논문 2020-1-5 http://dx.doi.org/10.29056/jsav.2020.06.05

실행코드 비교 감정에서 주변장치 분석의 유효성

김도현*, 이규대**†

Study on the comparison result of Machine code Program

Do-Hyeun Kim*, Kyu-Tae Lee**†

요 약

소프트웨어의 유사성 비교는 소스코드를 대상으로 한다. 소스코드는 프로그램 언어로 표현된 개발자의 지적 저작권으로 보호된다. 문서형식으로 작성된 프로그램 소스코드는 개발자의 전문지식과 아이디어가 포함된 내용을 포함하고 있다. 소프트웨어 저작권의 불법도용을 판단하기 위한 감정 작업은 원본과 비교본의 소스코드를 대상으로 파일의 구성과 내용을 검증하는 방법으로 수행된다. 그러나 실제적으로 피고소인 측의 불성실한 목적물 제공으로 소스코드의 일대일 비교감정이 어려운 상황이 증가하고 있다. 이 경우 실행코드에 대한 비교감정이 수행되어야 하며, 역어셈블 방법, 역공학기법, 기능실행의 시퀀스 분석 등의 간접적인 방법이 적용된다. 본 논문에서는 소스코드제공이 어려운 상황에서 시스템과 실행코드 파일을 대상하는 하는 감정사례를 통해 간접적인 비교결과의 유효성에 대해 분석하고, 감정결과에 활용하는 방안을 제시한다.

Abstract

The similarity of the software is extracted by the verification of comparing with the source code. The source code is the intellectual copyright of the developer written in the programming language. And the source code written in text format contains the contents of the developer's expertise and ideas. The verification for judging the illegal use of software copyright is performed by comparing the structure and contents of files with the source code of the original and the illegal copy. However, there is hard to do the one-to-one comparison in practice. Cause the suspected source code do not submitted Intentionally or unconsciously. It is now increasing practically. In this case, the comparative evaluation with execution code should be performed, and indirect methods such as reverse assembling method, reverse engineering technique, and sequence analysis of function execution are applied. In this paper, we analyzed the effectiveness of indirect comparison results by practical evaluation . It also proposes a method to utilize to the system and executable code files as a verification results.

한글키워드 : 목적코드, 유사도 측정, 가중치 설정, 역컴파일, 저작권보호

keywords : binary code, similarity value, weight factor, reverse compile, copyright

1. 서 론

소프트웨어는 소스코드가 텍스트 형태로 작성되고, 파일로 저장되어 이동이 가능하기 때문에 항상 복제의 가능성이 있는 결과물의 특징을 갖는다. 최근 스마트미디어 장치와 같은 하드웨어 기반 응용 소프트웨어의 개발로 상품성이 높은

* 국립 제주대학교 컴퓨터공학과
** 국립공주대학교 정보통신공학부
† 교신저자: 이규대(email: ktlee@kongju.ac.kr)
접수일자: 2020.06.02. 심사완료: 2020.06.19.
게재확정: 2020.06.19.

제품이 출시되면서, 이와 유사한 제품의 판매로 원저작권자의 영업피해가 발생되고, 이로 인한 저작권 분쟁 사례가 법원과 경찰청을 통해 나타나고 있다.

분쟁이 발생되면 복제 여부를 판단하기 위한 목적물로 분쟁당사자들의 개발에 사용된 프로그램 소스코드가 제공되어야 한다. 그러나 복제의 혹의 피고소인 측은 여러 가지 이유로 원본 소스코드의 제공을 거부하고, 제품에 내장된 실행코드만 제공하여, 소스코드를 대상으로 하는 유사성 비교가 어려운 상황이 증가하고 있다. 한편 원 개발자의 경우도 제3의 업체에 개발의뢰를 하여 제품을 개발하는 사례가 발생하면서, 상대 제품에 대한 복제 의혹을 가지면서도, 자사의 원본 소스코드를 제공하지 못하는 경우도 발생하고 있다.

