

TCP/IP를 이용한 운영체제 독립 원격 제어 시스템

최인욱, 태진상, 노문진, 박성순
안양대학교 컴퓨터학과

The Operating System Independent Remote Control System By Using TCP/IP

In-Wook Choi, Jin-Sang Tae, Moon-Jin No and Sung-Soon Park
Dept. of Computer Science and Engineering, Anyang University

요 약

원격 제어 시스템이란 멀리 떨어진 컴퓨터 시스템을 자기 컴퓨터 앞에 있는 듯한 느낌과 수행 속도로 작업을 할 수 있도록 도와주는 시스템이다. 네트워크의 발달로 인하여 빠른 인터넷 환경을 접할 수 있게 되었고 전 세계가 하나의 네트워크 망으로 연결되어 어디서든지 접속이 가능하게 되었다. 이러한 인터넷 환경 속에서 원격 제어 시스템의 필요성을 느끼게 되었고 개발, 발전되었다. 또한 원격 제어 시스템을 운영 체제에 독립적인 언어인 자바로 코딩하여 어떤 운영 체제에서도 원격 제어가 가능하게 되었다. 앞으로 원격 제어 시스템은 인터넷 속도가 발전함에 따라 더욱 발전하여 컴퓨터와 컴퓨터의 원격 제어가 아닌 PDA등과 같은 휴대용 단말기에서도 원격 제어가 가능하게 될 것이다.

1. 서론

컴퓨터의 많은 보급으로 인간은 컴퓨터를 이용하여 문서 및 기타 작업을 해결하였다. 이제 컴퓨터는 인간에게 없어서는 안 될 필수품이 되었다. 이에 따라 많은 문서 작업 파일을 디스켓을 이용하여 주고받았고, 어디서든지 자기가 필요한 문서 파일을 전자 메일, 혹은 기타 네트워크를 이용하여 전송하고 작업을 하였다. 이러한 작업에 한계가 생기자 문서 작업을 멀리서도 자기 컴퓨터에 네트워크를 통해 접속하여 자기 컴퓨터 화면을 보면서 문서 작업을 하게 되는 환경을 생각하였다. 이러한 생각은 컴퓨터가 네트워크를 이용한 통신이 가능하게 되면서부터 가지게 되었다. 그래서 사람들은 이러한 생각을 구현하기 위해 초창기에는 Telnet을 이용하여 접속하였고 후에 빠른 인터넷이 보급되어 그래픽 화면까지 보아가면서 자기 컴퓨터 앞에 있는 것처럼 사용하게 되었다.

현재 대부분 컴퓨터가 Microsoft사의 Windows 라는 운영체제를 쓰고 있다. 하지만 전문 분야에서는 Linux 혹은 Solaris SPARC 등 일반인들이 잘 사용하지

않는 운영체제도 사용하고 있다. 여기서 한 가지 "서로 다른 운영체제에서도 호환이 가능한 프로그램은 없을까?" 하는 의문점이 생긴다. 이러한 문제는 운영체제에 독립적인 자바라는 언어로 해결한다. 자바로 프로그래밍을 하면 운영체제에 독립적인 원격 제어 시스템을 갖출 수 있다. 본 논문에서는 자바를 이용하여 운영체제에 독립적이고, 원격으로 편리하게 자신의 컴퓨터를 제어 할 수 있는 시스템을 제안한다. 본 논문에서 제안한 "TCP/IP를 이용한 운영체제에 독립적인 원격 제어 시스템"의 기능을 보면 크게 화면 이벤트, 마우스 이벤트, 키보드 이벤트로 나누어진다. 네트워크 연결 상태도 중요하지만 이것은 각각의 기능에 어떤 네트워크를 이용하여 접속할 것인가에 대해서 설명할 것이므로 네트워크 부분은 제외하였다. 본 논문은 2장에서 설계에 대해서 기술하고, 3장에서는 구현 설명을 마지막으로 4장에서는 결론으로 향후의 발전 가능성 및 연구 과제를 제시한다.

