

논문 2014-2-3

리눅스 커널소스 프로그램의 유사성

김인회*, 이규대**

A comparison of kernel source program on Linux OS devices

Kyu-Tae Lee*

요 약

정보기기의 개발제품에서 분쟁이 발생한 경우, 저작권 검증하기 위한 자료로써 소스코드가 사용되어야 하나, 도용이 의심되는 제품의 개발자는 검증이 어렵게 하기 위해 소스코드를 제공하지 않는 경우가 발생한다. 특히 임베디드 시스템의 경우, Linux OS의 kernel을 탑재하고, 그에 따른 응용프로그램이 다양한 언어를 사용하여 실행되도록 제작된다. 이러한 시스템이 복제 또는 도용되는 경우, 기존과 같이 단순히 한 프로그램 비교방법으로는 감정 결과를 얻을 수 없다. 즉, OS에 따라 운영방식의 차이, 그리고 사용언어에 따른 프로그램의 구조적 차이 등으로 분석에 어려움이 있다. 특히 운영체제의 공개 프로그램이 비중이 높아 유사성 비교 결과가 다르게 나타난다. 본 논문에서는 두 제품의 커널소스 전체에 대한 비교 방안을 조사 분석하였다.

Abstract

In case of conflict on the IT device product, it would be used the source code to verify the ownership. but it is general things for illegal user not to submit his own code for shielding them. On the embedded system, it is figured out for the device to have kernel as Operating system and application program with any languages. It is hard to verify these system which has OS and app in general approach on illegal usage. There are many difficulties on analisys as the difference to algorithm and language and operating system, especially comparision result can not be guaranties by the open source on Linux OS. In this paper, we estimated the comparision result with whole kernel source code.

키워드 : 커널소스, 하드웨어 감정, 정보기기

1. 서 론

정보단말기로 사용되는 대부분의 정보기기는 프로세서를 기반으로 하는 입출력 하드웨어와 운

* 공주대학교 대학원 정보통신공학과

** 공주대학교 정보통신공학부 (교신저자)

(email: ktleee@kongju.ac.kr)

접수일자: 2014.12.20 수정완료: 2014.12.26

영체제를 기반으로 하는 응용소프트웨어의 일체형으로 제작되고 있다. 정보기기의 제작자 관점에서는 독자적인 기능의 구현과 성능개선을 위한 목적으로 특정한 프로세서와 특정 입출력 인터페이스를 갖는 하드웨어를 설계하고 구현한다.[1] 특히 임베디드 시스템의 구성으로 제작되는 경우에는 운영체제를 사용한 응용기기를 제작하게 되며, 이 경우 많이 사용되는 OS가 Linux Kernel

이다.[12] 그러나 리눅스커널은 개발과정에서 실행파일형태로 시스템에 탑재되어, 소스의 내용을 관독하기 어렵게 구성된다.

이러한 시스템의 분쟁시 감정을 어렵게 하는 부분이 소스코드확보가 어려운 경우이다. 정보기기의 인터페이스기능이 복잡해지면서, 유사한 프로그램소스를 사용하는 현상이 증가하고 있다. 이러한 정보기기 분쟁에서 소스코드 기준의 감정을 어렵게 하는 부분이 프로그램 소스의 불성실한 제공이다. 감정인이 정확한 유사성 도출을 위해서는 분쟁 당사자들의 자료제공이 선행되어야 하나, 복제 의심을 받는 측에서 소스코드를 거부하거나 감정의 특성상 정보기기만 확보되어, 제품의 내부에 저장된 실행코드만으로 검증해야 하는 상황들이 발생한다.

본 연구는 커널코드에 대한 양측의 소스코드가 제공된 상황을 가정하고, 유사성비교에서의 과정 및 유사도의 객관성 확보에 대해 다룬다.

2. IT 정보기기 하드웨어 특징

정보기기는 기존의 대형시스템을 기반으로 개발되는 소프트웨어 중심의 기능에서, 휴대 가능한 소형 시스템으로 구현되고 있다. 제품이 휴대가능한 정도의 크기로 제작되기 위해서는 전원 사용의 시간과 내부 프로세서의 처리용량의 제한으로 기능의 제한이 따르는 단점이 있다.

