개인 휴대정보 단말기를 위한 내장형 리눅스 설계 및 구현

정홍섭, 박철우, 박규석
경남대학교 컴퓨터공학과

Design and Implementation of embedded Linux for PDA

Hong-seob Jung, Chol-woo Park, Kyoo-seok Park
Dept. of Computer Engineering, Kyungnam university

요 약

PDA(Personal Digital Assistant)는 GUI(Graphical User Interface) 환경이 필수적이다. 최근 기존의 개인정보 관리기능 외에 휴대단말기를 내장한 인터넷 기능, 멀티미디어 기능, 게임 등이 가능한 고성능의 PDA가 일반화됨에 따라, 기존의 단일 task 환경 대신 뛰어난 GUI와 멀티미디어 환경을 지원하는 내장형 운영체제가 사용될 전망이다.

본 논문에서는 Linux의 GTK+/GDK(GNU Image Manipulation Program Toolkit) 툴킷을 이용하여 Framebuffer를 직접적으로 사용할 수 있게 하고, 불필요한 내부 함수를 제거함으로써 X-윈도우시스템으로 인한 오버헤드 및 저장공간을 줄일 수 있는 내장형 운영체제를 설계 및 구현 한다.

1. 서론

기술의 발전으로 사용자가 언제, 어디서든 PC를 사용하고, 인터넷에 접속하고자 하는 욕구를 수용할 수 있는 장치 중의 하나로 PDA(Personal Digital Assistant)가 개발되었다. PDA는 개인용이나 업무용으로 계산이나 정보저장 및 검색 기능을 갖춘 손바닥 크기의 소형장치로, 스케줄 캘린더와 주소록 정보 등을 유지하는데 많이 쓰이고 있다. PDA의 입력 방식은 작은 키보드를 사용했었지만, 소형화에 추세에 따라 간단한 몇 개의 버튼과 터치스크린으로 대신하게 되었다. 따라서, 사용자의 입력은 화면에 나타나는 그래픽화면에 펜(stylus)을 이용하여 작업하는 방식으로 변경되었으며 사용자 편의를 위해 보다 편리한 GUI (Graphical User Interface) 환경을 갖추고자하는 연구가 활발히 진행되고 있다[1].

최근 기술의 발전으로 고성능의 micro process와 높은 용량의 비휘발성 메모리의 개발로 PC에 상응하는 성능을 가지고 있으며, 개인정보 관리기능 외에 무선통신을 연계한 인터넷·네트워크 기능, 멀티미디어 기능, 게임 등의 기능을 가지고 있다.

이에 따라, 기존 MMU(Memory Management Unit)가 없는 PDA에서 동작하는 단일 task 운영체제는 느린 처리 속도로 인하여 멀티미디어 구현이 어렵고 장치사용에 제한이 많아 점차 사라지게 되고, 뛰어난 GUI 환경과 멀티미디어 기능을 지원하는 내장형 운영체제 (Windows CE, Embedded Linux)로 전환되고있다.

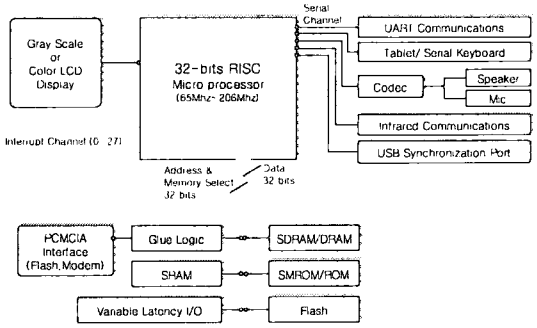
본 논문에서는 가격 경쟁력에 앞서고, 안정성, 네트워크 기능, 다양한 장비지원, 낮은 개발비용 등의 많은 장점을 지닌 공개 운영체제인 리눅스를 이용하여 PDA 환경을 지원하는 내장형 리눅스 시스템을 설계하였다.

2. 관련 연구

2.1 PDA 내부 구조

최근 PDA는 그림 1과 같이 고성능 micro processor와 flash-ROM, RAM, LCD, Touch-screen, LCD, Audio CODEC, 기타 직렬입출력 장

치로 구성되어 있다[2].



<그림 1 > PDA 내부 구성도

2.2 Linux Framebuffer Device

표준 그래픽 처리의 표준화를 위하여 개발된 리눅스 커널 Device(/dev/fb)로 카드의 종류에 관계없이 파일의 입·출력을 통하여 그래픽 처리가 가능하도록 설계되었다. Device의 Memory Mapping을 이용하여 메모리를 직접 접근함으로서 고속의 그래픽 처리가 가능하다[3].