2. 설계

원격 제어 시스템의 기능은 화면 이벤트, 마우스 이벤트, 키보드 이벤트로 구성되고, 각각의 기능은 TCP/IP를 이용하여 통신을 한다. 본 논문에서는 제어가 될 컴퓨터를 Host PC라하고 제어를 하는 컴퓨터를 Remote PC라고 하겠다.

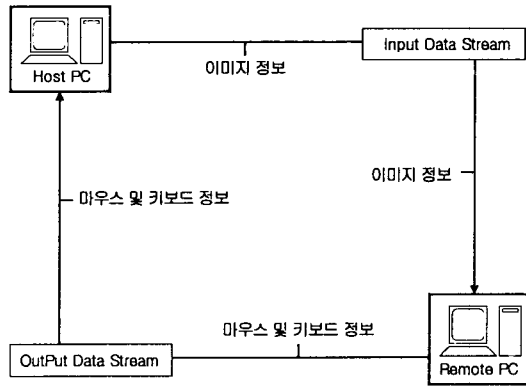


그림1 구성도

그림1과 같이 Host PC와 Remote PC 역할은 이미지 전송과 마우스 및 키보드 정보 전송이다. 이러한 간단

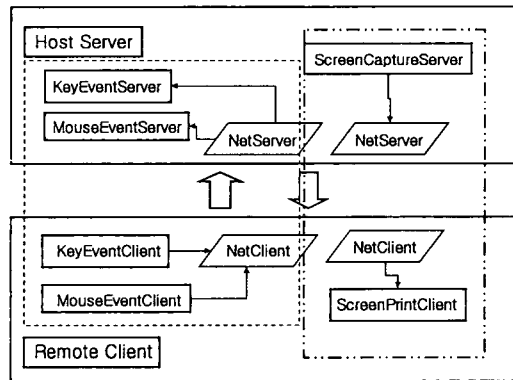


그림2 모듈 구성도

한 구조로 쉽게 원격 제어 시스템을 구성한다. 여기서 약간 복잡한 모듈 구성도는 그림2와 같다. 모듈은 크게 화면 이벤트와 마우스 및 키보드 이벤트 두 가지로 분류한다. 모듈을 두 가지로 분류한 이유는 많은 이미지를 전송하는 화면 이벤트는 TCP/IP로 마우스 및 키보드 이벤트 처리가 불가능하기 때문이다. 그래서 마우스 및 키보드 이벤트를 위한 소켓을 따로 두어 정보를 전송한다. Host PC와 Remote PC간의 동작 방식은 그림3을 통해 볼 수 있다. Host PC와

Remote PC는 접속 시 소켓 2개를 생성한다. 첫 번째

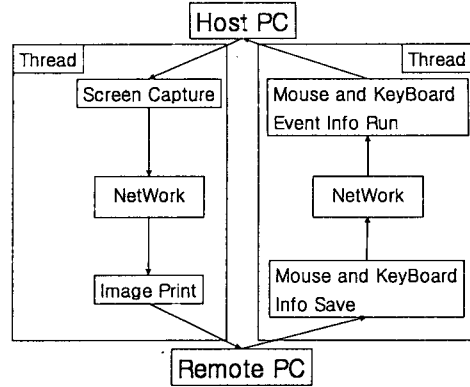


그림3 순서도

소켓은 현재 화면이 저장된 이미지를 전송하고, 두 번째 소켓은 마우스 및 키보드 이벤트가 저장된 정보를 전송한다. 다음은 원격 시스템의 가장 중추 역할을 하는 세 가지 기능인 화면 이벤트, 마우스 이벤트, 키보드 이벤트 순서로 기술한다.

2.1. 화면 이벤트

화면 이벤트란 Host PC에서 현재 화면을 이미지로 저장하고, 네트워크를 통해 Remote PC로 이미지를 전송하는 이벤트이다. 화면 이벤트는 많은 패킷 양을 차지하는 중요한 이벤트이다.

