
Embedded Linux 기반의 내장형 프린터 서버

차경환*

An Onboard Printer Server Based on Embedded Linux

Kyung-Hwan Cha*

요 약

기존 사무 환경에서 개인 PC를 이용하여 네트워크를 통해 프린터를 공유하는 경우 프린터와 직접 연결된 PC에 상당한 과부하가 발생한다. 프린터에 직접 연결된 PC 사용자는 개인 작업 시 이러한 과부하로 인해 불편함이 발생하게 된다. 본 논문에서는 이와 같은 문제점을 해결하고, 사무실 구성원들이 네트워크를 통해 원활한 프린터 작업을 수행할 수 있도록 프린터 서버 역할을 수행하는 Embedded Linux 기반의 Embedded Printer System을 구현 하였으며, 실제 환경에서 실험을 거쳐 성능을 검증하였다.

ABSTRACT

PC users could experience some inconvenience when they use their PC that was physically connected to network printers shared by other people in the network due to the overload.

This paper shows how to resolve this kind of overload problem and established the embedded printer system based on embedded Linux system which performs the printer server in order for any users in the office to do printing work more smoothly via current network. It's own capabilities and functions were verified throughout the various tests in practical environment.

키워드

Printer Server, Embedded Linux, Linux Device Driver, GUI, Qtopia.

I. 서 론

정보통신기술의 발전으로 인하여 인터넷 서비스의 다양화와 네트워크 기술의 향상은 우리에게 많은 편리를 제공하여 준다. 기존에 독립적인 기능을 가지고 동작 하던 가정 및 사무용기기는 인터넷과 결합하여 그 기능을 확장하게 되었다. 가정에서는 홈 네트워크 시스템으로 TV, 냉장고, 에어컨 등 모든 가전기기들이 하나의 네트워크로 연결되어 사용자에게 정보를 전달할 수 있게 되었다.

그리고 사무 환경에서는 사무실의 PC가 하나의 소규모 네트워크로 연결되어져서 프린터와 같은 정보기기들은 네트워크를 통해 공유하여 사무실 구성원 누구나 자신의 PC에 연결된 것처럼 프린터를 사용할 수 있게 되었다.

사무공간에서 네트워크를 이용하여 프린터를 공유하는 방법은 여러 가지의 경우로 나눌 수 있다. 첫 번째는 사무인원이 적은 소규모 사무실 환경에서 사용하는 방법으로 프린터를 사무실 내에 네트워크와 연결된 PC 중 하나에 직접 연결하고 다른 사용자가 사무실 네트워

크를 이용하여 프린터 작업을 하는 경우이다. 두 번째는 프린터 자체에 직접 네트워크 기능을 가지는 프린터 서버를 두고 사무실 구성원들이 네트워크를 이용하여 프린터 작업을 하는 경우이다.

인원이 적은 소규모 사무실 환경에서는 프린터 작업을 위해 별도의 프린터 서버를 두는 경우에 비용과 사무실 운용측면에서 비효율적이다. 더욱이 개인 PC를 이용하는 경우에는 다른 인원의 프린터 작업으로 인하여 발생하는 부하로 인하여 프린터와 직접 연결된 개인 PC 사용자가 불편을 겪게 되는 등의 문제점이 발생하게 된다 [1-3].

본 논문에서는 이와 같은 문제점을 해결하기 위해서 사무실 구성원이 네트워크를 통해 프린터 작업을 수행하기 위해 프린터 서버 역할을 수행하는 Embedded Linux 기반의 프린터 시스템을 구현하였으며, 실제 환경 하에서 실험을 거쳐 성능 및 기능에 대하여 검증하였다.

II. 기존의 프린팅 작업시 발생하는 문제점

2.1. 프린터 작업시 PC에 발생하는 부하

일반적으로 많이 사용하는 운영체제인 Windows 또는 Linux와 같은 OS(Operating System)의 경우는 다중 프로세스를 지원하는 운영체제이다. 그리고 일반적인 PC와 같이 CPU가 하나인 시스템일 경우 프로세서가 하나만 존재하기 때문에 각 프로세스들이 하나의 CPU를 짧은 시간동안 차지하여 수행되기 때문에 여러 프로세스들이 동시에 수행되는 것처럼 보여 지게 된다. 이를 시분할 방식이라고 한다. 그리고 대부분의 OS들이 시분할 방식을 사용하기 때문에 하나의 프로세스가 오랫동안 CPU를 차지하고 있을 때 다른 프로세스가 수행되지 못하므로 프로그램은 멈춰진 것처럼 보여지게 된다.

