

네트워크 환경에서의 소프트웨어 기반 키보드/마우스 공유 시스템의 설계 및 구현

이진수^o, 정선일, 이명오, 이상호, 노영주
한국산업기술대학교 컴퓨터공학과

{jsslee, airseonil, hiraga, sangho, yrho}@kpu.ac.kr

Design and Implementation of a Keyboard/Mouse Sharing System Based on Software in Network Environment

Jin-Soo Lee^o, Sun-Il Jung, Myung-Oh Lee, Sang-Ho Lee, Young-ju Roh

Dept. of Computer Engineering, Korea Polytechnic University

요 약

개인용 컴퓨터의 급속한 보급으로 인해, 1인 1PC 시대에서 한사람이 2대 이상의 PC를 사용하는 시대로 변화하고 있다. 요즘에 많은 컴퓨터 사용자들은 여러 대의 컴퓨터를 사용하여 업무를 수행하고 있다. 본 논문에서는 이와 같은 사용자에 편리한 작업 환경을 제공하고, 이를 통해 생산성 증가시키기 위해서 네트워크로 연결된 2대 이상의 PC 입력 컨트롤러(키보드, 마우스)를 통합하는 시스템을 설계 및 구현하였다. 본 논문에서 개발할 시스템은 기존 소프트웨어 혹은 KVM에서 제공하지 못하는 두 시스템간의 Clip board 및 파일의 공유 기능을 제공하여 사용자로 하여금 1대의 PC를 사용하는 것과 같은 작업 환경을 제공한다.

키워드: Keyboard/Mouse Sharing System, KVM based on software

1. 서 론

노트북 사용자가 급증 하면서 2대 이상의 PC를 사용하는 작업환경의 사용자가 늘어나고 있다. 2대 이상의 PC를 사용함에 따르는 불편함은 통합되지 않은 입력 컨트롤러(마우스, 키보드)를 사용하는 것이다. 이 불편함을 해소하기 위해서는 다수대의 PC 환경에서 각 PC의 입력 컨트롤러를 통합하는 것이 요구되었다.

본 논문은 기존의 KVM과 같이 하드웨어적으로 이와 같은 기능을 제공하는 것이 아니라, 소프트웨어로 이를 설계 및 구현한다. 이는 기존 KVM에서 구현하지 못한 파일의 공유 및 Windows의 Clip board 공유 기능을 제공할 수 있을 뿐 아니라, 추가 장비가 필요 없어 비용 절감 효과와 확장성이 뛰어나다.

본 논문의 구성은 먼저 제 2장에서 기존 소프트웨어 기반 KVM 시스템에 관해서 살펴보고, 제 3장에서는 이와 같은 입력 컨트롤러 통합 시스템에 요구되는 기능에 관해서 논한다. 제 4장에서는 본 연구에서 설계 및 구현하는 시스템에 관해서 논하고, 마지막으로 제 5장에서 결론을 맺는다.

2. 관련연구

본 장에서는 소프트웨어 기반 KVM 시스템 중에서 대표적인 시스템인 MaxiVista와 Jazz Multi PC에 관해서 논한다.

2.1 MaxiVista

MaxiVista[1]는 Virtual VGA 드라이브 기술을 통해 윈도우상의 화면 해상도를 확장하고 확장된 영역을 네트워크를 통해 다른 컴퓨터에 전송하여 Multi-Monitor 기능을 제공한다. 또한 이 시스템은 다른 컴퓨터로의 마우스/키보드 전환 기능을 제공하여 이 기능을 통해서 메인PC의 마우스나 키보드로 2개의 PC를 제어할 수 있도록 한다.

이 시스템의 장점은 마우스 제어의 전환 시에 지연이 거의 없고, 커널 레벨에서의 시그널 전송으로 인하여 빠른 반응속도를 제공한다는 것이다. 그러나 이 시스템은 Windows의 Clip board 공유 기능만 제공하고 파일 공유 기능은 제공하지 않는다는 단점이 있다.