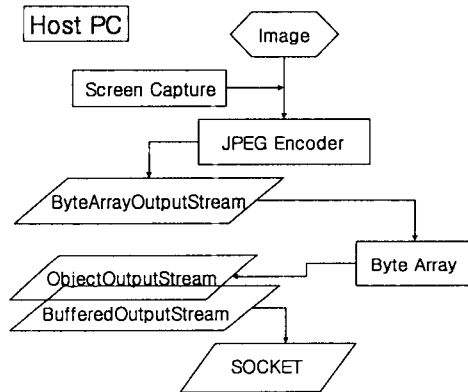


그림4 Host PC의 화면 이벤트 모듈도

Host PC는 이미지를 소켓을 통하여 전송한다. 이미지는 많은 용량을 차지하기 때문에 압축을 해야 한다. 많은 압축 기법이 있지만 많이 사용하는 JPEG 압축 기법을 이용하였다. JPEG 압축 기법을 선택한 이유는 첫째 프리웨어(Freeware)이고, 둘째 자바에서 JPEG

Encoder를 이용하면 Quality 조절이 가능하다. Low Quality로 압축한 JPEG는 압축률, 용량, 색상 면에서 많은 이점을 가진다. 그러나 단점으로는 그래픽 Quality가 좋지 않다. 참고로 Low Quality로 JPEG Encoder를 이용하여 압축하면 보통 한 이미지 당 약 50,000byte ~ 150,000byte 정도 용량을 차지한다. 소켓은 TCP/IP를 이용하여 전송하지만 내부적으로는 BufferedOutputStream과 ObjectOutputStream을 이용하여 기존의 OutputStream 보다 더 빠르게 전송한다. OutputStream은 해당 Byte를 하나씩 전송하기 때문에 반복문을 이용하여 하나씩 전달받지만 ObjectOutputStream은 Byte배열을 하나의 객체로 인식하여 객체로 전송하게 된다. 그러므로 Byte를 하나씩 보내는 것보다 더 빠른 속도로 Byte배열을 전송한다. 그림4는 Host PC 화면 이벤트의 자세한 모듈도 이다.

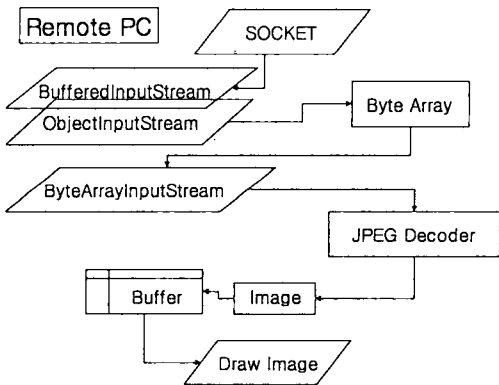


그림5 Rmote PC의 화면 이벤트 모듈도

Remote PC는 전송 받은 이미지를 화면에 출력한다. 그러나 전송 받은 이미지를 바로 화면에 출력하지 않고 이미지 5장 정도를 버퍼링하여 네트워크 속도 차에 따라서 효율적으로 이미지를 출력한다. Host PC와 Remote PC간의 원활한 전송을 위해서 버퍼링 뿐만 아니라 Host PC측에서 약 2초에 3장 정도의 이미지를 전송하고 Remote PC에서는 1초에 1장정도 화면에 출력한다. 그러므로 버퍼링과 전송 시간을 고려한다면 Host PC와 Remote PC는 약 1-3초 정도의 시간이 지연된다. Remote PC의 전송방식은 Host PC와 동일하다. 그림5는 Remote PC 화면 이벤트의 자세한 모듈도 이다. 이와 같이 많은 패킷의 양을 전송하는 화면 이벤트는 네트워크 속도와 이미지 압축률에 의하여 성능이 좌우된다.

