2002년 한국정보처리학회 춘계학술발표논문집 제9권 제1호

플랫폼에 독립적인 원격관리 시스템

김용권*
*영남대학교 컴퓨터공학과
e-mail : pbk4495@hanmail.net

A Study on The Independent-Platform-Remote-Management System

Yong-Gwon Kim
*Dept. of Computer Science, Yeung-NamUniversity

요 약

현재 대표적인 OS 인 Window,Unix,Linux 에 관계없이 클라이언트와,서버의 OS 에 상관없이 원격으로 관리할수 있는 시스템을 개발하고자하는데 그목적을 두었다 즉 클라이언트가 Linux 상에 돌아가고 서버가 Window 에서 돌아가는 시스템에서 Window 의 자원을 클라이언트가 Linux 상에서 쓰고 제어할수 있고 그반대로 Linux 의 자원을 Window 에서 쓸수있음을 제시한다.

1. 서론

현재 시중에 나와있는 시중의 원격제어나 관리시스템을 보면 특정 OS 에 한정을 받는다
예를 들어 Window 의 터미널서비스는 Window 의 계열에서만 작동되며 다른 원격제어시스템또한 Platform 에 제약을 받고있는 실정이다.

이러한 제약을 극복하고자 하여 특정 OS 에 구애받지않는 그러한 원격관리시스템을 개발하고자 하였다.

본 논문에서는 플랫폼에 독립적으로 원격제어관리 시스템을 개발하고자 선택한 언어인 자바에대해서 먼저 논하고 그리고 자바를 통한 개발단계 및 개발방법에 대해, 마지막으로 결론을 기술한다.

2. 플랫폼에 독립적인 언어 JAVA

JAVA 를 한마디로 말한다면 "Sun MicroSystem 사에서 개발한 객체 지향 프로그래밍 언어"라고 할 수 있겠다.
이 언어를 개발하게된 동기는 마이크로 웨이브 오븐과 원격 제어기 같은, 다양한 전자 장치에 삽입될 소프트웨어를 만들기 위하여 사용할 수 있는 플랫폼에 독립적인 언어가 필요한데서 비롯된 것이라 한다. 이것은 자바의 매우 중요한 장점으로서 서로다른 운영체제하에서도 동일한 코드가 사용될 수 있다는 것이다. Java 이외의 다른 프로그래밍 언어들은 각각의 운영체제에 따라서 그때마다 컴파일을 다시 해줘야 사용이 가능했었다는 것을 돌이켜 본다면 충분히 이해 할 수 있으리라 생각한다

자바는 인터프리터형 프로그래밍 언어이다.
즉, 자바는 실행 시에 코드를 읽어들여 기계어로 번역하면서 수행되는 프로그램을 만드는 언어이다.

그렇다면 C 나 C++, 파스칼 등과 같이 미리 기계어로 번역하여 변환된 바이너리를 수행하는 컴파일형 프로그래밍 언어에 비해 수행 속도가 현저하게 떨어지는 인터프리터 방식을 채택한 자바는 컴파일과 인터프리팅이 혼재하는 모델을 채택하고 있다.

컴파일을 하여 가상적인 이진 코드(좀더 정확하게 표현하여 자바에서는 바이트코드(bytecode)라고 부른다. 비트 단위가 아니라 바이트 단위의 문자로 구성되는 파일이기 때문이다.)로 변환한 다음에 그 이진 코드를 번역하면서 수행하는 방식이다.
가상적인 이진 코드는 플랫폼에 독립적인 형태로

번역이 용이하도록 설계된 자바만의 독특한 형식을 가진다. 자바는 이러한 방식을 통해 플랫폼 독립성과 수행 성능이라는 두 마리 토끼를 잡으려 하고 있다. 있을 것이다.

