제20회 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호 (2003. 11)

PDA 용 임베디드 리눅스 시스템 설계 및 구현

장승주*, 홍영일*, 류진영*
*동의대학교 컴퓨터공학과
e-mail : sjjang@dongeui.ac.kr,
kincrew@hotmail.com , ros207@naver.com

Design and Implementation of the Embedded Linux System for the PDA System

Seung-Ju Jang*, Young-Il Hong*, Jin-Young Ryu*
*Dept. of Computer Engineering, Dong-Eui University

요 약

본 논문에서는 PDA 용 임베디드 리눅스 시스템을 설계 및 구현한다. 개발을 위한 Host PC 를 구축하고 PDA 시스템에 적합한 크로스 컴파일러(Cross compiler), PDA 와의 터미널을 위한 미니컴(minicom) 등을 이용한 개발 환경을 구성한다. 본 논문에서 PDA 시스템에 리눅스를 포팅 하여 PDA 임베디드 리눅스 시스템으로 설계 및 구현한다.

1. 서론

최근 리눅스 OS 의 사용이 증가 하고 있다. 그 중 임베디드 시스템(Embedded System)이 리눅스의 한 부분으로 자리 매김 되고있다. 임베디드 시스템은 어떠한 장치가 다른 시스템에 의존하지 않고 독립적 기능을 수행하는 것으로 우리 생활 속에서 이미 많이 쓰이고 있다. 이러한 임베디드 시스템은 다양한 응용분야를 가지고 있어 각종 분야에 적용되고 있고 프로그램의 크기 증대에 따라 이를 안정적이고 효율적으로 통제 하기 위한 OS 가 필요하게 된 것이다[3,4].

운영체제에 따라 시스템의 성능 및 확장성에 영향을 미친다. 임베디드 시스템에서의 운영체제는 시스템의 크기 증대에 따라서 Multi-Tasking 과 같은 복잡한 기능을 요구하며, TCP/IP, GUI, Audio, Video 등 Network 이나 Multimedia 가 시스템의 기본으로 자리를 잡고 있다. 임베디드 시스템이 해야 할 일이 많아지고 복잡해 짐으로써 단순히 순차적으로 운영되는 프로그램으로는 시스템을 운영하기가 어렵게 되어 임베디드 시스템에도 운영체제의 개념이 필요하게 되었다. 그리고 임베디드 시스템의 특성상 실시간이라는 요소 또한 중요한 요소로 자리 잡고 있으며 이를 만족시켜야 만 하는 시스템도 늘어 나고 있다.

임베디드 시스템을 운영하기 위한 WinCE, pSOS, VxWorks, Palm OS 등 다양한 운영체제가 사용되고 있다. 이러한 OS 중의 하나가 바로 리눅스이다.

임베디드 리눅스 시스템은 많은 시스템에 의해 검증이 되어 있고 소스가 공개되어 있어 개발자 스스로가 커널을 조정하거나 수정 할 수 있다. 오류가 포함될 가능성이 적어 안정적이고, 다양한 응용이 가능해 성장 가능성이 높다[4].

본 논문의 구성은 2 장에서 관련연구를 살펴보고, 3 장에서 설계 및 구현에 대해 살펴보고, 4 장에서 결론에 대하의 논의한다.

2. 관련연구

현재 임베디드 리눅스의 활용처로 가장 많이 알려진 분야는 PDA(Personal Digital Assistant)용 임베디드 리눅스 시스템이다. PDA 는 휴대형 정보 단말기를 말한다. 노트북보다 훨씬 작은 소형 컴퓨터이면서 전자수첩보다 강력한 컴퓨팅 파워를 갖고 있다.

리눅스 PDA 운영체제라면 간편하면서 효율적인 운영체제를 구현한 Palm OS 를 꼽을 수 있다. 그러나 멀티미디어 기능이 강조되면서 WinCE 와의 힘겨루기에서 다소 밀리는 형상을 보이고 있다. WinCE 는 화려한 멀티미디어 기능과 데스크톱과 비슷한 사용자 인터페이스로 인해 높은 하드웨어 사양이 필요하고, 실행 속도도 느리다는 단점을 갖는다. 임베디드 리눅스는 이런 단점을 해결해 Palm OS 의 간결함, 직관적인 인터페이스, 빠른 실행 속도와 WinCE 의 화려한 멀티미디어 기능을 접목시킨 운영체제다[3].

