

논문 2012-07-17

# VxWorks 기반 소프트웨어를 위한 원격 테스트 도구의 설계

(A Design of the Remote Test Tool for Software  
Based on VxWorks)

박 송 화\*, 이 형 수

(Song-Hwa Park, Hyung-Soo Lee)

**Abstract** : The Quality of embedded system depends on the embedded software. As the complexity and the size of embedded software have been increasing, it is more likely that the software may include faults, and the reliability and stability issues are getting more important. In this paper, we propose a remote test tool for software based on VxWorks by using fault injection method. The test tool consists of test server and test client for testing on the cross development environment. The test server operates in the host system and user can not only test but also monitor the software by using it. The test client operates in the target system and it controls kernel objects and sends the input data into the software when receiving the control and data from the test server. We developed the prototype software and demonstrated the ability of testing software robustness by injecting faults.

**Keywords** : Software test, Remote test, Fault injection, VxWorks

## 1. 서론

임베디드 소프트웨어(Embedded software)는 모바일 단말, 디지털 가전기기, 자동차, 첨단무기, 산업용 기기 및 로봇 등의 제품에 내장되어 하드웨어를 제어하고 사용자에게 향상된 서비스 기능을 제공하기 위한 소프트웨어이다.

최근 임베디드 소프트웨어에서의 주요 이슈는 하드웨어의 성능 및 전력 소모 증대에 따른 고성능 컴퓨팅 파워, 소형화, 경량화이다. 또한 제품 내 소프트웨어 비중이 증가함에 따라 임베디드 소프트웨어의 기술 수준이 제품의 신뢰성 및 안정성과 직결되는 양상을 보이고 있다 [1,2].

소프트웨어는 다양한 서비스 산업들과 융합하여 산업 전반에 고부가 가치의 새로운 비즈니스 모델을 확산시키는 중요한 역할을 담당하고 있다. 이에 따라 소프트웨어 결함에 기인한 경제적 손실과

위험도 증가하고 있는 실정이다. 표 1은 소프트웨어 결함으로 인한 일부 산업 분야의 대표적인 피해 사례를 정리한 내용이다 [3].

소프트웨어 테스트의 중요성은 점차 증가되고 있으나, 일반적으로 개발자들은 빠른 시일 내에 제품을 개발해야 하기 때문에 소프트웨어 테스트에는 시간과 관심을 크게 두지 못하는 것이 현실이다. 제품의 품질 확보를 위하여 개발 프로세스 후반의 테스트는 전문 테스터에 의해 체계적으로 수행되고 있으나 단위 테스트, 통합 테스트와 같은 개발 단계에서의 테스트는 내부 개발자의 책임 하에 묵시적으로 수행되고 있는 경우가 많다 [4,5]. 개발 프로세스 후반의 테스트는 소프트웨어를 블랙박스로 간주한 테스트가 가능하기 때문에 테스트 기술을 전문적으로 훈련 받은 인력에 의한 체계화된 테스트가 가능하다. 그러나 프로세스 전반 테스트는 프로그램 단위를 한 번에 하나씩 또는 합쳐서 수행하는 것으로 프로그램 내부 구조에 대한 상세한 지식을 요구하기 때문에 개발 담당자가 직접 수행할 수 밖에 없다. 상대적으로 테스트 기술이 미흡한 개발자의 테스트는 발견된 결함을 디버깅하거나 시스템이

\* 교신저자(Corresponding Author)

논문접수 : 2012. 01. 28., 수정일 : 2012. 02.24.,

채택확정 : 2012. 04. 03.

박송화, 이형수 : LIG 넥스원

동작함을 증명하는 수준에 머무르고 있다. 결함 발견이 늦어질수록 수리 및 복구 비용이 증가하므로 개발자의 테스트를 강화하는 것은 중요한 일이다. 그림 1과 같이 개발 완료 이후 발견된 오류의 수정 비용은 개발 중 오류 수정비용의 2배가 넘는 것으로 알려져 있다 [6,7].