아래 그림 1은 Windows상에서 평상시 CPU 사용률과 프린터 작업 시 CPU사용률을 나타낸 그림이다. 그림 1에서 보듯이 작업대기시에는 최소 10%에서 최대 40%까지 CPU를 사용하는데 반해 대용량의 프린터 작업 시는 순간적으로 CPU 사용률이 100%까지 올라가게 되므로 프린터와 직접 연결된 사용자는 네트워크에 연결된 다른 사용자의 프린터 작업을 할 경우 아래와 같은 현상이 발생하게 되어 작업이 지연되는 등의 불편함을 겪게 된다.

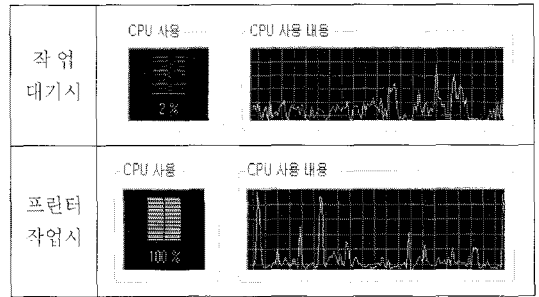


그림 1. CPU 사용률
Fig. 1. Using Rate of CPU

2.2 서로 다른 OS를 사용하는 경우

운영체제는 컴퓨터 하드웨어와 사용자 응용프로그램 사이에 위치하여 응용 프로그램이 하드웨어를 쉽게 사용할 수 있도록 해주고 전체적인 시스템의 효율을 극대화시키기 위해 하드웨어 및 소프트웨어 자원을 관리하는 프로그램들의 집합이다. 그리고 대부분의 PC에서는 사용자의 편의를 위해서 Windows, Linux, MacOS 등의 범용 운영체제를 사용하고 있다. 이러한 각각의 운영체제에는 정보들을 저장하는 파일들을 다루는 규칙과 하드웨어에 데이터를 읽거나 쓰며 데이터를 찾는 방법인 파일시스템을 가지고 있다.

하지만 파일시스템은 운영체제마다 독립적이며, 서로 다르기 때문에 다른 기종의 운영체제에서는 서로 파일을 공유하거나 동일한 하드웨어 자원을 네트워크로 공유하여 사용하는 것은 특별한 프로그램이나 Solution을 사용하지 않는 이상은 어렵다. 그러므로 사무실 환경 내에서 프린터와 직접 연결된 PC사용자와 네트워크에 연결된 다른 사용자간에 서로 다른 운영체제를 사용하고 있을 경우, 서로 같은 운영체제 환경에서 프린터 작업을 해야 하므로 프린터와 직접 연결된 PC사용자는 작업을 잠시 중단할 수밖에 없는 불편함을 가지고 있다.

III. 시스템 설계 및 구현

본 논문에서 구현한 시스템은 그림 2와 같이 프린터와 직접 연결되며, 사무실 내의 모든 PC와 사무실내의 네트워크를 통하여 시스템과 연결되어 있다.

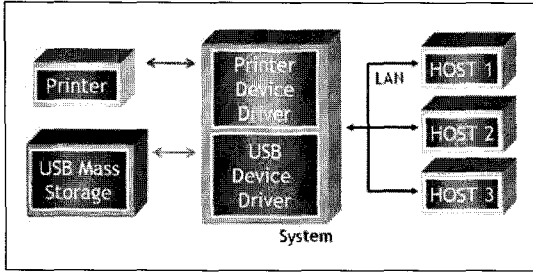


그림 2. 시스템 동작 환경도
Fig. 2. Configuration of System Operating

3.1 시스템 구성

본 시스템은 프린터가 PC에 연결되지 않고 시스템에 연결되어 독립적인 기능을 수행하기 위한 Embedded System이며, 네트워크상의 모든 PC가 서로 다른 운영체제를 사용하여도 시스템을 통하여 원활 한 프린터 작업을 가능하게 하였다. 그리고 사용자의 편의를 위하여 시스템에 터치 TFT-LCD와 USB 인터페이스를 탑재하여 USB 저장장치 내의 문서도 시스템 상에서 직접 프린터 작업을 가능하게 하고자 했다.