3. 설계와 구현

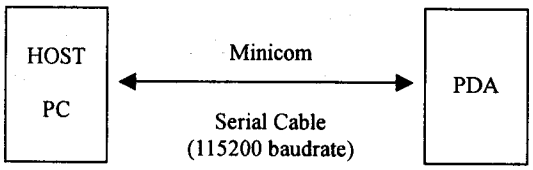
3.1 개발 환경

가상 메모리 압축 시스템 설계 환경은 아래와 같다.

[표 1] 기본 시스템 환경

	Host PC	Target board
Host	Intel Pentium III Celeron 933MHz 128MB ram	IPAQ H3630 (StrongArm-1110) 32MB ram
OS	Linux-2.4.13-1hl (Hancom 2.2)	Linux-2.4.18 -rmk3-hh6-j3
GCC	Gcc-2.96	Cross-2.95.3
Target Board ETC	Boot loader : linuette Ramdisk : mizi + debug File system : jffs2 (Journaling Flash File System version 2)	

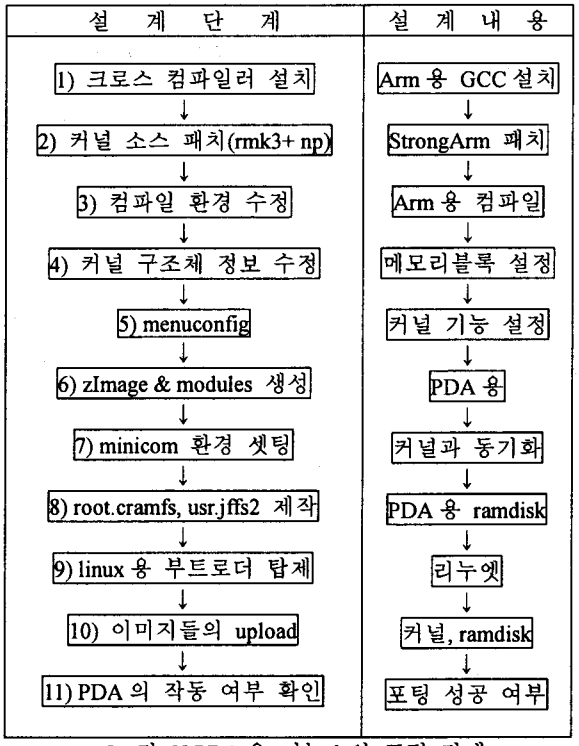
[표 1]은 본 PDA 시스템 설계에 사용된 호스트 컴퓨터와 PDA 의 개발 환경에 대한 개괄적인 설명이며, 호스트 PC 와 PDA 는 아래의 [그림 1]과 같이 통신 에뮬레이터인 minicom 으로 Serial Cable 을 사용하여 115200 baudrate 로 1:1 로 연결된다.



[그림 1] HOST PC 와 PDA 의 연결

3.2 설계 내용

3.2.1 Linux 를 PDA 에 포팅하는 단계 요약



[그림 2] PDA 용 리눅스의 포팅 단계

3.3 Linux 를 PDA 에 포팅 과정

3.3.1 크로스 컴파일러 설치

호스트 PC 에 StrongArm 용 크로스 컴파일러를 제작하여야 한다. 이때 gcc 와 관련된 환경 (binutil, glibc, gcc 등)이 하나라도 맞아 떨어지지 않으면, 크로스 컴파일러가 재대로 컴파일 되지 않을 가능성이 많다. 본 논문에서는 크로스 컴파일러는 편의상 공식 ARM 리눅스 사이트에서 배포하는 cross-2.95.3.tar.gz 파일을 설치하여 사용하였다. 압축 해제시 I(install) 옵션을 넣어서 환경 변수를 자동으로 셋팅 되도록 해야만 재대로 설치가 된다. 그렇지 않으면, 사용 시 에러가 난다. 압축을 해제하면, $PWD/2.95.3/bin 의 arm-linux-??? 라는 크로스 컴파일러를 사용할 수 있다.

3.3.2 커널 소스 패치(rmk+ np)

우리가 만들고자 하는 커널은 StrongArm-1110 용 커널이다. 이 커널을 만들기 위해서는 기본적으로 아래와 같은 파일들이 필요하다. ftp.mizi.com에서 이미 patch 된 커널을 받아 사용해도 된다. 본 논문에서는 linux-2.4.18-rmk3-hh6-j3 커널을 중심으로 한다.