표 1. 소프트웨어 결함으로 인한 피해 사례  
Table 1. Case of damage caused by software defects

기업명	소프트웨어 결함으로 인한 피해 사례	시기
BMW	연료 인젝션 펌프 리콜	2002
현대자동차	뉴EF 소나타 앞쪽 측면 좌우 에어백 임의 작동	2002
팬택엔 큐리텔	다수 모델에서 소프트웨어 결함 발생	2003
포드	Escape 파워트레인 엔진 정지 유발로 인한 2001 ~ 2003년 생산량 중 363,440대 리콜	2004
한국은행	운용 프로그램 내 소프트웨어 간 충돌로 5시간 동안 은행 업무 장애	2004
국내 이동통신사	소프트웨어 결함으로 인한 통신사 간 번호이동 전산사고	2004
현대자동차	미국 판매 아반떼 12만대 리콜, 에어백시스템 소프트웨어 결함	2005
아우디	국내 판매된 아우디 A6 728대 감속 센서 등 소프트웨어 결함 리콜	2005
서울시	신교통카드시스템 오동작으로 인한 교통대란	2004
동경 증권 거래소	미즈호증권의 매도주문 오발주	2004
	소프트웨어 패치처리 문제로 인한 증권거래 정지	2006
우리은행	전산장애로 인한 창구 업무 전면 중단	2006
토요타	2010년 산 프리우스 하이브리드 43만여대 리콜	2010
한국항공 우주연구원	나로호 우주 발사 실패	2010

본 논문에서는 소프트웨어 개발 단계에서 소프트웨어의 단위 테스트를 통하여 소프트웨어의 신뢰성을 향상시킬 수 있는 테스트 도구를 제안한다. 제안하는 테스트 도구는 VxWorks 기반의 소프트웨어를 실제 보드에 탑재하여 동작시키고 개발자가 원격으로 테스트할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 테스트 대상으로 하고 있는 VxWorks, VxWorks 기반의 소프트웨어 개발 및 디버깅을 위한 교차 개발 환경과 테스트 도구에 사용된 결함 주입 기법에 대하여 간략하게 소개하고 3장에서 본 논문에서 제안하고자 하는 VxWorks 기반 소프트웨어를 위한 원격 테스트 도구를 기술한다. 그리고 4장에서는 본 논문에서 제안한 테스트 도구를 검증한 실험 결과를 제시하고 5장에서 결론으로 본 논문을 맺는다.

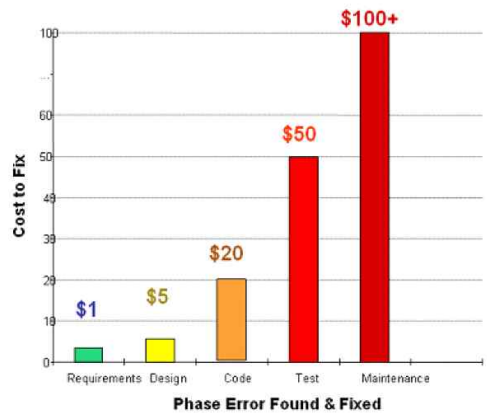


그림 1. 소프트웨어 개발 주기 내 수정 비용 [8]

Fig. 1. Fix cost during the software development cycle [8]

## II. 관련 연구

### 1. VxWorks

VxWorks는 WindRiver 사에서 개발한 실시간 운영체제(RTOS: Real-time operating system)이다. VxWorks의 커널인 마이크로 커널은 선점형 멀티태스킹(multi-tasking)이며 총 태스크(task)의 단계는 256개이고 스케줄링 방식도 높은 우선순위를 가지는 태스크가 먼저 실행하는 방식을 지원한다. 같은 우선순위를 가지는 경우, 라운드로빈(round-robin) 방식의 스케줄링을 이용한다. VxWorks는 200 개 가량의 모듈을 지원하는 형식으로 되어 있어 개발자들은 필요한 모듈만 사용하여 시스템에 맞는 운영체제를 구성할 수 있다. 그 외 다음과 같은 장점으로 인하여 많은 임베디드 시스템에서 사용되고 있다.

