Chapter 15. 소켓과 표준 내용 확인 문제

1. 표준 입출력 함수를 사용할 때 얻게 되는 장점 두 가지는 무엇인가? 그리고 그 두 가지 장점을 얻게 되는 이유는 무엇인가?

표준 입출력 함수를 사용하게 되면, 버퍼링을 통한 성능향상과 좋은 이식성 장점을 얻게 된다.

표준 입출력 함수는 시스템(운영체제)에 관계없이 컴파일러가 지원하도록 ANSI C에서 표준으로 정의 하였기 때문이다.

여러 번의 데이터를 전송해야 할 것을 버퍼링을 이용하면 많은 데이터를 묶은 패킷으로 전송하게 된다. 패킷에는 헤더 정보가 추가되며, 이는 전송되는 데이터 크기의 관계없이 일정한 크기 구조를 갖는다.

따라서, 전송 단위인 패킷이 증가할 수록, 헤더 정보의 전달도 많이 수반된다.

그리고 출력버퍼로 데이터를 전달하는 데에도 많은 시간이 소요된다. 전달횟수(패킷)이 증가될 수록 데이터 전송 시간도 길어질 수밖에 없다. 버퍼링을 이용한다면 다량의 데이터를 묶어서, 회당 전송하게 되는 데이터는 증가하고 패킷 수는 감소하므로 전송 속도를 향상시키는 데 도움이 되기 때문이다.

1. 표준 출력 함수를 이용해서 데이터를 전송하는 경우에는 다음과 같이 생각하는 것은 옳지 않다.

“fputs 함수 호출을 통해서 데이터를 전송하면, 함수가 호출 되자마자 데이터의 전송이 시작될 거야!”

그렇다면 위와 같은 생각이 옳지 않은 이유는 무엇이고, 위와 같이 생각하기 위해서는 추가로 어떠한 과정이 필요한지 설명해보자.

표준 입출력 함수를 이용하며, 소켓의 입출력 버퍼 뿐만 아니라 추가적으로 표준 입출력 함수의 입출력 버퍼를 얻게 된다. fputs 함수를 호출하면 전달하고자 하는 데이터가 표준 출력 함수의 출력 버퍼로 전송될 것이다.

fflush() 함수 호출을 통해 , 함수출력버퍼를 비우고, 소켓의 출력버퍼로 데이터가 전송될 수 있도록 하여야 한다.

확인 문제 유형 분석

|  |  |
| --- | --- |
| 문제 01 | 시스템 함수와 비교/대조하여, 표준 입출력 함수가 가지는 장단점 파악하기 (장점) |
| 문제 02 | 표준 입출력 함수의 특징, “fflush( )” 함수 호출과 역할 |