[표 2] 커널 소스 패치를 위해서 필요한 파일들

파일	용도
Linux-(커널버전).tar.gz	linux 커널의 PC 용 버전으로 이것에 동일 커널 버전의 arm 및 StrongArm 패치를 적용한다...
Patch-(커널버전)-rmk(숫자).gz	Arm patch 로 커널에 arm 용 CPU 를 사용할 수 있도록 patch 할 때 쓰인다. 요즘의 rmk 는 nico patch 가 되어 있다고 한다..
diff-(커널버전)-rmk(숫자)-np(숫자).gz	Nico patch 로 커널에 StrongArm 용 CPU 를 사용할 수 있도록 patch 할 때 쓰인다.

3.3.3 컴파일 환경 수정

커널소스의 루트에 있는 Makefile 을 [그림 3]과 같이 커널버전 및 아키텍쳐, 크로스 컴파일러를 변수를 수정한다.

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 18
EXTRAVERSION = -rmk3
.....
ARCH := arm
.....
CROSS_COMPILE = (크로스 컴파일러가 설치된 루트 폴더의)/bin./arm-linux-
.....
```

[그림 3] 수정된 Makefile

또한, include 의 아키텍쳐도 Make 할 아키텍쳐와 동일하도록 [그림 4]와 같이 수정해준다.

```
ln –s include/asm-arm iclude/asm
ln –s include/asm/arch-sa1100 include/asm/arch
ln –s include/asm/proc-armv include/asm/proc
```

[그림 4] include 의 아키텍쳐의 수정(심볼링크)

3.3.4 커널 구조체 정보 수정

커널 소스의 drivers/mtd/maps/ 폴더 아래에서 대상 PDA 의 커널 구조체 부분을 결정하는 sa1100-flash.c 파일의 구조체 부분을 수정했다. (menuconfig 에 의해서 자동으로 설정이 되기는 하나, 정확하다고 볼 수는 없었다. 여기서는 linux-2.4.5-rmk6-np1 의 미리 만들어진 zImage 의 mtdblock 영역크기를 참고로 수정 했다. 이때, 주의해야 할 점은 새로 만들어 올릴 이미지가 구조체에서 정해준 영역의 크기를 넘어서지 않게 하기위해서 미리 만들어진 zImage 의 크기보다 작은 zImage 를 컴파일해야만 한다는 것이다.)

```
.....
static struct mtd_partition h3xxx_partitions[] =
{
  {name: "H3XXX boot firmware",size: 0x00040000,offset: 0,
  mask_flags: MTD_WRITEABLE, },

  {name: "H3XXX params",size: 0x00040000, offset: 0x40000},

  {name: "H3XXX kernel",size: 0xc0000,offset: 0x80000},

  {name: "H3XXX root cramfs",size: 0x140000,offset: 0x140000},

  {name: "H3XXX usr jffs2",offset: 0x280000,size: 0xd80000}
};
.....
```

[그림 5] 커널 구조체를 수정한 부분

위의 [그림 5]는 커널의 mtdblock 구조체 정보를 선언하고 있는 /drivers/mtd/maps/sa1100-flash.c 파일의 해당 mtdblock 구조체 정보를 수정한 것이다.
"static struct mtd_partition h3xxx_partitions[] =" 부분을 수정하였는데, 이것은 본 논문에서 사용한 CPU 의 커널 configuration 에서 CPU 의 모델명을 선택해 주었을 때, zImage 의 mtd 구조체의 설정이 위의 함수를 토대로 만들도록 설계되어 있기 때문이다.

3.3.5 menuconfig

해당 CPU 에 맞게 System type 과 MTD 를 설정하고, LCD 와 Minicom 의 사용을 위해서 Virtrual terminal 기능을 사용한다. 특히 minicom 과 kernel, 부트로더가 서로 동기화 되도록 Character Devices 의 Serial dirvers 의 Default baudrate 를 115200 으로 맞춰준다. File systems 에서 jffs2 를 지원하도록 한다. Console driver 에서 Vga text console 을 disable 시키고, frame-buffer support 의 LCD 를 support 시켜줌으로써 LCD 를 사용 가능 하도록 한다. 터치스크린 부분은 System type 의 SA-11x0 의 Implementations 의 HAL abstraction layer 에서 Hardware micorcontroller 를 선택해줘야 Character Devices 에서 선택이 가능하다.

참고로 menuconfig 부분은 커널의 버전이나 패치 상태에 따라서 다를 수 있으며, 개발자가 직접 수정하거나 만들어 넣을 수도 있다.

3.3.6 zImage & modules 생성

일반 PC 에서의 커널 컴파일과 동일하게 make 한다. 단, make modules_install 은 PC 용이 아닌 관계로 /tmp/modules 폴더 아래에 설치된다.

3.3.7 minicom 환경 셋팅

minicom –s 옵션을 사용하여 minicom 의 환경설정을 해준다. Configuration 중 Serial Port Setup 에서 Serial device 를 /dev/ttyS0 로 Bps/Par/Bits 를 115200 8BN 으로 각각 설정한다. 참고로 이 Serial device 는 host PC 의 minicom 과 PDA 의 부트로더를 이어주는 매개체이며, Bps/Par/Bits 를 115200 8BN 으로 적용해 준 것은 앞서 커널 menuconfig 의 baudrate 와 PDA 의 config.scr 를 통한 설정이 동일해야만 호스트 PC 와 PDA 가 동기화가 되기 때문이다.

3.3.8 root.cramfs, usr.jffs2 제작

필요에 따라서 ramdisk 를 작성할 수 있다. 리눅스의 기본적인 부분(폴더 및 파일)을 갖춘 ramdisk 폴더를 만들고, 그 폴더를 ramdisk 이미지로 변환하면 된다. Ramdisk 를 만드는 법은 [그림 6]과 같다.