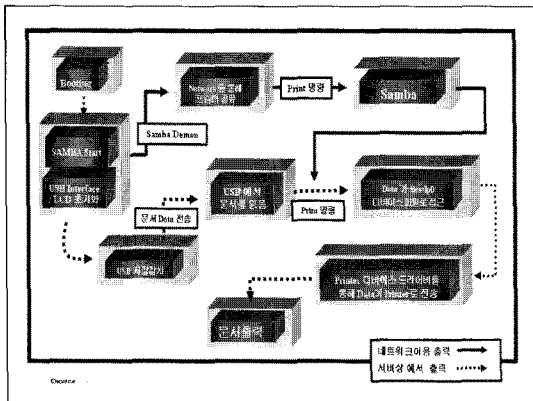


그림 3. 시스템 블록도
Fig. 3. Block diagram of Embedded Network System

시스템 구동을 위한 운영체제로는 Embedded Linux를 사용하였으며, 네트워크를 통해 프린터를 공유하기 위해서 SAMBA를 포팅하고, GUI는 PDA등에 사용되는 Linux GUI인 Qtopia를 포팅 하여 PC상의 프린터 작업 외에도 시스템 상에서 원활 한 프린터작업과 효율적인 시스템 관리를 할 수 있도록 하였다.

3.2 하드웨어 구현

구현하려는 시스템은 Ethernet과 USB Interface, 그리고 GUI 사용을 위한 터치패드와 LCD Interface가 구현 되어야 한다. 또한 시스템은 Embedded Linux를 사용하여 구현할 것이므로, Embedded Linux를 포팅 할 수 있어야 한다. 이런 조건을 감안하여, Ethernet과 시리얼 통신환경이 구축되어 있으며, GUI를 위하여 터치패드와 LCD 인터페이스, 그리고 USB포트가 구현되어 있는 개발보드 중 400Mhz로 동작하는 ARM10계열의 Intel-PXA255가 탑재되어 있고, 64Mbyte의 SDRAM과 ROM을 지원하는 개발보드를 사용하게 되었으며, 프린터 작업을 하기 위한 하드웨어는 기본적인 TTL Logic으로 구현 하였다. 그림 4는 개발보드의 메모리 맵을 나타내었다.

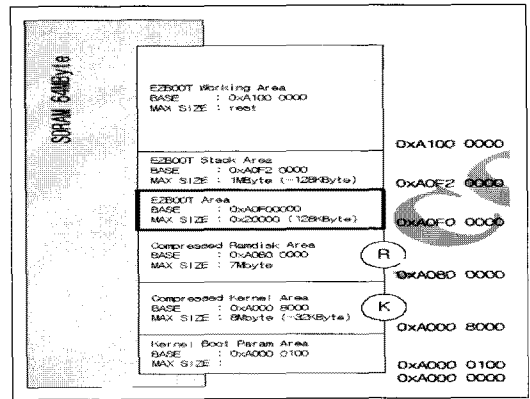


그림 4. 전체 시스템 메모리 맵
Fig. 4. Memory Map of Total System

3.2.1 프린터 디바이스 드라이버 구현

프린터 디바이스 드라이버를 제어하기위한 디바이스 파일에는 lp0, lp1 등이 있다. 하지만 실제 병렬포트를 지원하는 것은 linux/drivers/parport 디바이스 파일로 병렬포트의 하드웨어적인 인터페이스를 정의하는 곳이다. 프린터 디바이스 드라이버의 동작은 그림 5와 같다.

커널이 부팅을 시작하면 프린터 디바이스 드라이버가 커널 내에 등록이 되어지고(Irq_desc[irq]), 프린터 작업을 위해 프린터 디바이스 드라이버가 호출되어지면(인터럽트 서비스 루틴을 호출하면), 인터럽트 서비스 루틴에 의해 프린터가 실행되어진다. 그리고 프린터의 데이터 전송 모드는 SPP, EPP, ECP로 나누어진다. SPP 방식은 1970년대에 개발된 단방향 전송방식이며 8비트

데이터 전송을 지원한다. ECP는 양방향 고속 전송을 위한 규약으로 12bit로 데이터를 전송하는 방식이다. 그리고 EPP방식은 인텔에서 만든 규약으로 양방향 병렬 포트를 지원한다. 고속의 ECP, EPP방식을 사용하기 위해서는 프린터가 ECP, EPP 방식을 지원해야 한다. 따라서 시스템과 프린터간의 호환 문제가 발생하기 때문에 본 논문에서는 SPP 방식을 이용하여 프린팅 작업을 진행하였다.

