IT CookBook, 쉽게 배우는 데이터 통신과 컴퓨터 네트워크(개정) 11장 연습문제 해답

1. ① 동기, ② 동기점, ③ 대화
2. ① 토큰
3. ① 동기점, ② 재동기
4. Well-known 포트
5. ① 추상 문법, ② 전송 문법
6. ASN.1
7. ① 압축, ② 암호화
8. ① 압축 해제, ② 비손실 압축, ③ 손실 압축
9. ① 동시성, ② 동시성 제어

10. ①, ②, ③, ⑤

(설명④) 가장 마지막에 설정된 주동기점 이후의 부동기점 처리 과정에 대해서만 송수신자의 관점이 일치하지 않으므로 이에 대한 복구 절차를 진행한다.

11. ①, ③, ④, ⑤

(설명②) 해제 토큰은 통신 양단 간의 연결 해제 과정을 제어하기 위해 사용한다.

12. ②, ⑤

(설명②) 주동기점 간의 전송 단위는 대화로 정의되는데, 두 세션 사용자 사이에 교환되는 대화의 한 단위가 완료되는 지점에서 이루어진다.

(설명⑤) 부동기점에서의 반복 과정에도 불구하고 복구가 이루어지지 않더라도 주동기점의 경계를 넘어 되돌아가지는 못한다.

13. ①, ②, ③, ⑤

(설명④) 단일 세션 연결 방식은 초기 서비스 구축 시간이 필요하여 서비스 시간이 짧은 응용 환경에서는 사용하지 않는다.

14. ①, ②, ③, ④, ⑤

15. ①, ③, ④, ⑤
(설명②) 일반적으로 응용 환경에서 데이터를 표현하는 방법은 컴퓨터마다 다르다.

16. ①, ④, ⑤
(설명②) 연속 문자 압축 방식은 특정 문자가 연속해서 몇 번 반복되는지를 표시하면서 압축하는 방식으로 비손실 압축 방식의 간단한 예이다.
(설명③) 연속 문자 압축 방식은 원 문서에 동일한 문자가 여러번 반복되는 경우가 적으면 압축 문서가 원 문서보다 커질 수 있다.

17. ④
(설명④) 원격 파일 서비스를 제공하는 파일 서버는 비상태 서비스로 구현되는 것이 일반적이다. 따라서 시스템이 다운되었을 때, 클라이언트와 서버 사이의 복구 절차가 거의 불필요하다.

18.
세션 계층에서 제공하는 토큰의 종류는 데이터 토큰, 해제 토큰, 동기 토큰, 액티비티 토큰 네 가지다.

- 데이터 토큰(Data Token) : 데이터를 전송할 수 있는 권리를 제공한다. 따라서 세션 사용자가 데이터를 전송하려면 반드시 데이터 토큰을 먼저 획득해야 한다. 데이터 토큰을 하나만 사용하면 통신 양단 중 한쪽에서만 데이터를 전송할 수 있다. 그러므로 데이터 토큰은 응용 프로그램 간의 반이중 전송을 지원하는 목적으로 이용할 수 있다.
- 해제 토큰(Release Token) : 통신 양단 간의 연결 해제 과정을 제어하기 위해 사용한다. 임의의 사용자가 연결을 해제하려면 해제 토큰을 획득해야 한다.
- 동기 토큰(Synchronization Token) : 세션 연결을 사용하는 과정에서 동기 처리가 필요한 시점에 사용한다.
- 액티비티 토큰(Activity Token) : 액티비티는 세션 사용자들 사이에 논리적으로 설정되는 단위로, 내용이 상호 독립적이라는 특징이 있다. 예를 들어, 파일 여러 개를 전송할 때 각 파일은 하나의 액티비티로 처리되며, 각 액티비티를 처리하는 과정에서 주동기점과 부동기점이 부여될 수 있다. 액티비티 단위의 시작과 끝 표시는 주동기점의 설정과 동일한 효과를 나타낸다.

19.