휴대가능한 정보기기의 내부구성을 보면 그림 1과 같이 핵심기능을 담당하는 프로세서를 중심으로 프로그램의 임시저장소로 활용하는 RAM, 응용프로그램을 저장하는 FLASH ROM, 정보기기의 입출력용으로 사용되는 키보드, LCD 및 터치패드, 그리고 외부 통신이 가능하도록 하는 통신포트(USB, COM) 등이 연결된다.

이러한 하드웨어적인 구성을 바탕으로 개발자

에 익숙한 프로그래밍언어를 사용하여, 응용프로그램을 제작하고, 완성된 프로그램 실행코드를 FlashROM에 저장하고, 전원을 입력하여 동작시킨다. 이때 응용프로그램은 프로세서의 각종 기능과 주변장치(key pad, LCD monitor, LCD touch screen, 통신포트)를 효과적으로 활용하여, 사용자에게 유용한 기능의 단말기기를 제공한다.

작성한 프로그램은 compile 과정을 통해 특정 프로세서에 일치하는 실행코드로 완성된다.

이상의 내부구조를 갖는 하드웨어는 design 작업을 통해 그림2와 같이 사용하기 용이한 제품으로 완성된다.

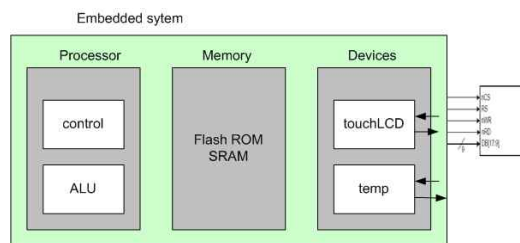


그림 1. 마이크로 컨트롤러 인터페이스 구성

단말기의 기능이 다양해지고, 핵심프로세서의 기능이 향상됨에 따라서, 임베디드 구조로 개발된 정보기기는 그림2와 같이 운영체제(OS)로 알려진 시스템관리용 프로그램이 탑재되어 제작된다. 운영체제는 일반 컴퓨터의 OS와 유사한 기능으로 주변기기, 메모리, 파일, 인터페이스 등 시스템의 모든 기능을 관리하는 능력을 가진 소프트웨어를 의미한다.

이러한 OS가 탑재되어, 단말기를 실시간으로 동작시키는 기기를 임베디드 정보기기라 한다. 따라서 소프트웨어 부분은 단순기능의 응용프로그램에 운영체제가 추가되어, 실행 가능한 프로그램을 제작하고, 실행하게 된다.

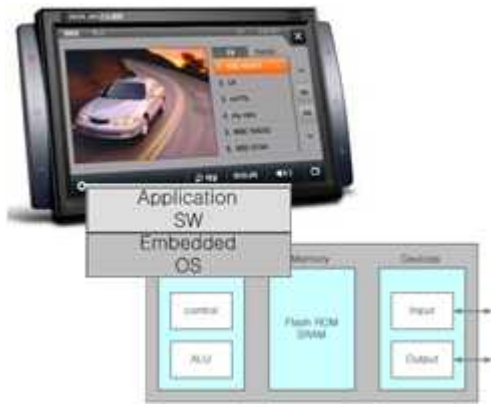


그림 2. OS 기반 정보기기 구조

3. 정보기기 구성

정보단말기는 사용자의 요구에 의해 새로운 기능이 계속 추가되고 있으며, 특히 동일한 입력 장치에서 다양한 기능을 구분하기 위해 시퀀스 지연기능을 사용하고 있다. 이 방법은 하드웨어적인 기술 및 소프트웨어적 기술이 연결된 기능으로, 유사도 감지시 응용 프로그램이나 하드웨어 모듈만의 비교방법에 포함되어야 할 항목이다.