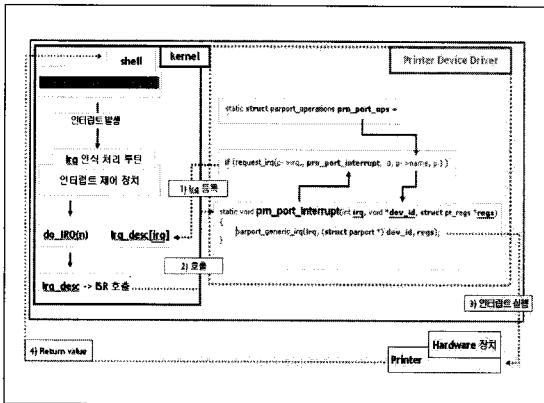


그림 5. 프린터 디바이스 드라이버
Fig. 5. Printer Device Driver

3.2.2 SAMBA 포팅 및 세팅

SAMBA는 서로 다른 운영체제 간에 프린팅 서비스를 가능하게 하기 위해 본 시스템에 SAMBA Server를 포팅 시키고 프린팅기능 외에 다른 기능을 사용하지 못하게 세팅했다. 그 이유는 SAMBA가 동작하는 시스템이 네트워크에 연결되어 있으면 다른 시스템에서 특정 디렉토리를 PC의 가상 하드디스크로 연결, 마치 자신의 PC에 연결되어 있는 장치처럼 사용할 수 있기 때문이다. 이런 모든 서비스를 특정 주소에서부터 연결할 수 있도록 세팅 가능하며 SAMBA Server의 각 디렉토리 및 파일의 권한을 파일의 Owner 및 Group이 Read, Write, Execute 등으로 설정을 각각 할 수 있다. 그리고 SAMBA 내부에 SWAT가 있어 웹상에서 SAMBA 설정 및 접속이 가능하다.

그림 6은 SWAT를 통하여 웹상에서 접속하는 과정을 보여주고 있다. 그림 7은 웹상에서 SAMBA를 실행하여 프린트 드라이브를 등록하는 과정을 보여주고 있다.

3.2.3 GUI/USB 인터페이스

대형 시스템에 내장되는 Embedded System은 콘솔을 사용하면 되지만, 본 논문에서 제안한 시스템은 소형의 시스템은 LCD와 터치패드가 결합된 터치스크린 위에서 아이콘과 가상 키보드를 이용하여 사용자가 시스템과 대화하는 편리한 GUI가 사용된다.

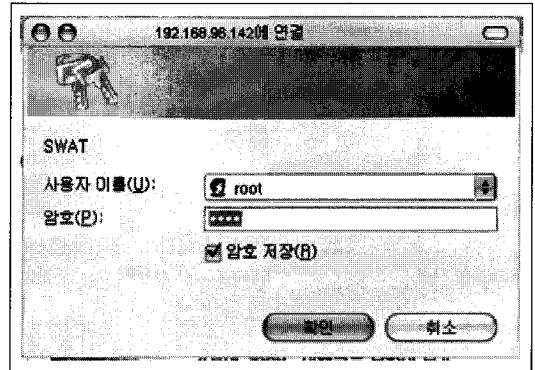


그림 6. SWAT
Fig. 6. SWAT

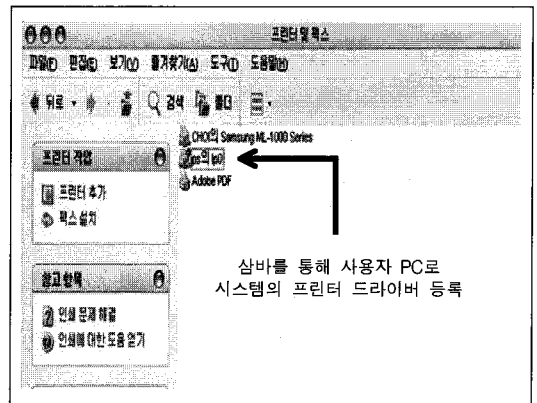


그림 7. Window상에서 삼바 실행
Fig. 7. SMBA Execution on Windows

이러한 시스템을 위하여 GUI를 위한 윈도우 시스템을 사용하게 되는데 이때 시스템의 외관을 결정짓는 윈도우 매니저와 윈도우 매니저 위에서 실행되는 응용프로그램을 개발해야 한다. Linux Window 시스템은 X-Window시스템과 Qt, GTK, Micro Window등이 있지만, 본 시스템에는 다른 GUI보다 많은 Widget 라이브러리를 지원하며, 크기가 작고, 개발하기 쉬운 Qt기반의 윈

도우 시스템인 Qtopia를 사용하였다.