• 동기점

큰 파일을 통째로 전송하는 것보다 작은 단위로 나누어 전송하는 것이 전송 오류에 쉽게 대처할 수 있다. 큰 파일 전체를 하나의 단위로 전송하면 오류가 발생하였을 때 전체 파일을 처음부터 다시 전송해야 한다. 그러나 논리적으로 작은 단위로 나누어서 전송하면 전송 오류가 발생한 부분만 재전송하면 된다.

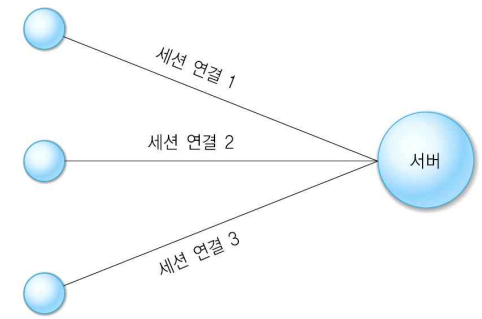
큰 파일을 작은 단위로 나누는 과정은 동기점(Synchronization Point)을 부여하는 과정이라고 볼 수 있다. 파일을 전송하는 중간 중간에 동기점을 부여함으로써, 송수신자가 해당 위치까지는 데이터 전송이 완료되었음에 합의할 수 있다. 따라서 전송 과정에서 오류가 발생해도 전체 파일을 재전송하지 않고, 가장 가까운 시점에 설정한 동기점 이후에 전송한 데이터만 오류 복구를 하면 된다.

• 재동기

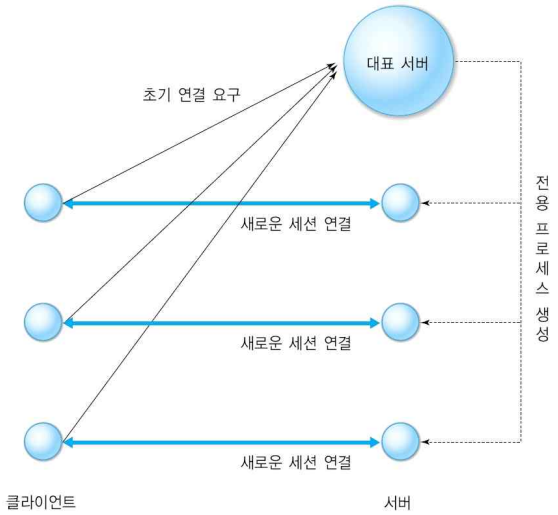
동기점은 데이터 전송 과정에서 임의의 시점에, 특정 지점에서 복구할 수 있도록 통신 양단 간의 합의로 지정된다. 세션 계층의 상위에는 적절한 구간으로 나누어진 지점에 동기점을 부여하고, 오류가 발생하면 해당 지점으로 돌아가 복구하는 기능이 구현되어야 한다. 이때 동기점을 이용한 일련의 복구 과정을 재동기(Resynchronization)라고 한다.

동기점은 주동기점과 부동기점으로 나누어지고, 주동기점 간의 전송 단위를 대화로 정의한다. 주동기점은 두 세션 사용자 사이에 교환되는 대화의 한 단위가 완료되는 지점에서 이루어진다. 부동기점은 대화 단위 내의 작은 부분에서 설정될 수 있다.

20.



[그림 11-3] 다중 세션을 지원하는 서버



[그림 11-4] 단일 세션을 지원하는 서버

다중 세션을 지원하는 서버에서는 임의의 네트워크 서비스를 제공하는 서버 프로세스가 다수의 클라이언트 프로세스에 동시에 여러 세션 연결을 설정할 수 있다. 각 클라이언트 프로세스와 설정된 세션은 논리적으로 연관이 없는 서로 독립적인 연결이다.

다중 세션을 지원하는 클라이언트-서버 환경은 서버가 제공하는 서비스 시간이 짧은 응용 환경에서 유용하다. 만일 각 클라이언트와 서버 사이의 서비스 이용 시간이 길어지면 특정 클라이언트와의 세션 연결이 길게 유지되므로 다른 클라이언트의 대기 시간이 무한정 증가되는 단점이 있다.