2.2. 마우스 이벤트

마우스 이벤트를 Remote PC에서 마우스 이벤트 정보를 저장하여 Host PC로 전송한다. 마우스 이벤트 정보는 MouseMoved, MousePressed, MouseReleased 값을 의미한다. 이 정보는 int 형으로 4Byte 값을 의미하면 자세한 내용은 그림6과 같다. 예를 들어 int형

Java의 Integer은 4비트 임으로 2147483647이 최대의 크기 이므로 정보는 다음과 같이 분리된다.

KeyBoard 정보			Mouse Pressed 정보			Mouse X축 좌표			Mouse Y축 좌표		
1	2	1	1	2	4	5	4	4	5		

Mouse 버튼 정보

0 : MouseMoved

1 : MousePressed

2 : MouseReleased

Mouse의 X, Y축의 좌표는 현재 좌표를 2로 나눈 좌표 이므로 최대 해상도 2000 X 2000 까지 가능하다

그림6 마우스 이벤트 정보

4Byte로 1211245445값을 생성하였다면 121은 키보드 정보이므로 배제시키고, 1은 MousePressed 정보이고, 245는 현재 마우스 X좌표를 2로 나눈 것이고, 445는 마우스 Y 좌표를 2로 나눈 좌표이다. 해석하면 좌표 (490, 890)에서 마우스 버튼을 누른 상태이다. Host PC는 마우스 좌표를 전송 받아 2를 곱해서 정상적인 좌표로 바꾼다. 하지만 홀수 좌표는 얻을 수가 없다. 마우스 좌표 정보의 최대 지원 가능 해상도는 2000 X 2000 (2로 나누면 1000이지만 999로 한다.)이다. 단 듀얼 모니터는 지원하지 않는다. 전송 방식은 DataOut-

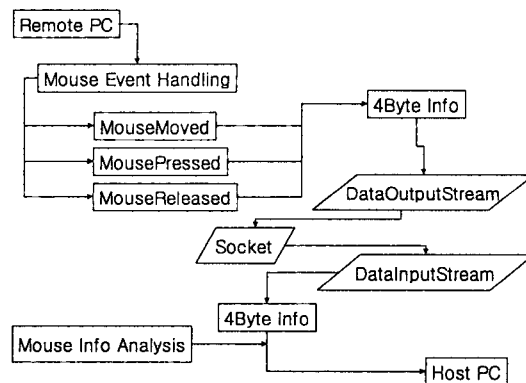


그림7 마우스 이벤트 모듈도

putStream과 DataInputStream을 이용하여 int형을 한 번에 전송하여 속도가 빠르다. 마우스 이벤트의 자세한 모듈도는 그림7과 같다.

2.3. 키보드 이벤트

키보드 이벤트란 Remote PC에서 키보드 이벤트 정보를 저장하여 Host PC로 전송한다. 키보드 정보는 마우스 정보와 함께 압축하여 DataOutputStream과 DataInputStream을 이용하여 전송한다. 키보드 정보는 int형 4Byte 앞자리 3개를(최대 217) 이용한다. 키보드 버튼은 200키 이상이 없기 때문에 현존하는 키보드의 모든 종류를 충분히 지원한다. 자바는 지원하지 않는 키보드 버튼이 있다. '윈도우 키', '한/영 전환 키', '탭 키'가 지원하지 않는 키보드 버튼이다. '탭 키'와 '윈도우 키'는 중요하지 않지만 '한/영 전환키'는 문서 작업 상 많은 비중을 차지하여 문제된다. 그렇지만 Microsoft사의 Windows는 우측하단에 한/영 전환 기능이 지원되고, Linux도 한/영 전환키(Shift +