2.3 GTK+(GNU Image Manipulation Program Toolkit)/ GDK Library

GUI 기반 application framework 제공하는 툴킷으로 X-window상에서 그래픽처리 프로그램 개발을 위한 Library로 설계되었으며 현재 다양한 응용프로그램의 개발에 사용되고 있다. C언어를 기반으로 하는 객체지향 구조로 적용이 쉽고, 많은 프로그램 언어에서 사용하는 Open Source Library이다.

GTK는 윈도우의 다양한 Widget(윈도우, 버튼, 콤보박스, 체크박스 등)의 처리를 담당하고, GDK는 GTK 구현을 위한 저 수준 Library로 이벤트처리 및 그래픽처리(점, 선, 박스 등)를 담당하고 있다[4].

2.4 JFFS2 (Journalling Flash File System version 2)

Log-structured 파일시스템인 JFS을 Flash 장치에 적용시킨 기술로 Axis Communications 사에서 설계되어 현재 Open Source로 되어 있다.

PDA와 같이 전원 공급이 불안정한 시스템에서 전원 공급의 중단에 따른 파일시스템 보호를 위하여 사용되고 있다. 읽기 전용이었으나, 현재 버전 2 에서 쓰기 및 블록압축을 적용하여 읽기 속도 및 저장용량이 증가하였으며, 압축에 따른 쓰기 시간 증가와 XIP (eXecute-In-Place) 미 지원, 블록 단편화에 처리에 따른 추가 시간 소요 등의 단점이 있다[5].

3. 시스템 설계

3.1 Embedded Linux Kernel 구성

내장형 커널의 주요 역할은 메모리관리, 프로세서 및 스레드 생성, 내부프로세서간의 통신, 인터럽트 관리, XIP-ROM 파일 시스템, RAM 파일 시스템, Flash 관리, TCP/IP 네트워 지원 등이다. 이에 따라, 적은 자원을 효율적으로 사용하기 위하여 커널은 부팅 시 필요한 모듈을 제외한 나머지는 실행시간에 모듈로더에 의하여 적재된다. PDA의 용도에 따라 커널을 구성하게 되며 주변장치 모듈, 네트워 모듈, 파일시스템 모듈 등이 선택적으로 커널에 적재된다[6].

다음은 PDA에 적합한 내장형 리눅스 커널의 일반적인 구성이다.

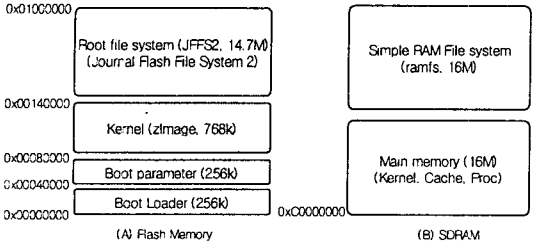
```
Kernel Version: Linux-2.4.12 ac6-rmk2

Loadable Module : Enable Kernel module loader
System Type : SA-1100 based
General Setup :
    · Network, System V IPC, Sysctl support
    · Kernel core(ELF), ELF format support
    · FastFPE math emulation (m)
    · Power Management support
Memory Technology Devices :
    · MTD partitioning support
    · Direct char device access (M)
    · Caching Block device access
    · RAM/ROM/Flash chip driver
      - CFI support: CFI advanced configuration
        · Specific CFI Flash geometry selection
        · 16,32-bit buswidth, 1,2-chip interleave
        · Intel flash chip support
    · Mapping drivers : StrongARM SA-1100
Block devices
    · Loopback device support (m)
    · Network block device suppirt (m)
    · Disable Ramdisk
Character devices
    · Virtual terminal, Support for console
    · Standard/ generic serial support(m)
    · SA1100 serial support, console on serial port
    · SA1100 Real Time Clock(m)
    · Touchscreen & Button support (m)
    · Unix98 PTY support (32)
    · Mice : Mouse Support (m)
File systems:
    · JFFS2, RAMFS(M), ext2(m),
    · /proc, /dev file system & automount, /dev/pts
Console drivers:
    · Disable VGA text console
    · Frame-buffer support
      - SA1100 LCD
      - Advenced low level driver options : 16 bpp
      - Support only 8 picels wide fonts : VGA 8x8
```

3.2 Flash Memory Partition

Flash는 전원이 없이도 정보저장이 가능한 영역으로 용도에 따라 여러 부분으로 분할되어 사용된다. 주로 사용되는 NOR-Flash의 경우 제조사의 Flash Translation Layer를 통하여 Block 단위의 입·출력을 수행하므로 Block 크기로 분할하여 메모리 접근 회수를 줄일 수 있다.