정보기기는 기본적으로 특정 프로세서를 기반으로 하는 하드웨어 회로보드와 입출력장치에 운영체제가 탑재된 임베디드 시스템를 기반으로 하는 응용소프트웨어가 실행되는 일체형으로 제작되고 있다. 개발자는 제품 목적에 부합되는 하드웨어적인 구성과 기능을 설계하고, 이 구조에 연결되는 입출력 인터페이스 프로그램 및 응용프로그램을 구현하여 하드웨어에 탑재하는 과정으로 하드웨어와 프로그램 소프트웨어를 함께 제작한다.

따라서 분쟁이 발생된 경우 소스프로그램에 의한 일대일 비교 방법으로 유사성을 도출하는 것이 객관적인 분석결과의 의미를 갖게 되지만, 하드웨어적인 구조를 기본으로 하는 프로그램 제작과정의 특성상, 소스프로그램에 부가하여, 하드웨어적인 구조나, 운영의 절차 등으로 복제의 가능성을 추론할 수 있는 방법이 가능하다[1]. 또한 실행코드 만으로 양측의 분쟁내용을 감정하는 경우에는 역공학기법, 기능 실행의 시퀀스 비교, 실행화면의 GUI 등의 비교 방법이 적용 될 수 있다.

본 논문에서는 소스코드제공이 어려운 상황에서 시스템과 실행코드 파일을 대상하는 하는 감정 사례를 통해 간접적인 비교결과의 유효성에 대해 분석하고, 감정결과에 활용하는 방안을 제시한다.

2. 감정목적물의 특징

프로그램 소스코드는 프로세서를 기반으로 제작되는 정보기기를 기반으로 개발되는 것으로 작성되는 응용소프트웨어는 주변장치의 연결과 신호체계에 의존하면서 개발되는 특징을 갖는다. 따라서 소프트웨어의 구조는 하드웨어 주변장치와 밀접하게 연동되도록 작성되기 때문에 프로그램이 동작되는 순서와 데이터의 가공과 결과를 저장하는 체계를 비교하면 직접적인 소스코드의 라인비교를 보완하는 유사성 정보를 얻을 수 있다.

일반적인 소형의 휴대가능한 정보기기의 내부 구성을 보면 그림1과 같다. 주요기능을 담당하는 프로세서를 중심으로 소스코드 프로그램의 저장소로 사용되는 RAM, 응용프로그램 및 운영체제를 보관하는 FlashROM, 정보기기의 입출력용으로 키보드, LCD 터치패드, 그리고 외부 통신이 가능한 통신포트(USB, COM) 등이 연결되어 있다[2].

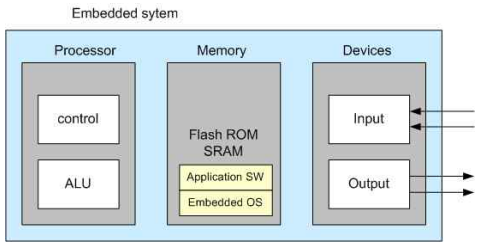


그림 1. 정보기기의 내부 구성
Fig. 1. Configuration of IT device

프로그램 작업은 이와 같은 하드웨어 구조를 완성하고, 이들의 신호체계의 연결 및 동작 순서의 시퀀스를 프로그램 언어로 구현하는 것이다. 이 결과는 텍스트 형태의 소스코드로 표현되고, 시스템의 언어로 변환되는 컴파일 작업을 수행하면, 기계어로 알려진 Hexa코드인 실행코드로 완성된다. 소프트웨어가 운영체제를 기반으로 작성되는 경우에는 시스템 메모리에 운영체제를 탑재하고, 운영체제를 기본으로 하는 응용프로그램을 작성한 후, 실행 가능한 프로그램으로 제작하여 운영된다.