단일 세션을 지원하는 서버에서는 각각의 하위 서버 프로세스가 하나의 세션 연결을 사용해 클라이언트와 통신한다. 클라이언트 프로세스는 하위 서버 프로세스와 연결하기 위해 최초의 연결 설정 과정에서 대표 서버와 연결을 시도한다. 이는 기술적으로 대표 서버의 포트 주소가 Well-known 포트로 할당되므로 클라이언트에서 연결 주소를 알 수 있지만, 하위 서버의 포트 번호를Well-known 포트로 지정하기는 현실적으로 불가능하기 때문이다. 대표 서버는 하위 서버 프로세스를 새로 실행시키고, 연결을 요청한 클라이언트와 세션 연결을 시켜주는 역할을 한다.

단일 세션을 지원하는 서버 방식의 단점은 클라이언트의 개별 요구마다 하위 프로세스를 생성하기 때문에 초기 서비스 환경 구축에 따른 오버헤드가 증가한다는 점이다. 즉, 프로세스를 새로 생성하고 실행 상태로 만들어 주는데 걸리는 시간이 길어 서비스 시간이 짧은 응용 환경에서는 사용하지 않는다. 대신 각 클라이언트가 서비스를 받으려고 무한정 기다리는 현상은 발생하지 않는다.

일반적으로 텔넷, FTP 등과 같이 인터넷 환경에서 많이 사용하는 TCP/IP 서비스는 단일 세션 연결 방식을 사용하며, 대표 서버의 주소는Well-known 포트로 설정되어야 한다.

21.

일반적으로 응용 환경에서 데이터를 표현하는 방법은 컴퓨터마다 달라, 문자 하나를 표현하는 방법이 ASCII 코드를 사용하는 컴퓨터와 EBCDIC 코드를 사용하는 컴퓨터가 있을 수 있으므로, 이들이 통신하려면 문자 코드를 변환하는 과정이 필요하기 때문이다.

각 컴퓨터에서 사용하는 데이터 표현 규칙(추상 문법, Abstract Syntax)으로 표현하는 특정 의미를 올바르게 송수신하려면 메시지를 전송하기 전에 변환하는 과정이 필요하다. 즉, 특정 컴퓨터에 독립적이면서 네트워크 전체에서 일관성을 지니는 새로운 표현 규칙(전송 문법, Transfer Syntax)으로 변환하여 전송한다. 이렇게 함으로써 전송 선로를 통해 교환되는 데이터는 공통 표현 규칙인 전송 문법으로 표현된다. 반대로 네트워크에서 데이터를 수신할 때는 전송 문법의 데이터를 자신의 컴퓨터에서 이해하는 추상 문법 형태로 변환하는 과정이 필요하다

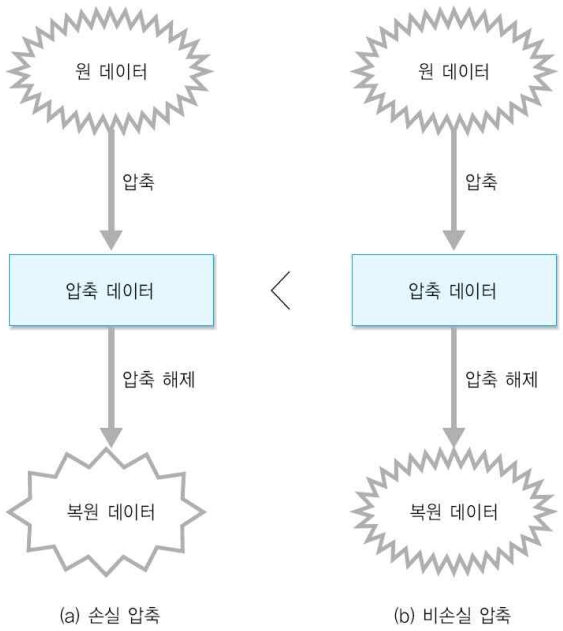
22.

교재 11장 322p-323p의 예 참고

23.