아래 그림 3은 본 논문에서 제안하는 Flash Memory 구성도로 접근 블록의 크기인 0x40000 (256K) 단위로 분할되었다[7].



<그림 2 > Flash Partition

Boot loader 영역은 커널의 주 메모리 적재 기능 외에 Flash 관리 (read, write, erase, partition, boot parameter, upload, download) 기능을 담당하고 있다. Boot parameter 영역은 Boot loader에 의하여 관리되는 커널 parameter, Flash partition, 커널의 분할위치 등의 부가 정보가 저장되는 영역이다. Kernel 영역은 커널의 이미지가 저장되는 영역으로 주로 512KByte를 사용하고 있으나 다양한 장비지원 및 버전 업그레이드를 위하여 768KByte로 설계하였다. Root 영역은 커널 적재 후 root(/)로 마운트 될 파일시스템이 위치하게 되며 JFFS2 파일시스템으로 설계하였다. 읽기 전용 파일 시스템인 CRAMFS (Compressed RAM File System)가 주로 사용되어 왔으나 읽기 전용에 따른 시스템 설정 적용이 불가능한 단점이 있어 일부 중요 시스템 파일 영역에서 사용되고 있다.

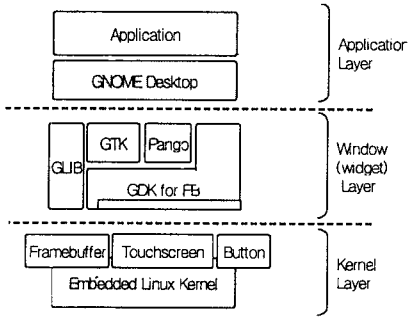
RAM은 프로세서를 위한 시스템 영역과, 정보저장을 위한 파일시스템 영역으로 사용된다. 시스템 영역은 커널, Flash 입·출력 Cache, Buffer Cache, Page Cache 등으로 16MB가 사용되며, 나머지 영역은 사용자가 정보 저장을 위하여 사용하는 RAMFS(Simple Ram File System) 영역으로 사용된다.

3.3 GUI 시스템 설계

PDA는 작은 크기로 인하여 터치스크린과 버튼을 입력장치로 사용하고 있다. 특히, 터치스크린에 크게 의존하고 있으며, 사용자 편의를 위한 윈도우 Widget 구현과, 필기인식 등의 높은 GUI 환경이 요구된다.

본 논문에서는 그림 4와 같이 X-window를 기본 윈도우 시스템으로 채택하고 있는 기존 Linux와 달리 Framebuffer device를 직접 처리하는 GTK GUI 툴킷을 이용하여 불필요한 윈도우 처리 및 Network 처리에 따른 오버헤드와 저장 공간를 줄일 수 있도록 설계하였다.

X-window 제거에 따른 유연성 손실의 단점이 있지만, 기본 Library인 XLib는 프로그램 개발이 어려워 많은 응용프로그램들이 GTK 툴킷을 기반으로 작성되고 있어 대부분의 응용프로그램의 실행이 가능하다.



<그림 3 > GUI system 구성도

4. 시스템 구현

4.1 개발 환경

본 논문에서 구현된 System의 개발 환경은 다음과 같다.

```
- System :          PentiumIII 1GHz (mem: 512MB)
- OS     :          Linux Redhat 7.1 (kernel 2.4.12)
- Cross-Compiler & linker :
        · GNU arm-linux-gcc          : 2.9.3
        · GNU arm-linux-ar, ld       : 2.9.5
```

또한, Embedded linux 개발환경은 Compaq사의 iPaq H3600 series로 사양은 다음과 같다.

```
- Processor:    Intel StringARM(SA-1110)
                32-bit RISC Processor
- Memory:       Flash 16MByte, SDRAM 32MByte
- Display:      TFT LCD 240x320 (12bit color)
- Interface:    Button x8, USB/RS232C, IrDA
- OS : Linux 2.4.12-ac6-rmk2
```