2.2 Jazz Multi PC

국내에서 프리웨어로 개발된 이 소프트웨어는 마우스/키보드의 통합을 목적으로 개발되었다. Server와 Client 시스템으로 나누어져 있으며 각각을 네트워크로 연결되어있는 2대의 PC에 설치하면 두 PC간 마우스/키보드 통합 환경이 구축된다. 이 시스템은 마우스/키보드 제어의 전환 시에 지연시간이 길다는 단점과 Clip board 공유와 파일 공유 기능을 제공하지 않는다는 단점이 있다.

3. 입력 컨트롤러 통합 시스템의 요구사항

입력 컨트롤러의 소프트웨어적인 통합을 위해서는 각 입

력 컨트롤러의 신호를 완벽하게 읽어와 제 2PC로 전달해야 한다. 이를 위해서는 먼저 제 2PC의 해상도와 제 1PC의 해상도가 다를 수 있으며, 이것에 대한 보완책이 있어야 한다. 다음으로 입력 제어가 넘어간 제 2PC의 마우스 커서가 하는 일을 제 1PC에서 감지할 수 있어야 한다. 파일을 선택했는가?, 파일의 복사를 시도하는가?, 키보드 이벤트가 발생하는가?, 혹은 마우스가 제 2PC에서 1PC로 다시 넘어 오려고 하는가? 등을 실시간으로 감지하고 이에 대응할 수 있는 준비가 되어있어야 한다. 그림1과 그림2는 PC들이 물리적으로 연결된 입력 컨트롤러의 모습과 제 2PC로의 제어신호의 방향 전환을 보여준다. 이를 위해 컨트롤러 제어 전송 외에 별도의 프로토콜을 두어 두 PC간 대화가 이루어 져야 한다.

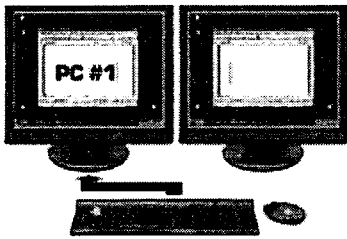


그림2. 제 1PC가 마우스/키보드를 사용하는 경우.

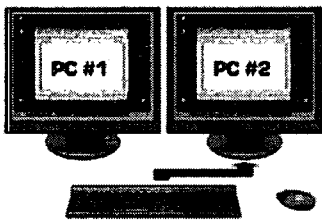


그림2. 제 2PC로의 마우스/키보드 제어가 전환된 경우.

Server는 파일의 전송을 담당하며 시스템에서 2PC간의 파일 공유를 담당한다. Secondary System은 Primary System 동작방식의 역순으로 받은 시그널을 2PC에 적용시킨다. 본 장에서는 각 기능별로 상세 설계에 관해서 논한다.

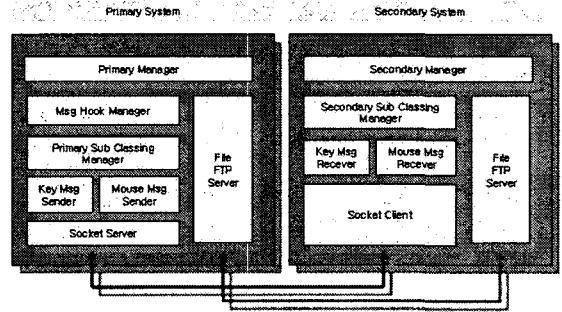


그림3. S/W 구조.