3. 목적물의 간접적 비교항목

정보기기는 하드웨어와 소프트웨어로 구분되고 있으며, 기능의 목적과 성능은 소프트웨어의 실행 상황에 따라 달라진다. 따라서 소프트웨어 프로그램이 기기의 목적을 나타내는 중요한 작업이 된다. 두 기기의 유사성을 판단하는 목적물로 소프트웨어 프로그램을 확보해야하는 이유가 된다. 프로그램은 텍스트 형태의 소스코드로 작성되고, 이때 개발자가 지정하는 변수의 이름, 함수의 흐름도, 프로그램의 실행순서 등이 개발자 고유의 아이디어를 포함하고 있다. 따라서 소스코드의 비교는 정확한 유사도를 추출하는데 기본자료가 된다. 그러나 최근 소스코드의 확보가 어려운 상황의 감정 비교 요구가 발생하고, 한정된 자료만으로 유사성을 도출해야하는 경우가 증가하고 있다[3].

정보기기는 소프트웨어와 하드웨어로 구성되고, 소프트웨어는 하드웨어를 기초로 하여 작성되는 특징이 있기 때문에 소스코드의 부재 시, 하드웨어의 구성도, 동작의 순서도, 주변기기간의 인터페이스 정보 등을 분석함으로써 간접적인 유사성 비교 자료로 활용 가능한 특징이 있다.

3.1 시스템의 간접적 특징항목

1) 임베디드 시스템으로 구성되는 정보기기는 운영체제와 응용프로그램을 구성되고 있으며, 대부분의 운영체제는 공개프로그램으로 사용하고 있다. 또한 교차개발 방식의 개발과정으로 작업하기 때문에 원본 소스프로그램은 개발용 컴퓨터에 저장되고, 시판용 시스템에는 실행코드만 저장되어 유통된다[4]. 따라서 원 개발자가 소스프로그램의 제공을 거부하면, 시판용 시스템에서 실행코드를 추출하여 유사성을 비교해야하는 상황이 된다. 이 경우에는 시스템의 동작순서도, 하드웨어의 구성도, 시스템의 외관 등을 간접정보로 추출하고, 실행코드가 확보되면, 실행코드의 일대일 비교가 불가능하고, 객관성이 부족하기 때문에 ACSII 코드형태의 정보에서 특정 문자열을 검색함으로써, 유사성 비교 정보를 추출한다.

2) 시스템의 주변기기는 프로세서와의 신호교환을 위해 특정 코드를 규정하고, 입출력 데이터를 교환하는 특징이 있다. 이때 사용되는 신호체계는 공개적인 규칙을 사용하는 경우와 개발자의 특정 코드를 사용하는 방식이 있다. 개발자의 특정 코드가 사용되고 있는 정보가 확인되는 경우에는 신호전달 중간단계에서 제어신호를 추출하여, 비교함으로써 도용의 근거자료로 활용이 가능하다[5][6]. 리모컨의 신호방식이나, RS232, TCP/IP 등의 직렬 통신신호체계에서 적용 가능한 방법으로 감정전문가는 개발단계의 전문성을 필요로 한다.

3) 운영체제와 함께 작성되는 프로그램은 공개프로그램으로 제공되는 운영체제를 사용하기 때문에 개발자의 응용프로그램 분량이 공개프로그램에 비교하여 극히 적은 크기로 포함된다, 따라서 전체 커널에 대한 유사성 비교는 90%이상의 유사성으로 나타나기 때문에 개발자의 고유 저작권을 확인하기 어렵다, 이 경우는 커널내의 디바

이스 드라이버 부분을 분석하여, 부분 소스코드에 대한 비교를 수행한다. 특히 특정 문자, 변수 등이 확인되는 경우, 유사성을 증명하는 자료로 활용이 가능한 특징 있다.