정보기기의 하드웨어는 그림1과 같이 프로세서를 중심으로 하는 입출력부분, 프로그램저장부분, 통신부분으로 구분할 수 있다. 최근에 출시되는 프로세서는 자체적으로 다양한 인터페이스 블록을 내장하고 있기 때문에 프로세서가 입출력 기능을 포함하는 경우가 많아지고 있다. 정보기기 하드웨어의 기능에 따른 유사성 비교는 표준화된 방식에 의한 부분을 제외하고, 응용프로그램에서 작성된 프로그램을 비교하고, 특정 인터페이스 칩이 사용되는 경우를 고려하여 판단하는 것이 타당하다.

응용프로그램은 사용자인터페이스부분으로 사용자에게 편리한 입력기능과 출력기능을 제공

하여, 정보기기 사용자의 기기활용도를 높이는 부분이다. 사용목적에 따라 기기의 하드웨어 부분과 접속되며, 운영체제의 기능을 효과적으로 사용한다.

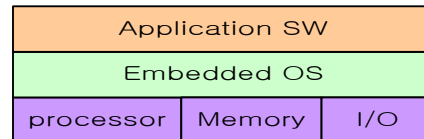


그림 3. Embedded system architecture

정보기기의 기능은 이 응용프로그램의 동작에 의해 결정되는 것으로 제작자는 사용자의 요구에 적합한 응용프로그램은 많이 제작해서 보급하고 있다. 따라서 응용프로그램은 제작자 고유의 기능과 아이디어가 포함된 부분으로 응용프로그램의 기능비교를 통해 기기의 유사성 여부를 도출할 수 있다.

3.1 시스템의 설치 환경

임베디드 시스템이란 특정 프로세서의 메모리에 운영체제(OS)와 운영프로그램(kernel)을 장착하고, 이를 이용하여 특정한 입출력장치(LCD, button)로 동작되는 시스템을 말한다.

이러한 시스템을 구성하기 위해서는 동작에 적당한 프로세서를 선정하고, 그 프로세서에 설치 가능한 운영체제(OS)를 선정한 후, 적절한 방법으로 프로세서의 메모리에 실장을 시키고 동작이 가능하도록 설치하여야 한다.

초기 아무것도 설치되어있지 않은 시스템에 OS와 kernel을 설치하는 과정을 알아보고, 이러한 시스템을 운영하기위한 절차 및 개발방법을 기술한다.

3.2 임베디드시스템의 설치

임베디드 시스템은 사용하는 프로세서와 운영

체제에 따라 설치방법이 상이하기 때문에 여기서는 EMPOS-tiny 보드를 중심으로 설치과정을 알아본다. 이 보드는 intel Xscale PXA255 프로세서를 사용하고 있으며, 유용한 I/O장치(LCD, button, IrDA, GPIO, Ethernet 등)를 가지고 있다. 초기 아무것도 설치되어있지 않은 시스템에 운영체제와 사용자프로그램을 설치하기 위해서는 시스템에 전원이 들어왔을 때 동작이 가능한 부트로더 라는 프로그램을 설치해 주어야 하고, 이 부트로더에 의해 운영체제(OS)가 설치된다.

모든 프로세서는 초기 전원이 들어왔을 때 처음 시작하는 주소에서 프로그램이 시작되도록 만들어지기 때문에, 보통 이 시작주소에 부트로더가 있는 곳으로 이동하는 명령이 위치한다. 부트로더가 실행되면 몇가지 명령어에 의해 자체 실행을 하거나, 포팅할 OS를 메모리의 특정 위치에 저장하고 실행하는 역할을 담당한다.

또한 사용하는 운영체제 및 사용자 프로그램의 개발결과는 시스템에 이식하기 위해서는 시스템과 PC와의 연결이 필요하며, 이를 위해 그림과 같이 PC와 target 시스템 간에 모니터링을 위한 RS232 케이블, 고속전송을 위한 Ethernet 케이블, Flash Rom의 read/write를 위한 JTAG cable 이 연결된다.

이러한 케이블은 시스템의 개발 및 업그레이드를 위한 것이며, 실제 사용을 할 때는 사용되지 않는다. 호스트 PC는 target 시스템의 개발에 사용되는 것으로, 필요한 유틸리티와 개발 툴(tool)이 설치되어야 한다. 이때 타겟 시스템의 OS와 같은 운영체제가 요구되므로, 리눅스 OS가 사용되는 경우는 PC에도 운영체제는 리눅스로 설치되어 있어야 한다.