- 1) 선점형 스케줄러 기반의 빠른 멀티태스킹 커널
- 2) 확장된 태스크 간 통신 및 동기화 방법 지원
- 3) 효율적인 메모리 관리 방식 지원
- 4) 멀티 프로세서 지원

VxWorks에서 실행 단위는 태스크이며 기능을 수행하기 위해서 태스크 간 통신(ITC: Inter-Task Communication)을 수행한다. VxWorks에서 제공하는 태스크 간 통신 방법은 다음과 같다[9].

- 1) 이벤트(Event)
- 2) 메시지 큐(Message queue)
- 3) 메시지 채널(Message channel)
- 4) 파이프(Pipe)
- 5) 공유 메모리(Shared memory)
- 6) 소켓(Socket)과 원격 프로시저 호출(RPC)
- 7) 신호(Signal)
- 8) 세마포어(Semaphore)
- 9) 뮤텍스(Mutex)와 조건 변수(Condition variables)

## 2. 교차 개발 환경(Cross-tool chain)

실제 소프트웨어가 수행 될 시스템과 개발하는 시스템이 다른 개발 환경, 즉 호스트 시스템(host system)을 이용해서 다른 프로세서가 있는 타겟 시스템(target system)에서 동작하는 소프트웨어를 생성하기 위하여 작업하는 환경을 교차 개발 환경이라고 한다.

일반 소프트웨어의 개발과 같이 임베디드 소프트웨어 개발도 컴파일러, 링커, 인터프리터, 통합 개발환경 그리고 이와 관련된 개발 툴을 사용한다. 임베디드 시스템의 경우, 대부분 특수 목적을 수행하도록 제작되거나 경량화 및 소형화되어 제작되므로 직접적인 소프트웨어 개발환경을 보유하기엔 하드웨어적인 자원이 제한되어 있다. 따라서 임베디드 시스템만을 사용하여 소프트웨어를 개발하는 것은 거의 불가능하다[10].

교차 개발 환경은 표 2와 같이 호스트 시스템, 타겟 시스템 및 호스트 시스템과 타겟 시스템 간의 통신을 위한 요소로 구성된다.

## 3. 결함 주입 기법

결함 주입에 대한 개념은 초창기 하드웨어의 안정성 및 신뢰성을 점검하기 위한 테스트 방법으로 사용되었고, 1990년 Unix 운영체제에 대한 검증 도구인 FUZZ가 발표되면서 소프트웨어 테스트 방법으로 사용되고 있다.

표 2. 교차 개발 환경 구성 요소

Table 2. Component of cross-tool chain

요소	특징
호스트 시스템	* 하드웨어 시스템을 제작한 후, 해당 시스템에서 구동 될 소프트웨어를 개발하는 시스템 * 교차 컴파일러(Cross compiler) - 호스트 시스템에서 동작하나 타겟 시스템의 CPU에 의존적인 코드 생성 * 교차 디버거(Cross debugger) - 타겟 시스템에서 수행 중인 소프트웨어를 호스트 시스템에서 관찰 및 디버깅이 가능하도록 해 줌
타겟 시스템	* 개발 대상인 보드나 하드웨어로 독립된 동작이 되지 않는 시스템 * 독립된 동작을 위해서는 커널, 디바이스 드라이버, 응용 프로그램 등이 호스트 시스템을 통해서 개발된 뒤에 다운로드나 장입되어야 함
통신을 위한 요소	* 시리얼 케이블(Serial cable), 이더넷 케이블(Ethernet cable) - 호스트 시스템과 타겟 시스템의 정보 교환 및 상태 파악을 위한 연결에 이용 * 터미널(Terminal) - 타겟 시스템의 셸이나 동작 상태를 전시

결함 주입 기법은 소프트웨어 테스트 중 블랙박스 테스트의 일종으로 주로 COTS(Commercial Off-The-Shelf)를 대상으로 프로그램의 내부 구조 및 코드에 대한 사전 지식 없이 테스트를 수행하는 방법이다. 결함 주입 기법은 소프트웨어 시스템의 외부에 노출된 API(Application Programming Interface), 네트워크, 파일, 사용자 인터페이스(User Interface) 등에 대하여 프로그램이 예상하지 못한 다양한 값을 입력하여 소프트웨어의 취약점을 점검하는 방법이다[11, 12].

