x 좌표에 대응된다. 다음은 안시 터미널의 좌표계이다.



ANSI 제어코드가 대소문자를 구별해서 사용한다는 점도 주의해야 한다.

커서 이동 코드

1. 커서 위치 이동 ESC[행;열H

   커서를 (행,열)위치로 이동한다. 화면 왼쪽 위가 (0,0)이다.

2. 커서 위치 이동 ESC[행;열f

   커서를 (행,열)위치로 이동한다. 화면 왼쪽 위가 (0,0)이다.

3. 커서를 위로 이동 ESC[#A

   커서를 #줄 만큼 위로 이동한다. 커서가 화면 제일 위에 있다면 이 제어코드는 무시된다.

4. 커서를 아래로 이동 ESC[#B

   커서를 #줄 만큼 아래로 이동한다. 커서가 제일 아래 줄에 있다면 이 제어코드는 무시된다.

5. 커서를 오른쪽으로 이동 ESC[#C

   커서를 #만큼 오른쪽으로 이동한다. 커서가 화면 제일 오른쪽 끝에 있으면 이 제어코드는 무시된다.

6. 커서를 왼쪽으로 이동 ESC[#D

   커서를 #만큼 왼쪽으로 이동한다. 커서가 화면 제일 왼쪽 끝에 있으면 이 제어코드는 무시된다.

7. 커서 위치 저장 ESC[s

현재의 커서 위치를 저장한다.

8. 커서 위치 복원 ESC[u

저장된 커서 위치로 커서 위치를 옮긴다.

화면 지우기

9. 화면 지움 ESC[2J

전체 화면을 지우고 커서를 0,0 으로 옮긴다.

10. 한줄 지움 ESC[K

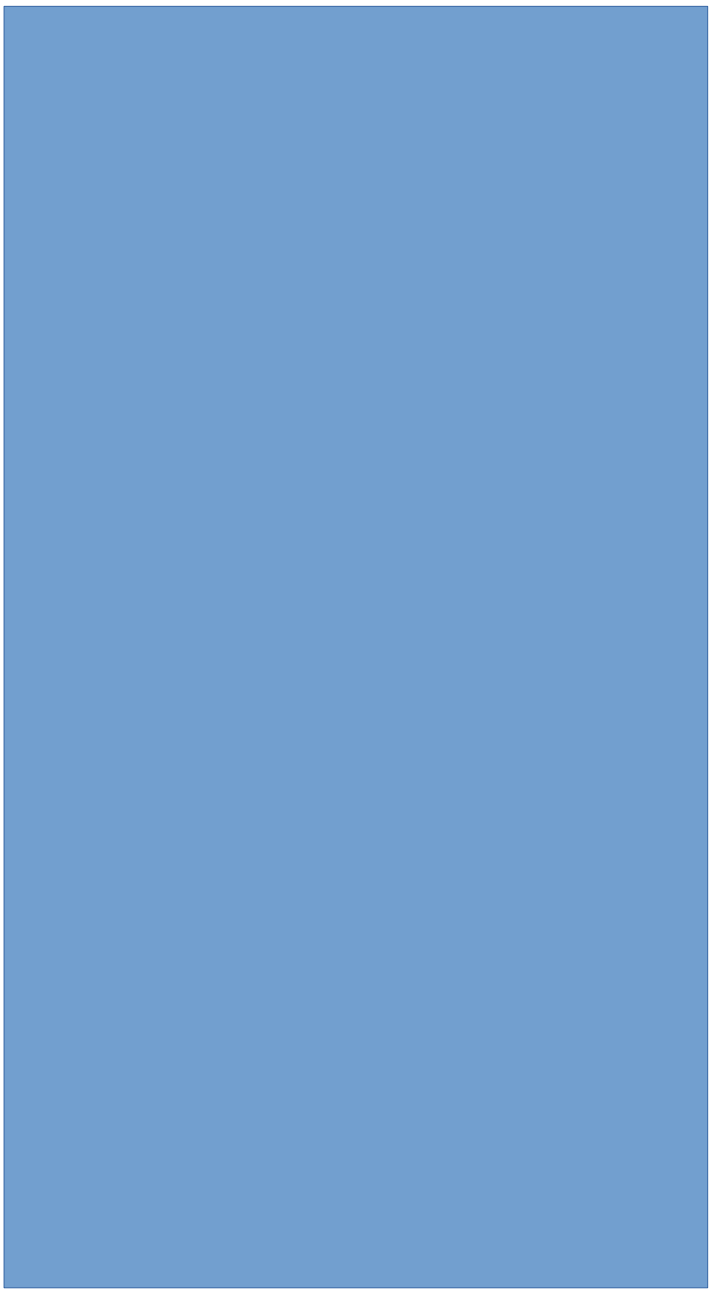
현재 커서가 위치한 줄을 지운다. 커서 위치는 변하지 않는다.