3.2 비교도구

유사성 비교에 사용되는 도구는 FlexHEX, exEyes, Beyondcompare,등 여러 가지 응용 프로그램이 상업용으로 활용되고 있다. 라인단위비교, 실행코드 비교, 함수단위 비교 등 목적에 따라 혼합해서 사용한다. 일대일 비교가 불가능한, 특정되지 않는 파일의 경우 다대다(N:N) 비교방법이 가능한 exEyes 비교도구를 사용한다[7][8]. 이 도구는 저작권위원회에서 보유하고 있는 파일간의 유사도 검출을 위해 제작되어, 감정에 적합한 범위를 유사성 비교 기준으로 하고, 유사도를 분석할 수 있는 있다. exEyes 를 사용한 비교수행 결과는 지정된 유사도 이상을 갖는 파일에 대해 세부 분석을 수행한다. 이 방법은 1차적으로 전체 파일에서 가능성이 있는 부분을 확인하는 것으로, 선별된 파일에 대해서만 육안 분석을 수행하여, 유사라인과 동일라인의 유의미성을 정밀하게 분석 판단한다. 분석도구는 유사성판별에서 참고자료를 제고하는 역할이며, 최종적인 유사성을 판단은 라인단위의 육안판독으로 도출하여야 오류를 줄일 수 있다.

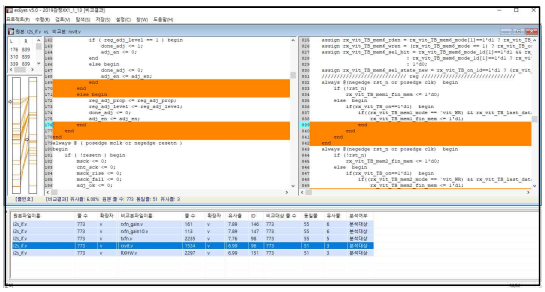


그림 2. exEyes v5.0 실행화면
Fig. 2. exEyes v5.0 window

4. 유사성 분석 사례

4.1 사례1

IPTV(Internet Protocol Television)는 디지털 텔레비전 서비스가 IP 네트워크를 통해 전달되는 시스템으로 정의된다. 인터넷 망의 데이터 속도가 증가하면서 비디오스트리밍을 제공하는 인터넷 프로토콜의 사용으로 IPTV 서비스가 가능해진 것으로, 텔레비전 콘텐츠가 STB(Set Top Box) 의 디스플레이로 방송서비스가 가능하도록 제작된 시스템이다.

임베디드 정보기기는 프로세서를 기본으로 인터페이스 장치와 디바이스드라이버 및 응용프로그램을 포함하는 소프트웨어가 연동되어 운영되는 특징이 있다[12]. 특히 임베디드 시스템과 같이 운영체제가 포함되는 정보기기는 교차개발방법으로 제작되기 때문에 소스코드가 개발용 컴퓨터에 내장되고, 목적물 시스템에는 그림 3과 같이 실행 파일만 저장되어 운영되는 기기로, 주로 셋톱박스, IOT 기능이 부가된 IPTV 등에 적용되고 있다[9][10].

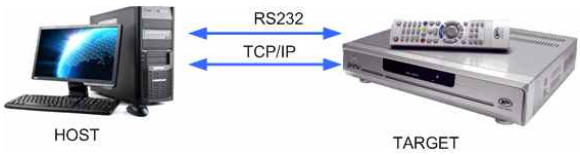


그림 3. 임베디드시스템 교차개발 환경
Fig. 3. Cross development environment

원고의 소스코드제공, 피고의 시스템과 시스템 내부의 실행파일만 제공 경우에는 간접적인 동작 시퀀스, 시스템의 구조적, 회로적 특징 비교가 가능하다.

운영체제의 공개프로그램 사용으로 동작 메뉴의 구조가 유사한 경우의 분석 사례로 개발자의 독자적 개발로 판단할 수 없는 사례를 보인다.