3.3 호스트측 개발환경

시스템 OS 리눅스의 설치과정을 알아본다.

1) minicom

타겟보드에서 발생하는 진행 상황을 호스트 PC에서 모니터링하는 통신프로그램으로 호스트 PC에 설치되어야 한다.

2) bootp

타겟보드와 네트워킹이 가능하도록 유동적 IP를 할당하는 프로그램을 설치한다.

3) tftp

타겟보드의 부트로더에서 용량이 큰 데이터를 전송받기위해서 사용하는 UDP 전송프로그램으로 Ethernet을 통해 호스트 PC의 파일을 타겟보드로 고속전송이 가능하게 해준다.

4) nfs

리눅스간 파일을 공유할수 있는 파일시스템으로, nfs를 사용하여 호스트 PC에서 개발한 내용을 곧바로 타겟보드에서 테스트가 가능하다.

5) 크로스컴파일러

호스트 PC에서 작성된 소스코드를 실행환경이 다른 타겟보드에서 동작하도록 하기 위해서는 타겟보드에 사용된 프로세서와 일치하는 목적코드(이진코드)로 변환해주는 컴파일러, 링커, 라이브러리 등이 필요하다. 여기서는 Xscale PXA255 (ARM) 프로세서를 사용하고 있으므로, ARM용 크로스 컴파일러를 설치한다.

3.4 부트로더

부트로더(bootloader)란 타겟보드에 전원이 들어오는 순간 실행되는 프로그램으로 일반 PC의 BIOS의 기능과 같다. 각종 하드웨어의 설정을 담당하고, 커널을 RAM으로 이동하는 중요한 역할을 한다.

부트로더의 중요 기능은 다음과 같다.

- 하드웨어 초기화
메모리 초기화, CPU 클럭 지정,
RS232 초기화, MAC 및 Ethernet 초기화
- Flash에서 RAM으로 memory 복사
Kernel image 복사,
ramdisk image 복사,
부트로더 자신 복사
- kernel booting
- command mode 제공
- Debug 함수 제공

또한 부트로더는 자체적용 유용한 명령어를 가지고 시스템을 지원한다. 시스템 분석용으로 사용된 EM-POSTiny 에는 32M의 Flash, 64M의 RAM이 장착되어 있으며, 그림과 같이 Flash에 bootloader, kernel, ramdisk, usr 이미지가 저장되어있다. 전원이 들어오는 순간 bootloader가 동작하면서 모든 image를 RAM으로 이동시키고 RAM에서 실행한다.

bootloader는 JTAG 케이블을 사용하거나 호스트 PC의 tftp명령어를 통해 타겟보드에 설치된다.

메모리 맵에서와 같이 프로세서의 초기동작위치(0번지)에 부트로더가 위치하도록 만들어지고, 전원 연결시 부트로더에 의해 Flash ROM의 내용이 RAM으로 이동되고, kernel을 실행시켜 OS에게 운영권을 넘기게 된다.

Flash 메모리는 ROM으로, 전원이 제거되어도 지워지지 않으며, 때문에 쓰고 읽을 수 있는 RAM으로 이동하여 실행하게 된다.

3.5 커널 구조

전원이 들어오고 bootloader 가 실행되면, Flash Rom에서 RAM으로 모든 이미지가 복사되고, 제어권은 시스템운동을 담당하는 kernel로 넘어간다. 운영체제의 핵심인 kernel은 타겟보드의 RAM에 상주하면서, 시스템에 필요한 환경설정과 수행되는 프로세서들을 스케줄링하는 일을 담당한다.

