60200307 우연정

목차

1. Inter Object Communication using Network

1-1.Stub-Skeleton

“123.45.2621:50/clogin#getUserInfo#kim#kim”

-> IP주소:포트번호/객체#함수이름#값1#값2(argument)

우리가 만들 수강신청 프로그램에서 클라이언트 쪽에는 View(PLogin)이 있고 Stub이 CLogin(컨트롤러 역할)을 상속받았다. 클라이언트 쪽 CLogin은 겉에서 보기에 서버 쪽 CLogin을 상속받아야 한다(똑같이 생겨야 한다). 즉 대리인이다.
Remote 해서 서비스를 하는 서버 오브젝트를 만들 때는 항상 서버 쪽 CLogin과 똑같이 생긴 인터페이스를 PLogin이 개발할 때 갖고 있어야 한다. 이를 위해서 인터페이스 클래스를 만들었고, 이 인터페이스를 통해서 개발할 때 함수 이름과 파라미터를 알 수 있다.

객체의 인터페이스는 함수 이름과 파라미터, 리턴 타입이 같은 것을 말한다. 어떠한 일을 시킬 때 함수 호출이 필요하다. CLogin이 일반적으로 무엇을 노출시키고 있냐면, ICLogin이라는 인터페이스를 하나 만들어서 노출하고 있다. 그러면 getUserInfo()라는 함수를 노출하고 있다. 즉 서버 쪽 CLogin을 호출하고 싶으면 ICLogin을 쓰라는 의미이다.

여기서 인터페이스는 상속받는 것이 아니라 realize이다. 해당 인터페이스의 함수를 구현한다는 뜻이다. (점선)

클라이언트 쪽 CLogin도 인터페이스를 노출하고 있다. 만약 내가 remote로 나를 쓰고 싶은 사람을 (remote service object) 인터페이스 파일을 노출하고, 이것을 써서 개발하도록 만든다. 그래야 컴파일 에러가 나지않는다.

RMI나 remote object라는 개념이 나오면 인터페이스 파일이 어딘가에 있고, 그것을 이용해서 Stub을 만든다. Stub을 상속받고 CLogin을 Stub으로 만들고, 서버 쪽 CLogin을 구현하는 것을 만들어서 대리인을 만든다. 인터페이스가 같아야 대리인 역할을 할 수 있다.

클라이언트와 서버의 인터페이스를 똑같이 맞춰야 클라이언트 쪽의 CLogin이 서버의 대리인 역할을 할 수 있다. 그렇게 되면 클라이언트 쪽 CLogin에 함수 getUserInfo()를 갖게 된다.

CLogin을 쓰는 PLogin은 쓰고 있는 것이 Stub인지 진짜 오브젝트인지 알 방법이 없다.
이렇게 인터페이스를 맞춰줌으로 인해서 CLogin의 이식성(portability)을 보장한다. 즉, 다른 오브젝트가 온다 할지라도 인터페이스가 동일한 오브젝트는 PLogin이 판단할 방법이 없기 때문에 PLogin의 코드가 변하지 않는다.

처음에 스텁이 스켈레톤을 찾아간다. 스켈레톤은 서버 소켓이다. 서버 소켓은 기다리고 있다

가 누군가가 리퀘스트를 준다. 실제로는 스텁이 아이피 주소를 쫓아서 네트워크를 찾아서 들어간다. 네트워크를 장착하면 내 네트워크 카드에 아이피 주소가 DNS서버에 등록이 되어있다. DNS서버에 등록하면 나의 서브넷이 어디에 위치하는지가 등록된다. 라우터가 찾아가서 분지시키면 내 네트워크까지 찾아온다. 찾아오면 내 프로그램이 소켓을 만들면서 포트 번호를 등록한다. 그래서 하나의 컴퓨터에 여러개의 포트가 생길 수 있다.
웹서버를 주면 포트 번호 8080이 등록된다. 고정된 포트 번호가 있다. 내가 ftp에서 파일 다운로드를 제공하고 싶으면 ftp 포트 번호가 고정된다. 8080처럼 표준화된 포트 번호도 있고, 내가 마음대로 쓰는 포트 번호도 있다.

웹에서는 "/clogin#getUserInfo#kim#kim" 이 부분에서 http 프로토콜을 쓴다. 웹에서는 함수의 이름이 많지 않다.

서버 소켓까지 포트 넘버를 찾아가면 세션(작업자)을 하나 만든다. 서버 소켓이 CLogin을 호출한다. 서버 소켓은 매핑 테이블을 가지고 있어서 어느 오브젝트가 어떤 이름을 가지고 있는지에 대한 매핑 테이블을 뒤져본다. 즉 서버 소켓이 호출할 때 쓰는 오브젝트의 이름도 클라리언트의 CLogin이 알고 있어야 한다. 이를 노출시키는 것은 서버 쪽의 CLogin이다. CLogin이 처음에 올라오면서 자신을 등록한다. 즉 실행될 때 서버 소켓의 Map에 자신의 이름을 m고 들어간다. 모든 remote 오브젝트는 서버 소켓(맵)를 알고 있다.

따라서 stub은 아이피와 포트 번호를 알고 있고 CLogin은 오브젝트 이름을 알고 있다. ARS 서비스에서 전화번호, 내선 번호, 원하는 서비스의 이름을 알고 있어야 하는 것과 마찬가지이다.

첫 번째로 CLogin이 스텁을 통해서 통신을 보내면 두 번째로 서버 소켓이 작업자를 하나 만든다. 그리고 나서 세 번째로 클라이언트 쪽 CLogin과 서버의 작업자(세션)끼리 통신을 한다. 네 번쨔로 클라이언트 쪽에서 서비스를 요구하면 작업자가 서비스를 부탁한다. 그래서 클라이언트와 작업자끼리 작업을 하다가 끝이 난다. 맵 테이블에서 받은 오브젝트 이름을 찾아서 문자열로 "getUserInfo"를 보낸다.

서버의 오브젝트가 서비스를 하기 위해서는 두 가지 맵 테이블을 모두 노출시키고 있어야 한다. 웹에서 나를 부를 때 쓰는 이름과 프로그램에서 쓸 프로그램의 구조를 가지고 있어야 한다. 인터페이스는 서버의 CLogin이 가진 인터페이스의 함수의 구조를 그대로 realize 해야 한다. 표준화된 인터페이스를 맞추는 것이다. 즉 자신의 인터페이스를 노출해놓고 클라이언트와 서버가 같은 인터페이스를 갖고 프로그램할 수 있도록(include) 만들어 놓고, 실제로 호출할 때에는 오브젝트의 이름이 노출되어 있다.(실행 시)