화면과 문자의 속성

11. 색깔 지정 ESC[#;...;#m

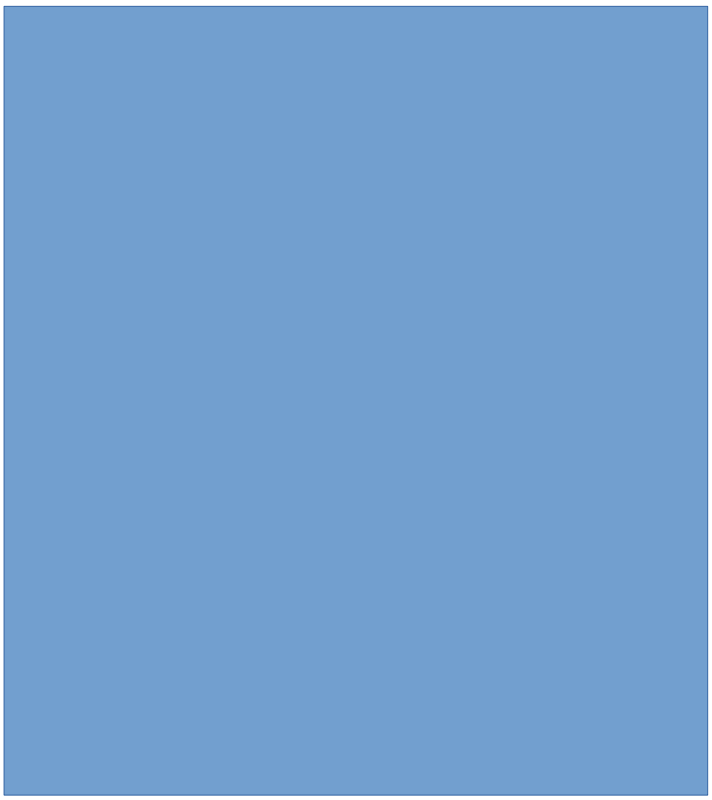
화면의 색을 바꾼다.

#값의 의미는 다음 표와 같다.



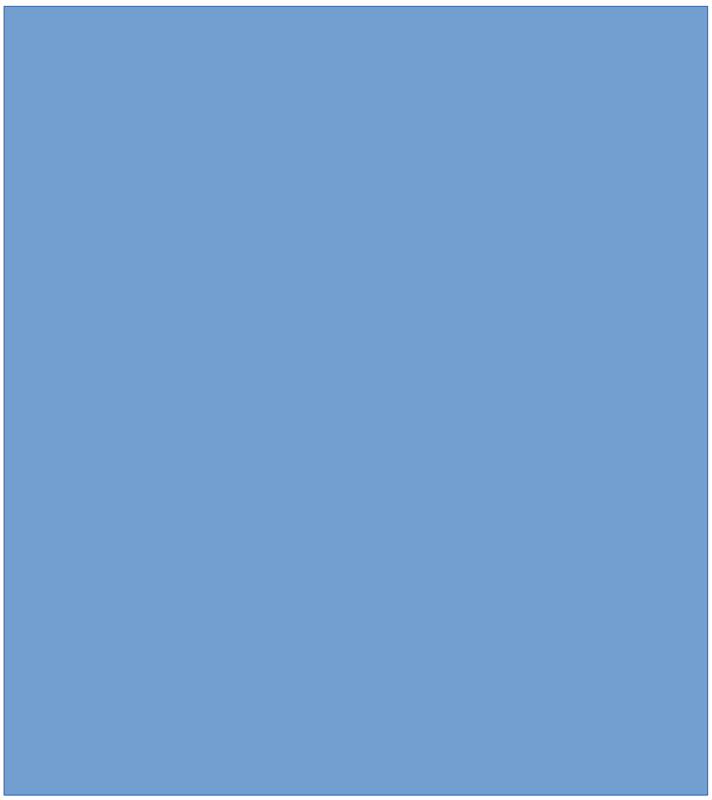
12. 화면 선택 ESC[=#h

화면의 넓이와 높이를 지정한다.



13. 화면 선택 ESC[=#l

화면의 넓이와 높이를 지정한다. 단지 ESC[7h 와는 달리 ESC[7l 은 자동 줄넘김이 OFF 인 것만 다르다.



X3.64 는 배경이 검은색에 흰글자가 나오는 보통의 화면 모드에서 시작하면 자동 줄넘김도 OFF 인 상태에서 시작한다.

기타 터미널 코드

안시 터미널 코드 이외에도 여러가지 종류의 터미널 코드들이 존재하며 여전히 이들 터미널 코드들 또한 널리 사용된다.

이중에서 VT-100, VT-220 과 FS-220B 의 제어 코드를 살펴보자

VT-100

VT-100 은 ANSI 터미널과 거의 비슷하다. ANSI 터미널에 몇가지 코드가 추가된 형태이다.

VT-220

VT-220 은 VT-100 에 몇가지 제어 코드가 다시 추가된 형태이다.

FS-220B