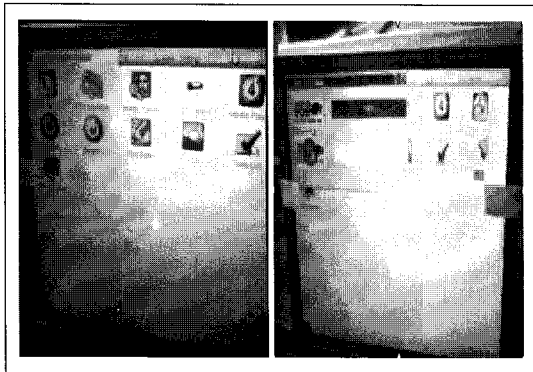


그림 8. GUI 실행화면
Fig. 8. Display of GUI

USB 인터페이스는 USB저장장치를 이용하여 시스템 상에서 직접 문서를 출력하기 위하여 구현하였고, USB는 Linux에 구현되어 있으므로 본 시스템에서는 USB가 지원되도록 커널 옵션을 수정한 후 사용하였다. Linux에서 USB 인터페이스를 사용하기 위해서는 특정 디렉토리와 해당 장치간에 Mount를 하여 사용하는데 본 시스템에서는 이 부분에 관해서 사용자가 사용하기 쉽게 GUI 응용프로그램을 작성하였으며, 사용자는 그림 8에서와 같이 터치LCD상의 아이콘을 통하여 USB 저장 장치내의 문서를 쉽게 열람하고 직접 프린터 작업이 가능하다.

IV. 결론

본 논문에서는 소규모 사무실 환경 하에서 네트워크를 통한 프린터 작업을 할 때 기존의 방식인 네트워크에 연결된 개인 PC를 이용한 방식에서 발생하는 여러 가지 불편함을 해소하기 위하여 제안되었다. 본 논문에서 구현한 시스템은 프린터 서버 및 대용량 프린터 작업시 발생하는 부하를 30%이상 저감할 수 있었다. 하지만 현재 대부분의 PC사용자가 이용하는 운영체제는 MS사의 Windows이다. 그리고 대부분의 문서작성을 위한 응용 프로그램들이 Windows기반에서 사용되고 있다. 따라서 본 시스템과 같은 다른 기종의 운영체제인 Linux에서 프

린터 작업을 수행하는 것은 그래픽 출력의 한계와 Power Point나 Excel 같은 프로그램의 문서들은 출력되지 않는다는 문제점을 가지고 있다.

그래픽 출력의 문제점은 차후 Ghostscripts를 시스템에 포팅하여 해결방안을 마련하고, Windows기반의 응용프로그램들은 다시 Linux에서 출력 가능하도록 파일 변환 응용프로그램을 제작하면 해결할 수 있을 것이다. 향후 이런 문제점들을 보완하고 과부하 지속시간 저감 등 다양한 환경에서 시스템의 부하를 저감 시킬 수 있는 관련 연구가 요구된다.

감사의 글

본 연구는 지식경제부의 지역혁신센터사업과 동서대학교 특별연구비의 결과로 수행되었음.

참고문헌

- [1] Daryll Stauss, Wook, "Linux Helps Bring Titanic to Life", Linux Journal, February 1998.
- [2] Dijkstra, E., "Communicating Sequential Processes", Technical Report EWD-123, Technical University Eindhoven, 1965
- [3] Goldberg, G., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Reading MA, 1989
- [4] 박영환 "임베디드 시스템 임베디드 리눅스". 사이텍미디어, June, 2002
- [5] Linux Kernel Documentation "Par Port-Low Level .txt"
- [6] 유영창, 안윤호, 오재경, 유명환, "TTEPERT 리눅스디바이스 드라이버", 한빛미디어, December, 2004
- [7] Alessandro Rubini, "Linux Device Driver(2nd Edition)", O'RELLY, July, 2001
- [8] 이연조, "임베디드 리눅스 프로그래밍", PCBOOK, March, 2002.

저자소개



차경환(Kyung-Hwan Cha)

1985년 부경대학교 전자통신공학과
(공학사)

1990년 부경대학교 전자통신공학과
(공학석사)

1996년 부경대학교 전자공학과(공학박사)

1995년~현재 동서대학교 컴퓨터정보공학부 교수

2003년~현재 동서대학교 유비쿼터스 어플라이언스
지역혁신센터 센터장

※관심분야: 임베디드시스템, 디지털신호처리