

60200307 우연정

목차

1. Stub & Skeleton
2. Java Reflection

## 1. Stub & Skeleton

1학기 때 만들었던 수강신청 프로그램은 하나의 프로그램 내에 모든 기능이 포함되어 있었으나, 현재는 클라이언트와 서버로 분리되어 있다.

클라이언트에는 '뷰'만 있고, 나머지 모든 기능은 서버에 존재한다. 클라이언트와 서버 간의 통신은 클라이언트 소켓과 서버 소켓을 통해 이루어진다. Stub(스텝)과 Skeleton(스켈레톤)이 이 통신을 중재한다. 클라이언트와 서버는 독립적인 프로세스로 동작하며, 그러므로 각 프로세스는 다른 프로세스의 메모리 주소에 직접 접근할 수 없다.

서로 다른 문맥에서는 객체의 메모리 주소를 알 수 없으므로 'new' 연산자를 통해 객체를 생성할 수 없다. 객체를 호출하려면 IP와 포트 번호를 앞에 추가해야 한다. 서버는 여러 클라이언트에게 서비스를 제공하기 때문에 1:n 관계를 형성한다.

서버는 IP(전화번호와 유사하다.)를 통해 자신의 위치를 노출하고, 포트 번호를 통해 특정 프로세스를 지정한다. 서버는 내부 서비스의 주소와 외부에 제공하는 서비스의 이름이 다를 수 있기 때문에, 매핑 테이블을 통해 내부 주소와 외부 서비스 이름을 연결해야 한다. 이 매핑 테이블은 스켈레톤 내에 존재하며, 클라이언트 소켓은 이 테이블을 통해 서버의 객체와 함수를 찾고 호출한다.

클라이언트에서 서버로의 요청은 IP, 포트 번호, 객체 이름, 함수 이름 및 파라미터 등을 포함한 문자열로 전달된다. 이것은 사실상 사용자 정의 프로토콜이다.

웹과 클라이언트-서버 모델의 주요 차이점은 화면이 어디에 위치하는지에 따라 다르다. 웹은 서버에, 클라이언트-서버 모델은 클라이언트에 위치한다. 클라이언트-서버 프로그램(예: 카카오톡)은 기능이 더 많으며 직접 화면을 그리기 때문에 속도가 빠르다. 반면 웹은 브라우저에 의존하며, 다양한 환경에서의 접근성을 목표로 한다.

## 2. Java Reflection

리플렉션은 자바에서 객체나 클래스의 내부 정보에 접근할 수 있게 해주는 API이다. 즉, 내부 주소를 외부에서 볼 수 있도록 노출해주는 것이 자바 리플렉션이다.

자바는 내부 주소 테이블을 가지고 있어, 이를 리플렉션을 통해 외부에서 볼 수 있다. 함수 이름을 문자열로 변환하여 호출할 수 있다. 객체의 메소드에 접근할 때 테이블에 오브젝트마다 함수 주소가 있다. 이를 통해 실행 시간에 동적으로 객체의 속성을 검사하거나 변경할 수 있다. 함수 이름을 문자열로 호출하면 된다. 즉 값으로 문자열에 담아서 줘야한다. Value Object에는 포인터나 참조가 아닌 실제 값을 저장해야 한다. 주소의 사용은 금지되어 있다.

자바 리플렉션을 사용하여 파라미터를 받을 수 있다. 자바는 객체의 직렬화(Serialization)를 지원하여, 객체를 통째로 직렬화하여 전송할 수 있다. 이 기능을 통해 파라미터를 쉽게 전송

하거나 객체 조립이 편해진다.

RMI(Remote Method Invocation) 프로토콜에서도 객체의 직렬화(Serialization) 기능을 활용하여 문맥이 다른 객체 사이의 통신을 가능하게 한다.