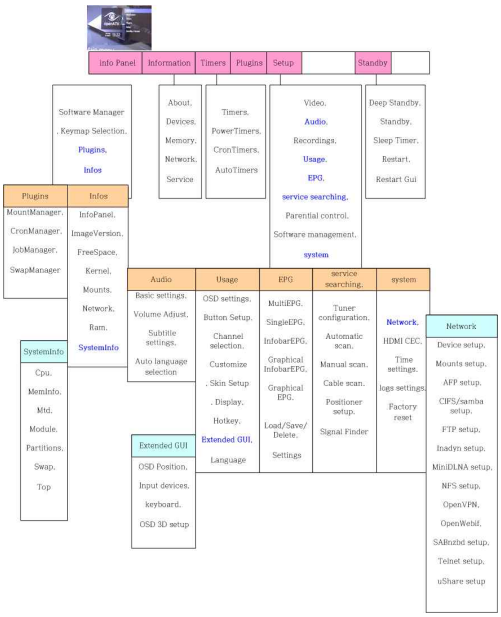


그림 4. 운영체제의 메뉴 구조분석
Fig. 4. OS architecture of menu

실행코드의 ASCII 코드 분석에서 특정 문자(제품의 모델명) 추출에 의한 도용의 의혹이 가능한 분석이 가능하다.

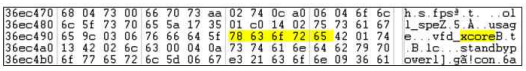


그림 5. 실행코드에 포함된 특정 문자열
Fig. 5. special text of Hexa code

고소인의 도용의혹에 대한 항목을 분석한 결과 동작절차부분은 공개프로그램의 사용으로 분석되었으며, 실행코드에서의 특정 문자열은 신제품이 개발되는 경우 공개프로그램의 규정으로 공지되는 시점에서 포함되는 것으로 확인되어 도용의 근거가 부족한 것으로 판단된 사례이다.

4.2 사례2

감정대상 시스템은 리눅스 운영체제를 기반으로 하는 고객 서비스 프로그램으로 , 디스플레이 화면에 홍보자료를 게시하는 셋톱박스(STB: set-top box)로 구성된다. 시스템은 하드웨어 개발과 소프트웨어로 설계 개발이 복합적으로 적용된 시스템이다. 양측은 소스코드 제출이 없으며, 제작된 시스템을 목적물로 제공한 사례이다.

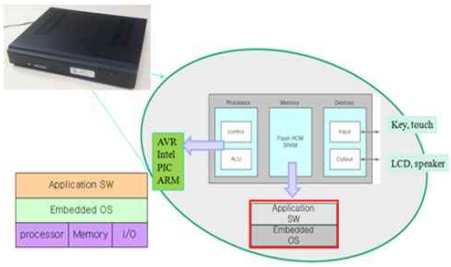


그림 6. 임베디드시스템 구성도
Fig. 6. Embedded architecture of device

두 시스템의 실행코드 추출이 불가하였고, 시스템의 보드와 회로부품 배치 등을 분석 수행하였고, 시스템에 사용되는 리모컨의 동작이 유사하여, 리모컨 신호분석으로 유사성을 도출하고, 유의미한 결과를 제시하였다. 목적물의 리모컨 신호와 일반적으로 사용되는 리모컨의 신호코드가 상이함에도, 분쟁대상 시스템에 동일한 리모컨 신호가 동작되는 부분을 특정하여 유사성의 특징으로 도출한 사례로 소스프로그램이 제공되지 않은 상태에서도 분석 가능한 요소로 활용할 수 있다.

리모컨에서 송수신되는 적외선 신호체계가 혼신을 방지하기 위해 기기마다 다른코드로 조합되도록 제작되는 특징임에도 두 시스템에 동일하게 적용되는 현상을 확인하고, 유사성을 도출한 결과를 보였다. 목적물의 코드는 표1과 같이 분석되었으며, 일반 코드는 표2와 같이 동일 기능에 상이한 코드가 할당되어 있는 비교결과를 예시하였다.