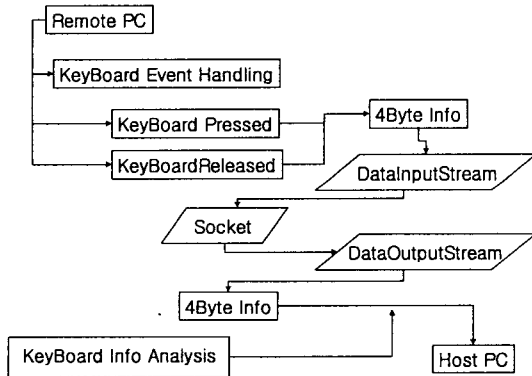


그림8 키보드 이벤트 모듈도

Space) 설정 기능이 지원된다. 키보드 이벤트는 Key-Pressed 와 KeyReleased가 있다. 키보드 이벤트 정보는 양수 값이면 KeyPressed이고, 음수 값이면 KeyReleased로 약속하여 KeyType의 정보를 생성한다. 예를 들어 KeyPressed 이벤트가 발생하면 106값이 생성되고, KeyReleased 이벤트가 발생하면 -106값이 생성된다. 키보드 정보는 전송 시 패킷 손실이 생기면 안 된다. 그래서 안전한 전송을 위해서 화면 이벤트와 소켓을 따로 두어 TCP/IP로 전송한다. 자세한 키보드 이벤트 모듈도는 그림8과 같다.

3. 구현

원격 제어 시스템은 총 3가지 프로그램으로 나누어 구현된다. 첫 번째 Host PC, 두 번째 Remote PC, 세 번째 Remote Applet으로 구분한다.

3.1. Host PC

Host PC는 제어 당하는 컴퓨터이다. Host PC의 기능은 현재 컴퓨터의 스크린 이미지를 캡처 하여 Remote PC 혹은 Remote Applet으로 전송하고, Remote PC 혹은 Remote Applet으로부터 마우스 및 키보드 이벤트 정보를 전송 받는다. Host PC 모듈도는 그림9와 같다. 이 모듈도는 앞에서 설명한 각각의 기능을 합해서 만들어진 모듈도 이다. Host PC는 스레드를 2개 생성하여 화면 이벤트와 마우스 및 키보드 이벤트를 비동기 적으로 실행한다.

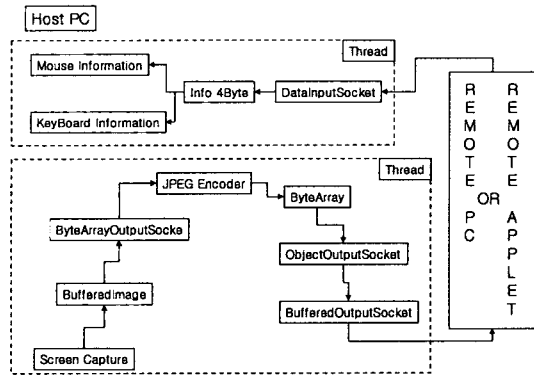


그림9 Host PC 모듈도

3.2. Remote Client & Remote Applet

Remote PC는 자체 실행 프로그램이고, Remote Applet은 인터넷에 접속하여 실행하는 Applet버전이다. 두 버전은 기능 차이도 없고, 모듈도도 같다. 그림