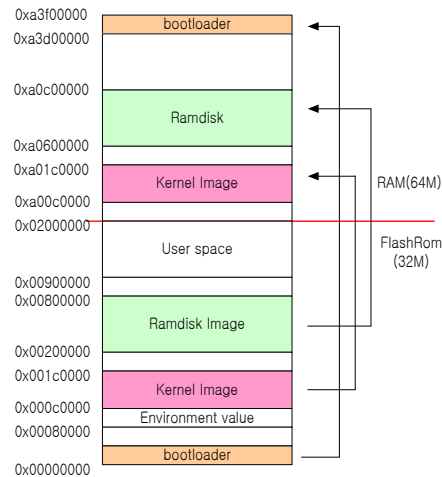


그림 4. 메모리 맵

커널은 마이크로커널(Micro kernel), 모노리틱(Monolithic) 커널로 구분한다. 마이크로커널은 핵심적인 기능만 구현한 최소 커널로 나머지는 서비스 프로세서로 이루어진다. 모노리틱커널은 커널내부에 시스템에 필요한 많은 서비스루틴을 포함한 구조로 구현이 간단하고, 시스템을 효율적으로 관리하는 장점이 있으나, 다양한 환경에 적용하기 어렵고 커널의 크기가 상대적으로 큰 단점이 있다. 마이크로 커널은 핵심기능과 서비스 모듈로 나누어 설계되고, 최소기능만을 커널에 포함함으로써, 기능 확장 및 다른 시스템으로의 재활용이 쉬워지는 장점을 갖는다.

그러나 서버에 메시지 전달방식의 접근은 태스크 스위칭에 많은 오버헤드를 가져와 자원의 효율적 이용이 낮다. Solaris, HP-UX, Linux 등이 모노리틱커널을 사용하고, sunsoft, Digital UNIX 등은 마이크로 커널을 사용한다.

1) 커널의 기능

앞에서 설명한 바와 같이 커널은 하드웨어와 서비스 API의 중간에 위치하며, 다시 그러보면 아래 그림과 같이 사용자 공간과 device 공간 사이에 위치하면서 프로세스관리, 메모리관리, 파일

시스템 관리, 디바이스관리, 네트워크관리 등 5가지의 기능 블록에 해당되는 기능을 수행한다.

- (1) 프로세스 관리
- (2) 메모리관리
- (3) 파일시스템 관리
- (4) 디바이스 관리
- (5) 네트워크관리

2) 커널의 생성

임베디드 시스템에서 linux 커널을 사용하기 위해서는 앞의 메모리맵 그림과 같이 kernel image 공간에 사용할 커널을 삽입해야 한다. 이때 kernel을 올리는 것은 일반 PC에 windowXP를 설치하는 것과 같은 개념으로 우리가 가정에서 가정용 PC에 winXP OS를 설치할 때, OS 디스크의 프로그램이 pentium PC에 적합하도록 작성되었기 때문에 가능한 것이다. 임베디드 시스템의 경우 사용되는 프로세서나, 하드웨어의 구조, 구성된 임베디드 시스템의 특성이 다르기 때문에 일반적으로 작성된 linux OS를 자신의 시스템에 적합하도록 변형시키는 작업을 한다(patch).

여기서 예로 설명하는 EMPOS Tiny 에서는 linux-2.4.19 버전의 커널을 사용하고 있으므로, 커널을 보드에 맞도록 컴파일하고, ARM patch, Xscale patch, EMPOS-Tinyqhem 패치를 각각 적용하고 다운로드 시켜야 사용가능하다.

- kernel 소스확보
- ARM patch 확보 및 적용
- Xscale 패치 확보 및 적용
- EMPOS-Tiny 패치 적용
- 커널 컴파일
- 커널 포팅

커널포팅으로 임베디드 시스템에 사용할 kernel이 Flash ROM 에 설치된다.

3) 파일시스템

파일시스템은 디스크의 파일들이 연속되게 하기위해 사용하는 자료구조라 할 수 있다. 파일을 저장하거나, 파일시스템의 형식을 지정하는데 사용된다. 리눅스에는 NTFS, VFAT, UFS, NFS 등의 파일시스템이 있으며, 일반적으로 사용되는 것으로 ext2가 있다. 2GByte의 파일크기와, 4TByte의 전체파일시스템 구성이 가능하다. 가변 블록크기, 확장기능 등이 제공된다. 디렉토리 구조는 그림과 같고, 각각의 디렉토리는 해당 목적에 맞는 파일들을 보관하고 있다.