4.1 Global Hooking 과 Hook data와 App 간의 메모리 공유

Global Hooking은 WH_MOUSE_LL의 Hook Type을 사용하며 전역적인 동작을 위해 동적 라이브러리(Dynamic Linking Library: DLL)로 구현하였다. WH_MOUSE_LL은 저수준의 마우스 메시지를 가로채어 마이너스 좌표 값을 얻을 수 있다. 마우스의 X 좌표가 마이너스로 감지되면 Window 프로시저로의 마우스 메시지 전달을 중단하고 마우스 cursor를 숨긴 후 hook된 입력컨트롤러의 제어 데이터를 네트워크를 통하여 제 2PC로 전송한다. 이때 데이터의 소켓 전송은 DLL이 아닌 APP에서 이루어져야 한다. 이는 Global Hook DLL에서의 소켓 사용은 시스템의 치명적인 성능 저하를 가져올 수 있기 때문이다. 이를 해결하기 위해 그림4와 같이 DLL과 APP 간의 공유 메모리 영역을 통하여 메시지 데이터를 전달하여 빠른 전송 속도를 얻을 수 있었다. [2],[5]

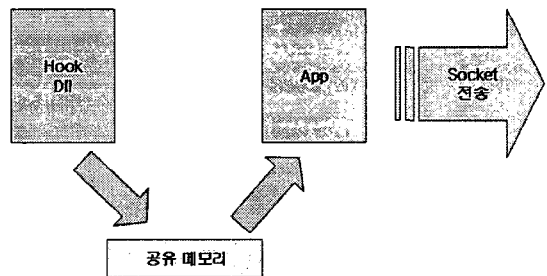


그림4. DLL과 APP 간의 메모리 공유.

4. 시스템의 설계

시스템의 기본기능은 크게 마우스/키보드의 Global Hooking, Hook data와 App 간의 메모리 공유를 통한 실시간 데이터 처리, TCP/IP를 통한 시그널 전송, 텍스트 및 파일의 Clip board, FTP를 통한 파일공유이다.

본 논문에서 구현할 시스템의 구성도는 그림 3과 같다. 그림 3.에서 Primary System은 MsgHookManager와 SubClassingManager의 단계를 거쳐 로컬 PC의 입력 컨트롤러 제어 신호를 얻는다. 이를 KeyMsgSender와 MouseMsgSender의 단계를 거쳐 각각의 메시지로 분리하여 Socket Server를 통하여 2PC로 전송한다. File FTP

4.2 TCP IP를 통한 제어 시그널 전송

제 2PC로의 마우스/키보드 메시지 전송은 TCP IP 비동기 소켓으로 이루어진다. 제 2PC로의 시그널 전송 시점은 1PC의 마우스 x좌표가 마이너스 좌표가 되거나 또는 단축키 등을 통한 2PC로의 제어 전환이 이루어졌을 시점이다. 이때 데이터를 제 2PC로 넘기는 시점에 제어 전송 지연을 줄이고 2PC의 반응속도를 높이기 위해 1PC 쪽에서는 Hook 된 입력컨트롤러의 제어 데이터의 연산 및 추출 작업을 하지 않으며 바로 소켓전송을 수행한다. 전송되는 구조체에는 입력컨트롤러의 시그널 외에 별도의 플래그(flag)들이 포함된다. 이들은 마우스가 위치한 PC 정보와 파일 전송을 위한 정보들을 표현하고 있으며, 이를 통하여 키보드 메시지의 전달여부, 파일의 전송여부 및 클립보드 데이터 전송 등을 결정한다.[3][4][6]

4.3 텍스트 및 파일의 Clip board

네트워크를 통한 텍스트 및 파일의 Clip board 공유를 지원하기 위하여, 시스템은 [Ctrl + V] 키의 입력이 있을 경우 먼저 Local System의 Clip board를 검사하고 정보가 없을 경우 1PC (마우스제어가 2PC 에 있을 경우)의 Clip board를 검사하여 해당 데이터를 2PC 로 전송한다. 이때 데이터가 TEXT일 경우에는 TCP IP 소켓을 이용하고, 파일일 경우에는 FTP를 이용하여 데이터를 전송한다. 파일 전송에 FTP를 이용하는 이유는 복수개의 파일 혹은 디렉터리의 전송 작업을 원활히 수행하기 위함이며 FTP는 보안상 전송 작업이 필요할 경우에만 open 되며 해당 작업이 없을 경우 close 상태를 유지한다. 아래의 그림5는 1PC 와 2PC 간의 시그널 및 파일의 전송 모습을 보여준다.

