Chapter 12. IO 멀티 플렉싱(Multiplexing) 내용 확인 문제

1. 멀티 플렉싱 기술에 대한 일반적인 의미를 말하고, IO를 멀티플렉싱 한다는 것이 무엇을 의미하는지 설명해보자.

멀티 플렉싱이란 물리적 장치의 효율성을 높이기 위해, 최소한의 물리적 요소를 사용해서 최대한의 데이터를 전달하기 위해 사용하는 기술을 의미한다. IO 멀티플렉싱 이란 2개 이상의 클라이언트로부터 연결 요청이 있을 때, 하나의 서버로부터 여러 클라이언트에 동시에 데이터를 전달할 수 있게 한다.

1. 멀티프로세스 기반의 동시접속 서버의 단점은 무엇이며, 이를 멀티플렉싱 서버에서 어떻게 보완하는지 설명해보자.

멀티 프로세스 기반의 동시접속 서버는 클라이언트의 연결 요청이 있을 때마다, 프로세스를 생성하여 서비스를 제공한다. 프로세스는 별도의 메모리 공간을 차지 할 뿐만 아니라, 상호간에 데이터를 주고받으려면 다소 복잡한 IPC 통신 방법을 취해야 한다. 그러나 멀티플렉싱 서버의 경우, 소켓의 파일 디스크립터와 select( )함수의 사용으로 하나의 서버가 여러 클라이언트와 데이터를 주고 받을 수 있도록 해준다.

1. 멀티플렉싱 기반의 서버 구현에서는 select 함수를 사용한다. 다음 중 select 함수의 사용방법에 대해서 잘못 설명한 것을 모두 고르면?
2. Select 함수의 호출에 앞서 입출력의 관찰 대상이 되는 파일 디스크립터를 모으는 과정이 필요하다.
3. Select 함수의 호출을 통해서 한번 관찰의 대상으로 등록이 되면, 추가로 select 함수를 호출하면서 재 등록의 과정을 거칠 필요가 없다.
4. 멀티플렉싱 서버는 한 순간에 하나의 클라이언트에게만 서비스가 가능하다. 때문에 서비스를 필요로 하는 클라이언트는 서버에 접속한 후 자신의 순서가 오기를 기다려야 한다.
5. Select 기반의 멀티플렉싱 서버는 멀티프로세스 기반의 서버와 달리 하나의 프로세스만 필요로 한다. 때문에 프로세스 생성으로 인한 서버의 부담이 없다.
6. Select 함수의 관찰대상에 서버 소켓(리스닝 소켓)도 포함을 시켜야 한다. 그렇다면 어떠한 부류에 포함을 시켜야 하며, 그 부류에 포함시키는 이유도 설명해보자.

서버 소켓(리스닝 소켓)을 데이터 수신을 확인하는 fd\_set에 추가해야 한다.

리스닝 소켓의 역할을 클라이언트의 연결 요청이 들어왔을 때, 이를 받아 들이는 역할을 하며, 연결 요청 역시 클라이언트가 서버에 보내는 일종의 데이터 이므로, 리스닝 소켓에 수신한 데이터가 존재하는지 확인할 필요가 있기 때문이다.

1. Select 함수의 호출에 사용되는 자료형 fd\_set의 정의형태는 윈도우와 리눅스에서 차이를 보인다. 그렇다면 어떻게 차이가 나는지 설명하고, 차이가 날수밖에 없는 이유에 대해서도 설명해보자.

리눅스는 파일 디스크립터가 생성될 때마다 1씩 증가하지만, 윈도우는 파일 디스크립터가 0에서 시작하지도 않고, 생성되는 핸들의 정수 값 사이에 어떠한 규칙도 찾을 수 없다. 따라서 생성된 핸들의 수를 기록하기 위한 변수가 필요하다.