4.2 Embedded Linux 구현

4.2.1 Kernel 구성 설정

PDA의 용도에 적합한 커널 설정 후 Flash Partition 정보를 적용한다.

```
./kernel/drivers/mtd/maps/sa1100-flash.c

max_flash_size = 0x01000000;

{   name: "H3600 boot firmware",
    offset: 0,
    size: 0x00040000,
    mask_flags: MTD_WRITEABLE
},{ name: "H3600 params",
    offset: 0x40000,
    size: 0x00040000
},{ name: "H3600 kernel",
    offset: 0x80000,
    size: 0x000C0000
},{ name: "H3600 root jffs2",
    offset: 0x00140000,
    size: MTDPART_SIZ_FULL
}
```

1) 크로스 컴파일

GNU Open Source 프로그램의 경우 대부분 소스 형태로 제공되고 있으며 autoconf 툴에 의하여 시스템 환경이 검사되고 Makefile이 생성되어 컴파일이 진행된다. Library 및 Platform 종속적인 특성을 파악하기 위한 실행시간 환경검사는 검사용 코드가 직접 작성되고 컴파일 및 실행을 통하여 검사(형 크기 검사, 스레드 검사 등)가 이루어지므로 크로스 컴파일의 경우 Target Platform 기계어로 컴파일은 가능하나 실행이 불가능한 문제가 있다. 따라서, 네트워크 연결을 통한 NFS (Network File system)와 rsh 명령으로 Target 기계에서 수행시켜 결과를 검사하는 방식으로 진행한다.

2) 커널 컴파일

Processor와 PDA에 환경에 따른 커널 패치 및 설정 후 cross-compiler를 이용하여 컴파일 한다.

```
/linux-2.4.13-ac4-rmk1

# make ARCH=arm xconfig
# make ARCH=arm CROSS_COMPILE="/usr/local/bi
n/arm-linux-" dep clean zImage modules modules_in
stall
# cp ./arch/arm/boot/zImage /h3600
# cp -r /lib/modules/2.4.13-ac4-rmk1/* /h3600/lib/mo
dules
```

3) Linux 기본 시스템 파일 컴파일

Linux 시스템을 구성하는 기본적인 실행 화일을 컴파일한다.

```
- modules-2.4.10    : 모듈검색 및 적재 (insmod)
- fileutils-2.4.10  : 기본 파일 처리 (ls, cp)
- sh-utils-2.0      : shell 처리 (stty, su, who)
- netkit-base-0.17  : network 기본 처리(inetd, ping)
- util-linux-2.10s  : 리눅스 기본 Utility(mkfs, kill)
- SysVinit-2.78     : SysVinit 명령 (init, shutdown)
- 기타

ex) 컴파일예

/modules-2.4.10
# configure --prefix=/h3600/modules --target=arm-linux
# make CFLAGS="-O2 -mapcs-32 -march=armv4 -mtu
ne=strongarm1100 -mshort-load-bytes" all install
```

4.2.2 GTK+/GDK 컴파일

Framebuffer Device GTK 툴킷의 컴파일을 수행 한다. GTK의 기본 라이브러리인 GLib, 폰트를 처리하는 Pango Library, PDA에서 사용하게 되는 장애자 지원(On-screen 키보드 등) ATK Library 및 GTK Library을 컴파일 한다[8].

```
/glib-1.3.1
# configure --target=arm-linux --prefix=/h3600 --ena
ble-debug=no --disable-mem-pools --without-x
# make all install

/pango-1.3.1
#configure --target=arm-linux --prefix=/h3600 --ena
ble-debug=no --with-include-modules=yes --without-
x
#make all check install

/gtk-1.3.1
#configure --target=arm-linux --prefix=/h3600 --enab
le-debug=no --without-x --with-target=linux-fb --di
sable-shadowfb --disable-modules --with-included=lo
aders=xpm, png, jpeg

#make all check install
```

4.2.3 시스템 설정 및 부팅

시스템 환경 설정에 필요한 /etc의 파일을 기존 Linux 시스템에서 복사하여 편집한다. 부팅시 /linuxrc, /etc/inittab, /etc/rc* 등이 필요하며 아래와 같은 작업을 수행한다.

```
- Power on
- Boot loader        (Flash 0x0)
- Kernel Loading

- /linuxrc 실행
   ·Floating point emulator module load
   ·mount -n /proc
   ·RAMFS module
   ·Unix socket module
   ·mount /mnt/ramfs
   ·make temp system
   ·exec sbin/init

- /etc/inittab 실행
   ·/etc/rcS.d/S* 실행
    -mount local filesystems
    -module dependencies check
   ·/etc/rc2.d/S* 실행
   ·/sbin/getty 실행 (Serial terminal)
```