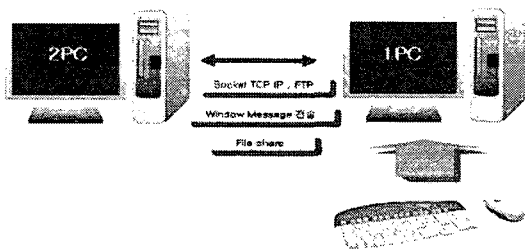


그림5. 1PC 와 2PC 간의 시그널 및 파일 전송 모습.

4.4 FTP를 통한 파일공유

시스템은 두 PC간의 파일을 사용자가 마치 한 개의 PC를 사용하는 것처럼 느낄 수 있도록 지원한다. 이를 위해 파일의 Drag & Drop 을 지원하여 1PC에서 2PC 혹은 2PC에서 1PC로의 데이터 전송이 가능하게 하며, 내

부적으로는 FTP를 통한 파일의 전송이 이루어진다. 이 기능은 Clip board의 확장된 기능이며 보내는 쪽의 선택된 파일정보를 저장 후 전송이 필요한 시점에 FTP를 통하여 처리된다. 또한 바탕화면, 탐색기등의 인스턴스 정보를 이용하여 파일의 마우스 Drop이 이루어지는 위치에 따라 해당 파일이 저장될 수신 Path를 결정 한다.

5. 결론 및 향후연구

본 논문에서는 2대 이상의 PC를 사용하는 작업환경을 통합하는 새로운 솔루션을 제시하였다. 본 솔루션은 Network상의 Windows OS 간의 입력컨트롤러 통합하는 방법을 사용하였다.

본 연구개발에서 제시한 솔루션은 원활한 입력 제어를 위하여 Global Hooking과 다른 프로세스간의 메모리 공유 기법을 사용하였다. 사용자로 하여금 1대의 PC를 사용하는 것처럼 느끼도록 파일의 공유기능이 설계 되었으며 데이터와 파일 각각에 적합한 프로토콜을 사용하여 시스템의 부담을 줄였다.

특히 PC간 마우스 Drag & Drop을 통한 파일의 공유는 기존 솔루션에 없는 본 연구의 핵심이 되는 기능으로써 향후 이기종의 OS간의 통합이 이루어질 경우 아주 높은 생산성을 줄 수 있는 기능이다.

향후 Linux와 Windows 혹은 Mac OS 와 Windows 간의 입력컨트롤러 통합 및 파일의 공유에 관한 연구개발이 진행되면 이기종 환경에서의 작업 생산성 또한 크게 향상될 것으로 기대하고 있다.

6. 참고문헌

- [1] <http://www.maxivista.com/kvm.htm>
- [2] Pat Villani, Programming Win32 Under the API
- [3] Jon C. Snader, Effective TCP/IP Programming
- [4] Anthony Jones, Network Programming for Microsoft Windows, Second Edition
- [5] 이종우(JongWoo Lee), 윤종완(JongWan Yoon), 김문희(MoonHee Kim)
NT - SPLASH : Parallel Benchmark Programs using Windows - NT Win32 API
한국정보과학회 학술발표논문집 1998년도 봄(A)
(1998년 4월)
- [6] MS - Windows용 TCP / IP 다이내믹 링킹 라이브러리의 설계 및 구현
고대건(Daegun Goh), 이영종(Youngjong Lee), 신운숙(Yunsuk Shin), 김영호(Youngho Kim)
한국정보과학회 학술발표논문집 1995년도 봄
(1995년 4월)