표 1. 목적물3의 리모컨 코드
Table 1. Remote control code of object3

custom	code	function
[08]	[16]	OK
[08]	[10]	power
[08]	[18]	up
[08]	[19]	down
[08]	[1A]	left
[08]	[1B]	right

표 2. 일반 IPTV의 리모컨코드
Table 2. Remote control code of IPTV

custom	code	function
[3915]	[00]	power
[3915]	[11]	up
[3915]	[15]	down
[3915]	[12]	left
[3915]	[14]	right

4.3 사례3

감정대상 시스템은 디지털앰프를 내장한 영업장용 BGM(Back ground musicplayer) STB(set-top box)로 리모컨과 전면패널의 터치센서에 의한 동작제어가 가능하도록 제작되었으며, 내부의 하드웨어 회로는 상업용 프로세서를 사용하였고, 탑재된 소프트웨어는 임베디드방식의 운영체제인 Linux kernel에 응용프로그램을 작성하였다.

운영체제는 방대한 소스코드가 공개되어 있고, 개발자의 프로그램은 일부만 사용되고 있는 것으로 전체 프로그램의 비교는 표3과 같이 독창성을 표현하지 못하고 있다. 원본과 비교본의 커널파일 비교에서 개발자의 독창적인 부분이 전체 커널의 일부로 작성되어, 유사성결과는 동일한 코드로 판단되는 오류가 나타난다.

표 3. 커널의 파일수 비교(공개커널 기준)
Table 3. File count comparison

기준커널		비교커널		유사 화일수	유사도 (%) 비교본 기준
기준	파일수 (MB)	비교본	파일수 (MB)		
A	25617	B1	26196	25504	97.3
		C1	25700	25540	99.3

유사성 비교를 위해 커널에 포함된 디바이스 드라이버 파일의 유사기능을 부분을 추출하고, 선별된 파일의 라인 비교를 수행하여, 코드의 유사성을 분석하였다. 디바이스 드라이버는 하드웨어 주변기기와 밀접하게 관련되는 신호체계를 구현하는 부분으로 하드웨어가 동일하면, 디바이스 드라이버의 코드도 유사성이 높게 작성되는 특징을 활용하는 방법이다. 본 사례에서는 그림과 같이 특정 기기를 나타내는 텍스트가 검출되어 유사성의 근거로 확인한 결과를 보인다.

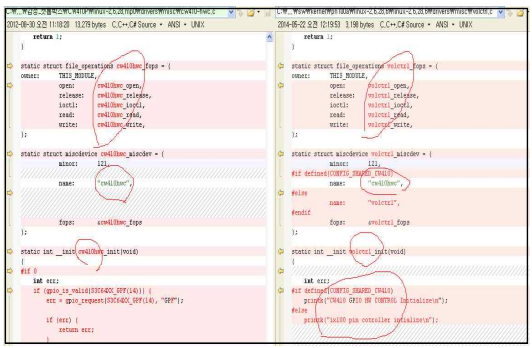


그림 7. 원본과 비교본의 특정 파일이 주석비교
Fig.7. Text finding from similar files

이와 같이 하드웨어 정보기기와 연동되는 프로그램 소스코드의 비교는 장치의 구성과 주변기기의 신호체계를 분석함으로써 실행코드의 비교결과를 보완하는 중요정보로 활용이 가능하다.

5. 결 론

스마트미디어 기기의 복제의혹에 따른 분쟁이 발생되면서, 복제 여부를 판단하기 위한 목적물로 개발에 사용된 프로그램 소스코드가 제공되어야 하나, 복제의혹의 대상자는 원본 소스코드의 제공을 거부하고, 제품에 내장된 실행코드만 제공하게 되어, 소스코드를 대상으로 하는 유사성 비교가 어려운 경우가 발생하고 있다. 정보기기는 구조적으로 특정 프로세서를 기반으로 하는 하드웨어 회로보드와 입출력장치에 운영체제가 탑재된 임베디드 시스템를 기반으로 하는 응용소프트웨어가 실행되는 일체형으로 제작되고 있다. 따라서 분쟁이 발생된 경우 소스프로그램에 의한 일대일 비교 방법이 불가능 한 경우, 하드웨어적인 구조를 기본으로 하는 프로그램 제작 과정의 특성상, 실행코드와 하드웨어적인 구조, 운영의 절차 등으로 복제의 추론이 가능하다. 본 연구결과에서는 실제적인 감정 사례를 통해 정보기기의 분석 및 유사성 도출 결과를 제시하였다. 이와 같이 개발과정에서 필연적으로 삽입되어야 하는 기능 외에 개발자의 전문 기술이 적용된 특징을 도출하고, 이와 유사하거나 동일한 부분이 다른 기기에서 추출되는 경우, 복제의 의심이 가능한 유의미한 결과의 활용 가능성을 제시하였다. 최근 지능적인 복제시도가 다양하게 적용되는 환경에서 감정전문가는 개발자의 입장에서 분석 및 유사성을 도출하는 과정으로 원 개발자의 저작권 보호에 활용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] Raj Kamal. Embedded systems Architecture Programming and Design, 2nd ed. MacGraw Hill Companies, p.5, 2015. ISBN: 0-07-049470-3