## III. 테스트 도구의 설계

### 1. 테스트 방법

테스트 방법은 여러 가지 기준에 따라 다양하게 분류되며, 실제 테스트에서는 여러 가지 방법이 혼합되어 함께 사용된다.

본 논문에서는 프로그램 내부 구조 및 코드에 대한 사전 지식 없이 테스트를 수행하기 위하여 결함 주입 기법을 사용한다. 결함을 주입하는 대상은 타겟 시스템의 커널 오브젝트와 테스트 대상 소프

트웨어의 API이다.

타겟 시스템에서 생성되는 커널 오브젝트 정보를 관리하고 사용자로부터 제어를 수신하여 커널 오브젝트의 삭제, 태스크 우선순위 변경 등의 방법을 사용하여 커널 오브젝트에 결함을 발생시키도록 한다. 또한 소프트웨어 API의 입력 데이터에 결함을 삽입하여 소프트웨어의 동작을 확인하도록 한다.

### 2. 테스트 도구의 설계

제안하는 원격 테스트 기법은 그림 2와 같이 테스트 서버와 테스트 클라이언트로 구성된다.

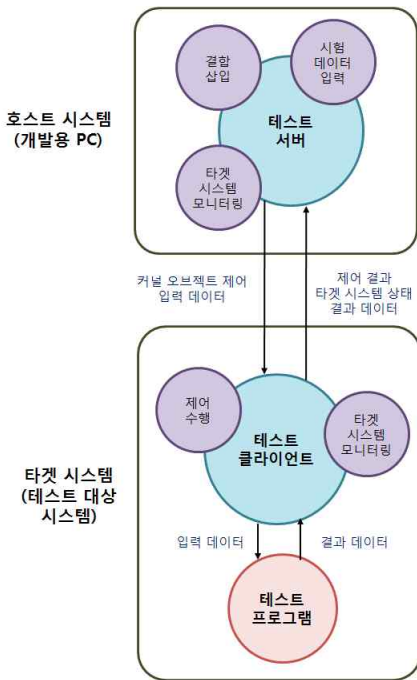


그림 2. 테스트 도구의 구조

Fig. 2. The structure of the test tool

#### 2.1. 테스트 서버

테스트 서버는 사용자로부터 커널 오브젝트 제어 또는 입력 데이터 송신 제어를 수신하여 테스트 대상 소프트웨어로 전송하는 역할을 수행한다. 또한 테스트 대상 소프트웨어의 상태 및 결과를 사용자 인터페이스를 통하여 사용자에게 제공한다.

커널 오브젝트에 결함을 주입하기 위하여 태스크 삭제, 태스크 우선순위 변경 및 메시지 큐 삭제 등의 방법을 사용한다. 커널 오브젝트에 결함을 발생시키기 위하여 테스트 서버는 테스트 클라이언트로부터

타겟 시스템의 커널 오브젝트 정보를 주기적으로 수신하여 관리한다. 최신 커널 오브젝트 정보를 사용자에게 전시하고, 사용자는 특정 커널 오브젝트를 선택하여 제어할 수 있다. 사용자가 커널 오브젝트를 제어할 경우, 테스트 서버는 제어 정보를 테스트 클라이언트로 송신한다.

테스트 서버는 커널 오브젝트 외에 API에도 결함을 주입할 수 있다. API로 입력되는 데이터 정보와 결함 삽입 여부를 사용자로부터 입력받고 데이터를 테스트 클라이언트로 송신한다. 이때, 사용자가 결함 포함을 선택한 경우에 입력 데이터 임의의 부분을 변경하여 송신하고 테스트 클라이언트는 수신한 데이터를 테스트 대상 소프트웨어로 송신한다.