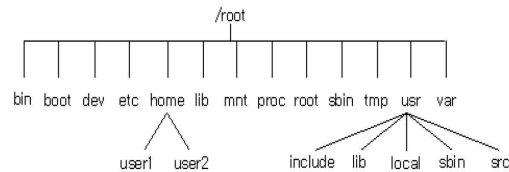


그림 5. 파일시스템 구조

4) Ramdisk

프로세서가 메모리만으로 동작하는 시스템의 경우, 또는 별도의 물리적 저장장치가 없는 경우, 시스템의 메모리 일부를 컴퓨터의 하드디스크처럼 사용하는 구조를 램디스크라고 한다. 임베디드시스템에서는 위에서 설명한 root 파일시스템을 ramdisk에 설정하고, 메모리의 일부를 하드디스크처럼 데이터를 읽고 쓰는 목적으로 활용한다.

그림4의 메모리 맵에서 ramdisk 영역이 이러한 목적으로 설정되고 사용된다.

5) 디바이스 드라이버

디바이스 드라이버는 커널과 하드웨어중간에 위치하여 하드웨어를 추상적으로 개념으로, 커널이 장치와 직접대화하지 않고, 잘 정의된 인터페이스라 할 수 있다.

드라이버는 문자 디바이스 드라이버, 블록 디바이스 드라이버, 네트워크 디바이스 드라이버의 3개로 분류된다. 리눅스에서는 모듈기능을 제공하여 디바이스의 동적 적재, 제거를 가능하게 한다.

4. 커널소스 비교

리눅스 커널은 공개프로그램으로 인터넷으로 다운로드가 가능하다. 본 연구에서 비교대상으로 사용한 리눅스 커널은 그림6과 같이 Linux2.6.28.6 버전으로 하였다.

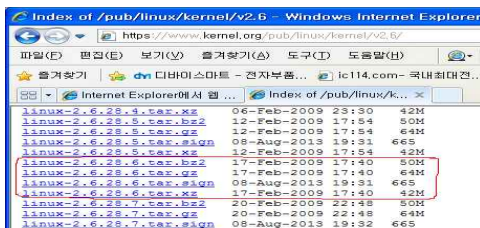


그림 6. 공개 linux kernel(2.6.28.6 버전)

두 개의 시스템이 동일 커널을 다운받아 제품 개발에 적용하였을 경우, 비교대상 커널은 다음 표1과 같이 설정된다. 항목1은 공개소스이고, 항목2는 원저작자의 소스코드, 항목 3은 복제의심이 되는 커널소스를 나타낸다.

표 1. 비교대상 커널 비교표

항목	커널	크기 (bytes)	파일수
1	linux_2.6.28.6	289.0M	26751
2	C-0	400M	26196
3	P-A	1,255M	60379

양측의 커널을 공개프로그램을 기준으로 비교한 결과는 표2에 있다. 이 결과는 복제의 의심이 되는 피저자의 소스코드에서 공개프로그램과 유

사한 부분이 99.3%이고, 결국 대부분의 파일이 공개프로그램이라는 것을 나타낸다.

표 2. 커널의 파일수 비교(공개커널 기준)

비교커널		유사 파일수	유사도(%) 비교본기준
비교본 파일	파일수 (MB)		
원저자 C-0	26196 (400M)	25504	97.3
피저자 P-A	25700 (298M)	25540	99.3

따라서 이결과는 커널소스 전체를 대상으로하는 비교결과에 신뢰성을 의심하게 하는 결과를 준다.

또한 양측의 소스를 직접 비교한 결과는 표3에 제시하였으며, 유사(동일)파일수 비교 99.5%, 파일크기 비교 99.3%로 거의 같은 파일 내용임을 보이고 있다. 즉, 피저자의 커널이 원저자의 커널과 거의 동일함을 나타낸다.