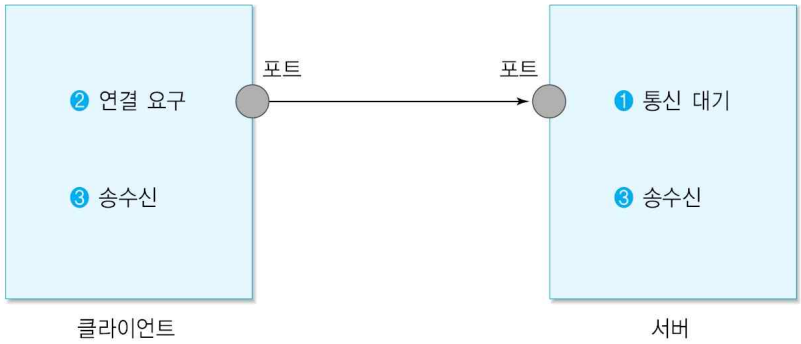
비손실 압축(Lossless Compression)은 압축 과정에서 원래 데이터의 내용을 분실하지 않는다. 즉, 압축 해제를 통해 얻은 데이터가 압축 이전의 데이터와 완전히 동일하다. 비손실 압축 알고리즘을 사용해 압축한 데이터를 해제하면 원래 데이터를 복원할 수 있다.

손실 압축(Lossy Compression)은 압축 해제한 데이터가 원래 데이터와 동일하지않다. 데이터 손실 정도는 응용 환경에 따라 허용 범위가 다를 수 있다. 예를 들어, 화상 정보나 음성 정보처럼 사람들이 감각적으로 느끼는 정보는 사용자 환경에 따라 손실 범위를 조절할 수 있다. 원래 데이터의 손실을 허용하면서 압축을 하는 이유는 압축 효율을 높이기 위함이다. 손실 압축 과정을 거친 압축 데이터는 압축 해제가 되었을 때, 원래 데이터를 완전히 복원할 수 없다.



[그림 11-9] 손실 · 비손실 압축

24.



[그림 11-10] 클라이언트와 서버의 연결

하나의 서버 프로그램이 다수의 클라이언트에 응용 서비스를 제공하는, 인터넷 응용 환경에서 가장 보편화된 연결 설정 방식이다. 클라이언트와 서버는 비교적 간단한 절차로 동작한다. 둘 사이의 연결 설정 과정에서 반드시 서버가 먼저 통신 대기 상태에 있어야 하는데, 이러한 비대칭 구조는 클라이언트와 서버가 연동되는 가장 기본적이고 단순한 방법에 대한 이론적 근거가 된다. 일반적으로 서버 프로세스는 클라이언트보다 먼저 실행되어 대기 상태에 있기 때문에 클라이언트의 연결 요청에 항상 응답할 준비가 되어 있다. 서버 프로세스는 일단 시작하면 영원히 종료되지 않고 실행되며, 다수의 클라이언트의 요청을

반복적으로 수행해준다. 클라이언트와 서버 사이의 네트워크 연결은 전송 계층의 포트 연결로 구현된다.

25.

동시성(Concurrency)은 임의의 동작이 외형상 동시에 진행되는 것처럼 보이는 것이다. 예를 들어, 일반 컴퓨터 시스템은 CPU가 하나므로 특정 시간에는 프로그램 하나만 실행할 수 있다. 그러나 운영체제에서 지원하는 시분할(Time Sharing) 방식은 여러 프로그램을 번갈아 가면서 빠르게 스위칭하여 실행하기 때문에, 사용자는 여러 프로그램이 동시에 실행된다고 착각한다.

동시성 제어는 여러 동작의 선후 진행 속도에 상관없이, 동시에 실행되어도 각 실행 결과가 항상 같은 결과를 제공한다. 즉, 독립적으로 실행되는 프로그램의 실행 순서가 결과에 영향을 주지 않는다.

클라이언트-서버 환경에서 동시성은 서버 하나가 여러 클라이언트에 동시에 서비스하는 경우를 의미하기도 한다. 즉, 임의의 클라이언트가 서버와 연결하여 서비스를 받는 동안 다른 클라이언트의 요청을 서버가 지원하면 동시성을 지원한다고 볼 수 있다.