#### 2.2. 테스트 클라이언트

테스트 클라이언트는 타겟 시스템에서 동작하며 테스트 서버로부터 제어 및 입력 데이터를 수신하여 테스트 대상 소프트웨어로 송신한다. 테스트 대상 소프트웨어로부터 결과를 수신하고 테스트 서버로 송신하며 주기적으로 타겟 시스템의 상태를 모니터링하여 테스트 서버로 송신한다. 커널 오브젝트 제어 및 타겟 시스템 상태 모니터링은 VxWorks 커널에서 제공하는 함수를 사용하여 수행할 수 있다.

#### 2.3. 테스트 서버와 테스트 클라이언트 간 연동

테스트 서버는 호스트 시스템에서 구동되며 테스트 클라이언트는 타겟 시스템에서 구동된다. 상이한 플랫폼에서 동작되는 두 소프트웨어 간 통신을 위하여 소켓 통신을 사용한다.

테스트 서버에서 TCP 서버가 구동되고 테스트 클라이언트에서 TCP 클라이언트가 구동된다. 소켓 통신을 사용한 테스트 서버와 테스트 클라이언트 간 연동 정보는 표 3과 같다.

### 3. 테스트 도구의 동작

본 논문에서 제안한 테스트 도구의 동작 절차는 그림 3과 같다.

- 1) 호스트 시스템과 타겟 시스템이 부팅된다.
- 2) 호스트 시스템에서 테스트 서버가 실행된다.
- 3) 타겟 시스템에서 테스트 대상 소프트웨어와 테스트 클라이언트가 실행된다.
- 4) 테스트 클라이언트가 테스트 서버로 접속한다.
- 5) 테스트 클라이언트가 타겟 시스템의 정보(커널 오브젝트 정보, 테스트 소프트웨어 상태 등)를 검색하고 테스트 서버로 송신한다.

표 3. 테스트 서버와 테스트 클라이언트 간 연동 정보  
Table 3. Communication information between test server and test client

방향	주기성	연동 정보
테스트 서버 → 테스트 클라이언트	비주기	커널 오브젝트 제어
	주기/비주기	입력 데이터
테스트 클라이언트 → 테스트 서버	비주기	제어 결과
	주기	타겟 시스템 상태 (커널 오브젝트 상태, 테스트 대상 소프트웨어 상태)
	주기/비주기	결과 데이터

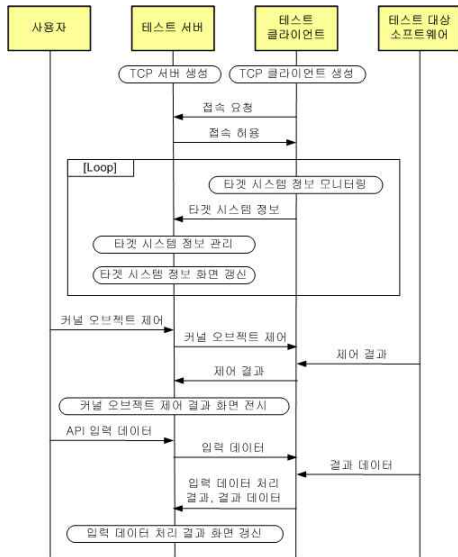


그림 3. 테스트 도구의 동작  
Fig. 3. Behavior of the test tool

- 테스트 서버는 운용자로부터 커널 오브젝트 제어 또는 입력 데이터 송신 제어를 수신한다.
- 테스트 서버는 수신한 제어를 테스트 클라이언트로 송신한다.
- 테스트 클라이언트는 테스트 서버로부터 수신한 제어를 사용하여 커널 오브젝트 제어 또는 입력 데이터 송신을 수행한다.
- 테스트 클라이언트는 제어 결과 또는 결과 데이터를 테스트 서버로 송신한다.

10) 테스트 서버는 테스트 결과를 사용자에게 전시한다.