아키텍쳐 중립적이며 이식성이 높다.
자바는 이질적인 네트웍 환경으로 배치될 수 있는 프로그램을 지원하도록 설계되었다. 이같은 환경에서는 응용 프로그램들이 다양한 하드웨어 아키텍쳐 위에서 실행될 수 있어야만 한다. 이런 다양한 하드웨어 플랫폼 안에서는 응용 프로그램들이 다양한 운영 체제 위에서 다수의 프로그래밍 언어 인터페이스와 상호 작용하면서 실행되어야 한다.
다양한 운영 체제를 수용하기 위해 자바 컴파일러는 바이트코드(다수의 하드웨어 및 소프트웨어 플랫폼에 효율적으로 코드를 전송하기 위해 설계된 아키텍쳐 중립적인 중간 형태)를 생성한다.
자바의 인터프리팅 속성은 바이너리 배포의 문제와 버전 문제를 한꺼번에 해결한다.
즉, 동일한 자바 언어 바이트코드가 모든 플랫폼 위에서 실행되는 것이다.

아키텍쳐 중립성은 진정한 이식 가능 시스템의 단지 한 부분일 뿐이다.

자바는 기본 언어 정의를 엄격하게 함으로써 이식성에서 한 단계 더 나아간다.
기본 데이터 유형의 크기와 그 연산자의 행동을 확고하게 규정한다. 프로그램들은 모든 플랫폼에서 동일하다. 데이터 유형의 하드웨어 혹은 소프트웨어 아키텍쳐 차이에 따른 비호환성은 존재하지 않는다.

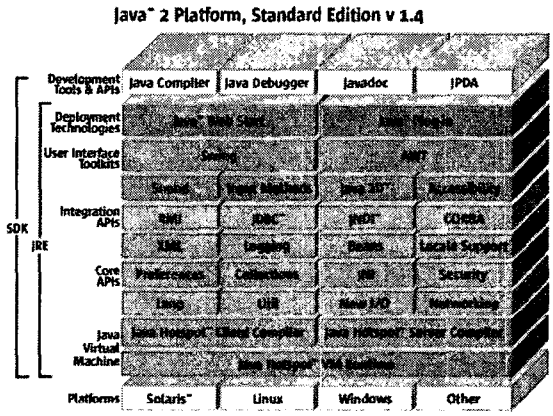
자바의 아키텍쳐 중립적이고 이식 가능한 언어 환경은 자바 가상 기계(Java Virtual Machine)로 알려져 있다.

이것은 자바 언어 컴파일러들이 코드를 생성할 대상이 되는 추상화된 기계의 명세이다.
특정 하드웨어와 소프트웨어 플랫폼을 위한 특정한 자바 가상 기계의 구현이 가상 기계의 구체적인 구현을 제공한다.

자바 가상 기계는 주로 이식 가능한 시스템 인터페이스의 산업 표준을 정의하는 POSIX 인터페이스 명세에 기반한다.
새로운 아키텍쳐에서 자바 가상 기계를 구현하는 것은 대상 플랫폼이 다중 쓰레딩(multi-threading)과 같은 기본적인 요건을 만족하는 한 상대적으로 수월한 일이다.

자바의 플랫폼적 속성
자바 프로그램이 플랫폼 혹은 아키텍쳐 독립적으로 수행될 수 있는 비결은 자바의 독특한 플랫폼적 속성에 있다.
자바는 기존의 운영 체제 위에 자바 프로그램을 수행할 수 있는 독특한 소프트웨어 플랫폼인 자바 가상 기계(Virtual Machine)를 필요로 한다.
자바의 이식(port) 작업은 본질적으로 자바 VM 의 이식.

자바는 멀티 쓰레딩을 지원하는 대부분의 현대 운영 체제로 이식되었으며 또 여러 운영 체제에서 실행되는 넷스케이프 브라우저는 자바 애플릿 실행을 위하여 각각 별도의 자바 VM 을 구현하고 있다.