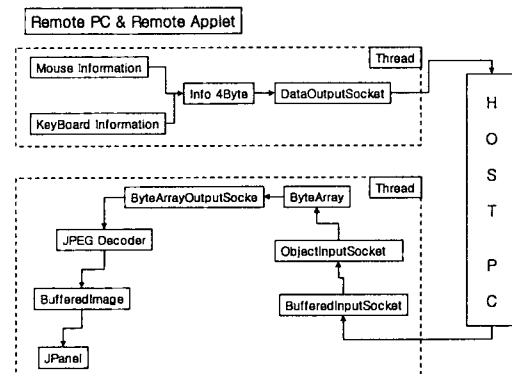


그림10 Remote PC & Remote Applet 모듈도

10은 Remote PC와 Remote Applet의 모듈도 이다. Remote Client와 Remote Applet은 자바의 Swing으로 구현했다. Swing으로 구현하면 Remote Applet은 Java Plug-in(JRE 1.3)을 시스템에 설치해야 한다. 그림11은 Java Plug-in 설치할 위한 HTML에 태그이다. 그림12는 Remote PC에서 Host PC로 접속하여 문서 작업을 하는 그림이다.

```
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
WIDTH = 170 HEIGHT =
150 codebase="http://java.sun.com/products/plugin/1.3/jinstall-13-
win32.cab#Version=1,3,0,0">
<PARAM NAME = CODE VALUE = "Clock2.class" >
<PARAM NAME = CODEBASE VALUE = "" >

<PARAM NAME="type" VALUE="application/x-java-applet;version=1.3">
<PARAM NAME="scriptable" VALUE="false">

<PARAM NAME="param1" VALUE="value1">
<PARAM NAME="param2" VALUE="value2">
<PARAM NAME="param3" VALUE="value3">

<COMMENT>
<EMBED type="application/x-java-applet;version=1.3" CODE = "Clock2.class"
CODEBASE = "" WIDTH = 170 HEIGHT = 150 scriptable=false
pluginspage="http://java.sun.com/products/plugin/1.3/plugin-
install.html"></EMBED></COMMENT>
alt="Your browser understands the <APPLET> tag but isn't running the applet.
for some reason."
Your browser is completely ignoring the <APPLET> tag!
</NOEMBED></EMBED>
</OBJECT>
```

그림11 Swing Plug-in HTML Tag

현재 원격 제어 시스템은 몇몇 운영 체제에서 자체적으로 지원 되고있다. 이는 원격 제어 시스템이 얼마나 중요한지를 알리는 지표가 된다. 현재는 컴퓨터와 컴퓨터간의 원격 제어 시스템이 구현되었지만 앞으로는 PDA 혹은 기타 휴대용 단말기, 전자 기기 등 다양한 부분에서 원격 제어 시스템을 응용한다면 신속하고 편리한 세상으로 변해 갈 것이다.

[참고문헌]

- [1] The JFC Swing Tutorial 정보 문화사 Kathy & Mary 저 / 규광 역
- [2] PROFESSIONAL Java Server Programming 정보 문화사 Danny Ayer외 14명 저 / 하수정 역
- [3] Beginning Java2 JDK 1.3 Edition 정보 문화사 Ivor Horton 저 / 추형준, 고규철, 정용욱 공역
- [4] 프로그래머를 위한 Java2 홍릉과학출판사 최재영, 최종명, 유재우 공저
- [5] Java Programming Bible Ver.2 (주) 영진 출판사 이현우, 김형국, 홍성남 공저

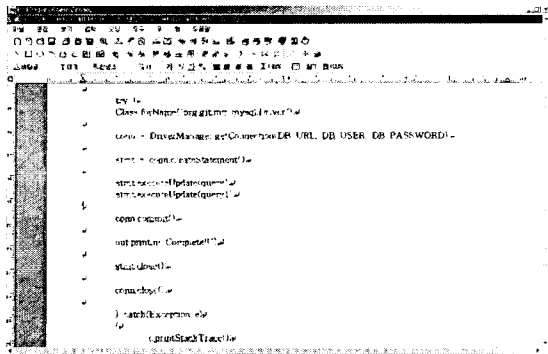


그림12 Host PC 스크린 샷

4. 향후 연구 방향 및 결론

원격 제어 시스템의 가장 중요한 것은 이미지 압축 방식이다. 본 논문에서는 JPEG 압축 방식을 이용하였지만 더 좋은 압축 방법으로는 이미지 압축 시 시간축을 이용하여 압축하거나, 바뀐 부분의 이미지만 전송하여 수정하는 방법 등이 있다. 압축률 및 그래픽 Quality가 좋은 이미지 압축 방법을 이용하면 프로그램 성능이 많이 개선되고 향상된다. 그렇게 되면 일반 문서 작업뿐만 아니라 그래픽 작업, 멀티미디어 작업 등 컴퓨터 작업 영역이 많이 확대된다.