사용자가 테스트 서버에서 테스트 동작을 선택하면 타겟 시스템에서 구동되는 테스트 클라이언트가 테스트 동작을 수행한다. 테스트 클라이언트가 테스트 결과를 테스트 서버로 송신하고 테스트 서버는 테스트 결과를 화면을 통하여 사용자에게 전시한다.

#### IV. 실험

##### 1. 실험 환경

본 논문에서 설계한 테스트 도구의 설계를 검증하기 위하여 프로토타입 소프트웨어를 구현하여 실험하였다. 실험 환경은 그림 4와 같으며 테스트 서버는 Windows XP에서 Visual Studio 2008을 사용하여 개발하였고 테스트 클라이언트는 VxWorks 6.4에서 WorkBench 2.6을 사용하여 개발하였다. 타겟 시스템으로 사용된 보드는 G4DSPXD3-24B8 보드이며 사양은 표 4와 같다.

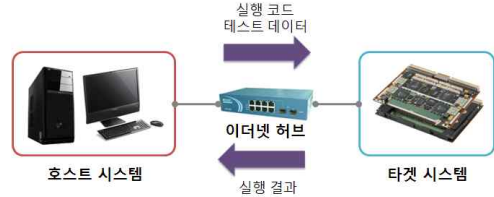


그림 4. 실험 환경  
Fig. 4. Experimental environment

##### 2. 테스트 방법

본 논문에서 제안한 테스트 도구는 커널 오브젝트와 입력 데이터에 결함을 삽입하여 소프트웨어의 동작을 테스트하는 것을 목표로 한다. 커널 오브젝트에 결함을 삽입하여 테스트가 가능한지 검증하기 위하여 커널 오브젝트 중 태스크와 메시지 큐에 대하여 테스트 삭제 및 메시지 큐 삭제 제어가 가능하도록 구현하였다.

테스트 서버는 그림 5와 같이 사용자로부터 테스트를 위한 입력 데이터 파일을 선택할 수 있도록 하였으며 사용자가 선택한 파일을 읽어 테스트 클라이언트로 송신하도록 하였다. 사용자가 결함을 포함하도록 제어하는 경우, 파일에서 읽은 테스트 데이터의 임의의 부분을 변경한 후 송신하도록 하였다.

표 4. G4DSPXD3-24B8의 사양

Table 4. G4DSPXD3-24B8 Specification

구분	사양
제조사	GEIP
프로세서	1GHz MPC7448 4개
메모리	* 512 MBytes (DDR SDRAM) * 32 MBytes (Flash memory per CN)
연동사양	* 4 x 10/100/1000 BASE-T Ethernet * 4 x RS-232 ports * 2 x StarFabric ports

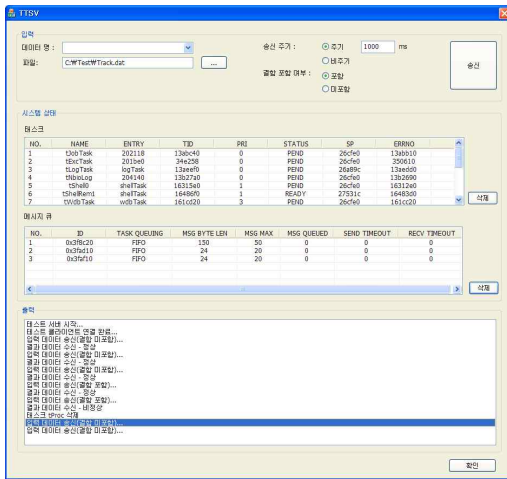


그림 5. 테스트 서버 화면  
Fig. 5. Test server screen

테스트 클라이언트는 테스트 서버로부터 제어를 수신하고 5 초 주기로 타겟 시스템의 상태를 모니터링하도록 하였다. 테스트 서버로부터 태스크 삭제 제어 수신 시, td() 함수를 사용하여 태스크를 삭제하고, 메시지 큐 삭제 제어 수신 시, msgQDelete() 함수를 사용하여 해당 메시지를 삭제하도록 구현하였다. td() 함수와 msgQDelete() 함수는 VxWorks에서 제공하는 함수이다.