```
mkcramfs 대상폴더 출력파일
mkfs.jffs2 –d 대상폴더 –e [Size] –l –o 출력파일
```

[그림 6] ramdisk 이미지의 제작방법

특히 root.cramfs에서는 fstab설정을 usr.jffs2에서는 modules부분 및 X 환경을 구성해주는 BusyBox를 수정 및 추가해 주었다.

3.3.9 linux 용 부트로더 탑재

우선 WinCE 가 설치되어 있다면, Active-Sync 가 어플리케이션을 설치해줘야 한다. 그후, Wince 의 Active-Sync 프로그램을 통해 부트로더의 이미지를 올리고 교체 작업을 한다.. 리눅스용 부트로더가 설치되 있을 경우, 해당 부트로더를 사용하거나 해당 부트로더 명령을 통해 linuette 부트로더를 업로드 시킬 수 있다.

3.3.10 이미지들의 upload

PDA 용 부트로더의 환경변수중 linuxargs, 램디스크의 시작주소, baudrate 및 partition 정보등을 지정해주고 파라미터를 save 한다.

또한 커널 이미지를 올릴 영역인 kernel 을 load 한 다음 xmodem 을 사용하여 zImage 를 업로드하고, root 영역과 usr 영역을 각각 load 해서 root.cramfs, usr.jffs2 를 업로드 시킨다.

3.3.11 PDA 의 작동 여부 확인

PDA 에 리눅스가 재대로 포팅이 되었는지 확인한다. /etc/fstab 의 정보대로 마운팅 되었는지를, ramdisk 이미지가 재대로 업로드되었는지, cat /proc/mtd 를 이용하여 mtd 블록의 설정이 재대로 되었는지를 확인하면 된다.

4. 결론

본 논문에서는 PDA 라는 매체를 타겟보드로 사용하여 리눅스를 포팅함으로써 임베디드 리눅스 시스템을 설계 및 구현하였다. 또한 타겟보드만으로 동작할 수 있도록 BusyBox 를 포팅하여 X 어플리케이션 및 터치스크린을 구현하였다. 향후 타겟보드나 소형 단말기의 사용이 보편화되면, 이에 관련된 분야에서 좀더 안정적이고 원활하게 작동할 수 있는 임베디드 리눅스 시스템을 구축하는 작업이 필요할 것이다. 본 논문은 이 작업의 기초적인 부분을 다루며. 앞으로도 이 환경을 이용하여 또 다른 연구가 시도 될 것이다.

참고문헌

[1] RUNNING LINUX 3rd Ed, Matt Welsh, Lar Kaufman ,Kalle Dalheimer, O'REILLY,1999

[2] Embedded Linux Hardware, Software, and Interf acing, Craig Hollabaugh, Ph.D., Addison Wesle y, 2002

[3] http://www.zdnet.co.kr/biztech/hwsw/ techtrend/article.jsp?id=47230&page=1&forum=0

[4] http://www.zkernel.com/linux/

[5] www.unix.co.kr/data/kldp_org/KoreanDoc/html/ EmbeddedKernel-KLDP/index.html

[6] CONCEPTS, TECHNIQUES, TRICKS, AND TRAPS Building Embedded LINUX SYSTEMS, Karim Yaghm our, O'REILLY, 2003

[7] www.mizi.com/en/developer/s3c2410x/

[8] embedded-inux.hanbitbook.co.kr/Article/swanmt d/swanmtd.html#toc4

[9] www.falinux.com/win/study/01/strongarm01.html

[10] www.linuxlab.co.kr/docs/minicom.htm