4.2.4 Root File System 구성

mkfs.jffs2 툴을 이용하여 Linux의 기본적인 작업에 필요한 File Utility, System Library, 시스템 설정 파일, 윈도우시스템파일, 공유 라이브러리 Image를 Flash Block 크기를 고려하여 생성한다.

```
# ls /h3600
  bin/  dev/  etc/  lib/  sbin/  usr/  /tmp@ /var@

# /mkfs.jffs2 -d h3600 -o /h3600.jffs2 -e 0x40000 -p
```

4.2.5 PDA의 Flash Memory에 적재

Flash 관리 및 부팅을 위한 Boot loader는 CRL/OHH ARM Boot loader를 사용하였으며, 커널 및 파일시스템 Image를 적재하기 전에 다음과 같이 Flash를 분할한다[9].

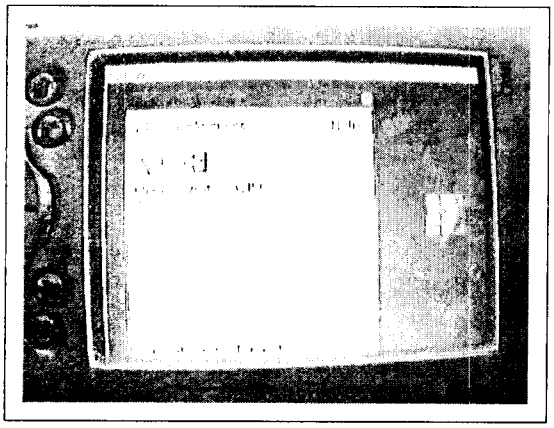
```
- partition define bootldr 0x0 0x40000 0
- partition define params 0x40000 0x40000 0
- partition define kernel 0x80000 0xc0000 0
- partition define root 0x140000 0xec0000 0x16
- partition save
```

Kernel Image와 Root Image를 적재 후 kernel의 parameter를 설정한다.

```
- load kernel
- load root
- set linuxargs "noinitrd root=/dev/mtdblock/3 init=/linuxrc console=ttySA0"
- params save
```

4.2.6 실행화면

그림 4는 PDA에 구현된 내장형 Linux의 GTK Demo 화면이다.



<그림 4> 내장형 Linux GTK Demo 화면

5. 결론 및 향후방향

본 논문에서는 Linux의 GTK+/GDK(GNU Image Manipulation Program Toolkit) 툴킷을 이용하여 Framebuffer를 직접적으로 사용할 수 있게 하고, 불필요한 내부 함수를 제거함으로써 X-윈도우시스템으로 인한 오버헤드 및 저장공간을 줄일 수 있는 내장형 운영체제를 설계 및 구현하였다.

GUI 환경을 지원하는 시스템은 크게 Microsoft PocketPC, Linux + QT/Embedded, Linux+ Kaffe+ Java AWT, Linux+ Microwindow+ GTK 등이 있으며 현재 Windows CE를 기반으로 제작된 PocketPC가 가장 많이 사용되고 있다. Linux는 아직 일반 사용자를 위한 응용프로그램이 적고, GUI 환경이 기존 Microsoft사의 Windows에 적응된 사용자들에게 이질감을 주어 향후 보다 미적이고 실용적인 GUI 환경과 개발자를 위한 통합개발환경(IDE)에 대한 연구 개발이 필요하다.

[참고문헌]

[1] "The Embedded Linux GUI/Windowing Quick Reference Guide", http://www.linuxdevice.com/articles, 2000. 3.

[2] Intel, "StrongARM SA-1110 Microprocessor Developer's Manual", 2000. 6.

[3] Geert Uytterhoeven, "The Linux Frame Buffer Device Sybsystem", Linux Expo 99, 1999. 07.

[4] Tony Gale, Ian Main, "GTK+ 1.2 Tutorial", http://www.gtk.org, 2001, 03.

[5] David Woodhouse, "The Journalling Flash File System", UK Unix User Group Linux 2001 Linux Developers Conference, 2001. 6.

[6] Gregory Haerr, "Overview of Linux for the Mobile Appllication Developer", the Mobile Application Development Conference, 2000. 09.

[7] Intel, "Understanding the Flash Translation Layer(TFL) Specification", 1998. 11.

[8] Alexander Larsson, "GTK+ for the Linux Framebuffer", Red Hat, 2001. 3.

[9] Georgia Tech, "CS3210 Operating System Design (Linux) Bootloader Project", http://www.cc.gatech.edu/classes/AY2002/cs3210_fall/, 10. 2001