테스트 대상 소프트웨어는 테스트 클라이언트로부터 입력 데이터를 수신하여 처리하고 결과 데이터를 송신하도록 하였다. 테스트 대상 소프트웨어는 5 개의 태스크와 3 개의 메시지 큐를 사용하여 구현하였으며 태스크와 메시지 큐의 역할 및 용도는 각각 표 5, 표 6과 같다.

표 5. 테스트 소프트웨어를 구성하는 태스크

Table 5. Task of configuring the test software

태스크 이름	역할
tRecv	소켓 통신을 통하여 입력 데이터 수신
tSend	소켓 통신을 통하여 결과 데이터 송신
tInc	tRecv 태스크로부터 입력 데이터를 수신하여 입력 데이터의 시간 정보를 1 초씩 증가시키고 그 결과를 tSend 태스크를 통하여 송신
tDec	tRecv 태스크로부터 입력 데이터를 수신하여 입력 데이터의 시간 정보를 1 초씩 감소시키고 그 결과를 tSend 태스크를 통하여 송신
tTimer	5 초마다 카운트 값을 증가시키고 logMsg() 함수를 사용하여 화면에 전시

표 6. 테스트 소프트웨어가 사용하는 메시지 큐  
Table 6. Message queue of configuring the test software

메시지 큐 이름	용도
msgQInc	tInc 태스크의 메시지 수신 용 메시지 큐
msgQDec	tDec 태스크의 메시지 수신 용 메시지 큐
msgQSend	tSend 태스크의 메시지 수신 용 메시지 큐

## 2. 실험 결과

### 2.1. 커널 오브젝트 결함 삽입

실험을 통하여 커널 오브젝트에 결함을 삽입하여 소프트웨어 테스트가 가능한 것을 확인하였다. 태스크 삭제 및 메시지 큐 삭제 결과는 각각 표 7, 표 8과 같다. 태스크 삭제 실험에서 사용된 tLogTask 태스크는 VxWorks 커널에서 시스템 부팅 시 자동으로 생성되는 태스크로 다른 태스크가 logMsg() 함수를 호출할 경우, 해당 메시지를 화면에 출력하는 기능을 수행한다. tLogTask 태스크를 삭제하는 경우, 테스트 대상 소프트웨어가 logMsg() 함수를 사용하더라도 로그 메시지만 출력되지 않고 그 외 기능은 정상 동작하는 것을 확인하였다.

표 7. 태스크 삭제 테스트 결과

Table 7. Test result of the task deletion

삭제 태스크	테스트 결과		
	데이터 수신	데이터 처리	데이터 송신
tRecv	수신 불가	-	-
tInc	수신	처리 불가	-
tDec	수신	처리 불가	-
tTimer	-	처리 불가	-
tSend	수신	처리	송신 불가
tLogTask	수신	tTimer: 처리 불가 그 외: 처리	송신

표 8. 메시지 큐 삭제 테스트 결과

Table 8. Test result of the message queue deletion

삭제 태스크	테스트 결과		
	데이터 수신	데이터 처리	데이터 송신
msgQProc	수신	tTimer: 처리 그 외: 처리 불가	-
msgQSend	수신	처리	송신 불가

본 실험에서 사용한 태스크 삭제와 메시지 큐 삭제 제어 외에도 VxWorks 커널에서 제공하는 함수를 사용하여 커널 오브젝트를 제어하여 커널 오브젝트에 결함을 삽입할 수 있을 것이다.

2.2 입력 데이터 결함 삽입

입력 데이터에 임의로 데이터를 변경하여 결함을 삽입함으로써 소프트웨어 테스트가 가능한 것을 확인하였다. 실험에 사용된 테스트 대상 소프트웨어에서 tInc와 tDec 태스크의 경우, 입력 데이터의 시간 정보를 1 초 증가시키는 기능을 수행하므로 시간 정보에 잘못된 값이 입력되는 경우에 오동작 하였으나 그 외의 경우에는 정상 동작하였다.