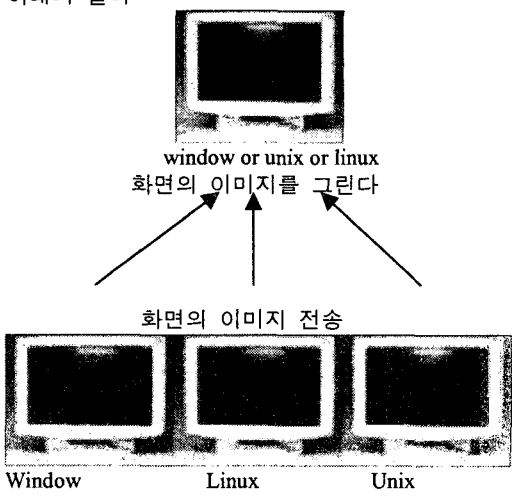
위의그림에서 보이는것처럼 플랫폼에 독립적이다는 것이다 즉 각기 다른플랫폼에 거기에 맞는 JVM 를 장착하여 프로그램의 소스 변환없이 같은결과를 가져올수 있게한것이다.

이는 자바가 탄생하면서 기본적인 모템이었다 이것은 즉 OS 에 상관없이 똑같은 소스로 같은결과를 창출할 수 있다는 것을 의미하고, OS 에 비의존적인 원격제어시스템을 만들수 있다는것을 의미하고,자바를 선택하게 된이유이다. 앞으로 OS 에 독립적인 언어로 바뀌는 추세이므로 다양한 언어로 프로그램을 만들수 있다

기본적기술은 Java 2 Standard Edtion 1.4 에 기본을 두고 특히 이미지를 읽어 들이는 기술과 이벤트를 두고 거기에 신호를 마우스나 키보드의 신호로 인식하게 하기 위해 Robot 이란 객체를 사용하였다

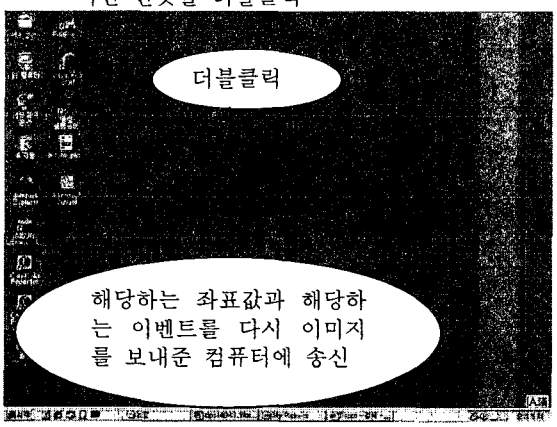
3. 원격관리리시스템 구현방법 및 단계

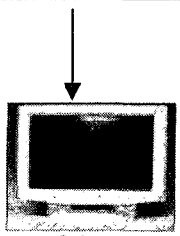
첫번째로 우선 원격제어하고자 하는 컴퓨터의 스크린 화면을 얻어온다 이이미지지를 원격지의 의 다른 컴퓨터에 전해준다 그럼 이미지를 자기의 컴퓨터에 그려준다 그것을 그림으로 설명하면 아래와 같다



둘째로 화면의 이미지를 받아온 컴퓨터는 그이미지에다 이벤트를 주어 그이미지에서 발생한 이벤트에 해당하는 값을 가지고 다시 그값을, 이미지를 보내준 컴퓨터에 다시보낸다 이를 그림으로 설명하면 다음과 같다