[2] http://blogspot.designonchip.com/2009/10/rtl-engineer.html, Oct. 2009.

[3] Kyu-Tae Lee, Hyun-Chang Lee, Jang-Geun Ki, “Establishment of the Subtitle on Materials for Evaluating Intellectual Ownership”, International Journal of Signal Processing, Image Processing and Pattern Recognition, vol.10, no.9, pp.79-88, Sep., 2017. http//dx.doi.org/10.14257/ijsip.2017.10.9.09

[4] M. M. Swift, B. N. Bershad, and H. M. Levy, “Improving the Reliability of Commodity Operating Systems”, ACM Trans. on Computer Systems, vol.23, no.1, pp.77-110, Sep., 2003. DOI: 10.1.1.107.2596

[5] M. M. Swift, M. Annamalai, B. N. Bershad, and H. M. Levy, “Recovering Device Drivers”, ACM Trans. on Computer Systems, vol.24, no.4, pp.333-360, Nov., 2006. http://u.cs.biu.ac.il/~wiseman/2os/bugs/swift1.pdf

[6] M. Rajagopalan, M. A. Hiltunen, T. Jim, and R. D. Schlichting, “System Call Monitoring Using Authenticated System Calls”, IEEE Trans. on Dependable and Secure Computing, pp.216-229, July 2006. DOI: 10.1109/TDSC.2006.41

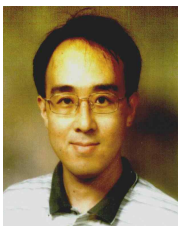
[7] T. Naughton, W. Bland, G. Vallee, C. Engelmann, and S. L. Scott, “Fault Injection Framework for system Resilience Evaluation”, Proc. of the Resilience‘09. pp.23-28, June, 2009. https://www.christian-engelmann.info/publications/naughton09fault.pdf

[8] http://www.scootersoftware.com/features.php?zz=features_focused, Oct. 10, 2018.

[9] V.J. Mooney, D.M. Blough, “A hardware-software real-time operating system framework for SoCs”, IEEE Design & Test of Computers, vol.19, no.6, pp.44-51, Nov., 2002. DOI: 10.1109/MDT.2002.1047743

[10] Do-Hyeun Kim, KyuTae Lee, “Management of Reliability and Delivery for Software Object Material”, Journal of Software Assessment and Valuation (JSAV), vol.15, no.2, pp.51-57, Dec, 2019. http://dx.doi.org/10.29056/jsav.2019.12.07

저 자 소 개



김도현(Do-Hyeun Kim)

2000.8 경북대 전자공학과(정보통신전공) 박사
2004.9～현재 국립 제주대학교 공과대학 컴퓨터공학전공 교수.
<주관심분야> 사물인터넷, 예측 및 최적 제어, 모바일 컴퓨팅, 임베디드 소프트웨어

이규대(Kyu-Tae Lee)

1991 고려대 전자공학과 박사
1992～현재 국립공주대학교 정보통신공학부 교수
<주관심분야> 신호처리, VLC, 저작권보호, 임베디드 시스템, 상황인식 및 학습