표 3. C-0과 P-A 비교

C-0	P-A	유사파일수 /크기	유사도 /크기 (%)
파일수(크기)	파일수(크기)		
26196(400M)	25700(298M)	25578(296M)	99.5(99.3)

이상과 같이 커널소스의 비교는 공개프로그램이 차지하는 비중이 높아, 전체 커널을 대상으로하는 비교는 유사도 결과의 객관성에 논란의 여지가 있다. 따라서 디바이스 드라이버, 라이브러리, 유틸리티 등의 세부프로그램을 대상으로하는 유사성 비교가 요구된다.

5. 결 론

리눅스 커널소스는 운영체제와 연결된 하드웨

어 블록과의 인터페이스되어 프로그램으로 링크시키는 방법으로 구현된다. 따라서 커널의 비교는 개발환경에서 사용된 하드웨어 구조와 커널 관련 프로그램 소스를 사용하여 유사성을 비교한다. 그러나 많은 경우 개발소스를 제공하지 않는 사례가 있고, 이 경우 zimage 파일을 확보하여 간접적인 비교가 수행되는데, 이 경우에는 감정 결과의 신뢰성을 보장하기 어렵게 된다.

본 연구에서는 양측의 커널소스가 제공되는 상황을 예시하여, 커널소스 전체에 대한 유사성 비교과정을 진행하였다. 연구결과에서 커널소스의 전체비교는 공개프로그램의 비중이 높아 개발자의 소스코드 내용이 유의성을 갖지 못하는 것으로 확인되었다. 따라서 커널소스의 유사성 비교에는 커널소스 전체비교보다는 라이브러리, 디바이스, 유틸리티 등으로 세분화 하여야 원저작권자의 독창적 내용을 판단할 수 있는 것으로 보인다.

참고자료

- [1] 정익래, 홍도원, 정교일; 디지털포렌식 기술 및 동향; 전자통신동향분석 제22권; 2007. 2
- [2] 홍도원; 디지털 포렌식 기술; 한국전자통신연구원; 2007
- [3] 임경수, 박종혁, 이상진; 디지털포렌식 현황과 대응방안; 보안공학연구논문지, 2008. 11
- [4] 류희수; 정보보호: 디지털 세상의 CSI, 그 가능성은?, 정보통신진흥협회, 2007
- [5] 조용현; 디지털 포렌식을 위한 절차와 도구의 중요성; (주)시큐아이닷컴 CERT팀, 2007
- [6] 김도완, 윤영선; SW소스코드 저작권보호를 위한 통합 가이드; 컴퓨터프로그램보호위원회, 2009. 4
- [7] 길연희, 홍도원; 디지털 포렌식 기술과 표준화 동향; IT standard & test TTA journal, 2008, 8
- [8] 방효근, 신동명, 정태명; 소프트웨어 포렌식: 프로그램 소스코드 유사성 비교 및 분석을

중심으로; 디지털 포렌식 연구 창간호, 2007. 11

- [9] 전상덕, 홍동숙, 한기준; 디지털 포렌식의 기술 동향과 전망; 정보화정책; 2006. 11
- [10] 전병태, "프로그램 복제도 감정기법 및 감정비 산출에 관한 연구" 프로그램심의조정위원회 결과보고서 2002.
- [11] 이규대, "임베디드시스템의 이진코드 추출 및 분석", 한국소프트웨어감정평가학회 논문지, 5권1호, pp27-38, 2009.5.
- [12] <https://www.kernel.org/pub/linux/kernel/v2.6/>

저 자 소 개



김인회(In-Hoe Kim)

2000 공주대 정보통신공학과 졸업
2005 공주대 전기전자정보공학과 석사
2013.3 ~ 공주대 정보통신공학과 박사과정
2000.7 ~ 시스템뱅크 부설연구소
<관심분야> 회로 및 시스템, 신호처리, VLC



이규대(Kyu-Tae Lee)

1984 고려대 전자공학과 졸업
1986 고려대 전자공학과 석사
1991 고려대 전자공학과 박사
2001 미 조지아텍 교환 교수
'2007~2009: 한국전자통신연구원 초빙연구원
2006 미 시카고주립대 교환교수
'92. 3 ~현재 : 공주대 정보통신공학부 교수
<관심분야> 회로 및 시스템, 신호처리, VLC