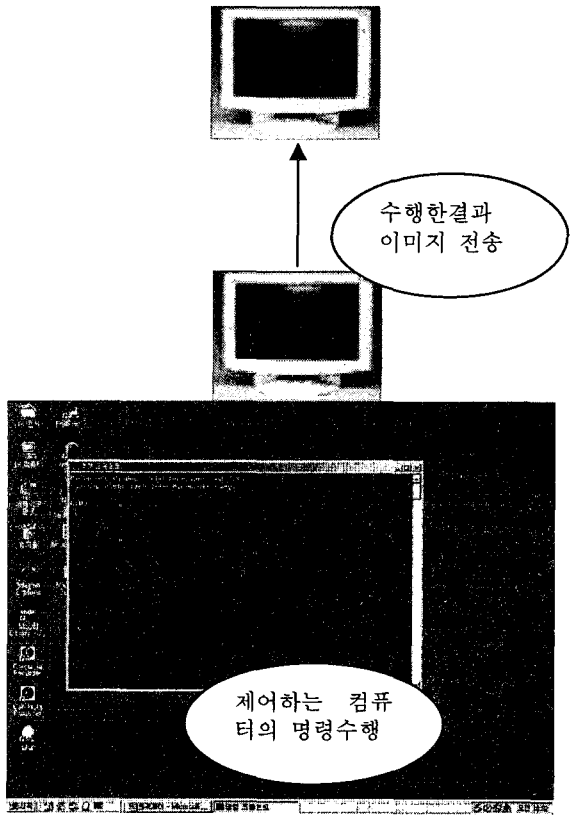
1- 1의 컴퓨터에서 다른컴퓨터에서 받은이미지에 어떤 한곳을 더블클릭





2-1

셋째 원격제어를 당하는 컴퓨터는 이명령의 메시지를 받아들여 자기컴퓨터에서 키보드의 조작이나 마우스을 핸들링 한것처럼 받아들여 거기에 맞는 명령을 수행하게 되고 그 결과를 첫단계에서와 똑같이 제어를 하는컴퓨터에 전송하게된다 이결과를 반복하면서 자바의 특성(미래에는 이런 특성을 가진 언어가 많이 출현할것이라고 예상된다)인 플랫폼에 독립적이기 때문에 서버나 클라이언트의 OS 에 제약을 받지 아니한다 이를 도식화하면 아래와 같다

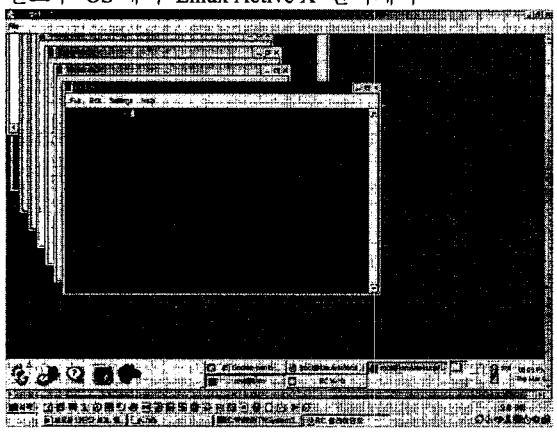


이러한 작업를 계속 반복하면서 원격지에서 플랫폼에 상관없이 원격제어 및 관리가 가능하게 되는 것이다 다음은 프로그램을 실행시킨 결과에 대해 기술하겠다.

4. 프로그램실행

원격제어를 하는 쪽에 OS 는 Window 2000 이고 원격제어를 당하는 쪽의 OS 는 RED HET Linux7.2 로 Active X 상황에서 실행하여보았다 실행결과는 다음과같다

윈도우 OS 에서 Linux Active X 원격제어



5. 결론

OS 에 제약을 받지 않고 그래픽기반으로 원격제어를 할 수 있는것을 알아보았다. 또한 이원리를 웹브라우저에 활용하면 인터넷상에서 웹브라우져가 있는 곳이라면 어디에서든지 클라이언트 프로그램없이 웹브라우져상에서 원격제어를 할수있다 이것을 업체에 활용하면 작업을 OS 에 상관없이 원격지에서 일을 할 수 있다. 즉 사업장의 OS 가 Unix 상에서 이루어지는 작업이라도 OS 에 불문하고 웹상에서 그작업을 할수있음을 의미한다.

6. 참고문헌

http://yalli.new21.org/html_lec/java2/2-5.htm
java.sun.com/j2se/1.4