실험 결과를 통하여 입력 데이터에서 기능 수행을 위하여 무결성이 보장되어야 하는 데이터 요소에 결함을 삽입하여 소프트웨어의 강건성을 시험할 수 있다는 것을 확인하였다.

V. 결론

본 논문에서는 VxWorks 기반의 소프트웨어를 위한 원격 테스트 기법을 제안하였다. 제안한 테스트 도구는 교차 개발 환경에서 소프트웨어를 테스트하기 위하여 각각 호스트 시스템과 타겟 시스템에서 구동되는 테스트 서버와 테스트 클라이언트로 구성되며 타겟 시스템에 커널 오브젝트 제어 또는 입력 데이터를 송신하여 소프트웨어의 동작을 확인할 수 있도록 설계하였다. 제안한 테스트 도구의 프로토타입 프로그램을 구현하여 소프트웨어의 동작이 시험 가능함을 확인하였다.

개발 단계에서의 소프트웨어 테스트는 개발자에게 귀찮지만 소프트웨어 수리 및 복구 비용을 감소시키며 소프트웨어의 결함을 줄이기 위해서는 중요한 활동이다. 소프트웨어의 안정성과 신뢰도를 증가시키고 소프트웨어 결함에 의한 수리 비용을 감소시키기 위해 VxWorks 외에도 다양한 환경에서의 임베디드 시스템을 위한 소프트웨어 테스트 도구의 지속적인 연구 및 개발이 요구된다.

참고문헌

- [1] 소프트웨어 산업백서 2010, 정보통신산업진흥원, 2010.
- [2] 박송화, “VxWorks 기반 소프트웨어를 위한 테스트 도구의 설계,” 대한임베디드공학회 학회지, Vol. 1, No. 1, pp.45-48, 2011.
- [3] 차순일, “소프트웨어 테스팅 산업 현황과 전망,” 정보과학회논문지, Vol. 28, No. 11, pp.79-85, 2010.
- [4] P. Rumeson, C. Andersson, M. Host, “Test processes in software product evolution - a qualitative survey on the state of practice,” Software Maintenance and Evolution: Research and Practice, Vol. 15, pp.41-59, 2003.
- [5] A Co., Ltd., “Technical Report of SQA Evaluation Model and Guide for Embedded Software Test Process Improvement,” 2004.
- [6] 조규진, “불황 속 소프트웨어 테스팅 시간/비용 절감을 위한 ‘그린코드’ 구현 전략,” 주간기술동향, Vol. 1409, pp.27-37, 2009.
- [7] 박광민, 금대현, 이성훈, “AUTOSAR 임베디드 소프트웨어의 모델기반 개발 및 테스트 방법 -

사례연구 : 운전자 위치제어 시스템,” 대한임베디드공학회논문지, Vol. 4, No. 4, pp.164-173, 2009.

- [8] S.R. Rakitin, “Software Verification and Validation for Practitioners and Managers, ,” Artech House, 2001.
- [9] VxWorks 6.4 Kernel Programmer’s Guide, Wind River.
- [10] K. Yaghmour, “Building Embedded Linux Systems,” O’Reilly, 2004.
- [11] 김기범, 최영한, 양진석, 홍순좌, “결합 주입을 이용한 소프트웨어 보안 테스트,” 정보보호학회 논문지, Vol. 16, No. 5. pp.65-71, 2006.
- [12] J.M. Voas, G. McGraw, “Software Fault Injection Inoculating Programs Against Errors,” Wiley, 1997.

## 저 자 소 개

### 박 송 화



2005년 부산대 전자전기 정보컴퓨터공학부 학사.

2007년 부산대 전자전기 정보컴퓨터공학부 석사.

현재, LIG 넥스원 선임연구원.

관심분야: Real-time OS, 소프트웨어 테스트

Email: shpark25@lignex1.com

### 이 형 수



2000년 경북대 전자전기 공학부 학사.

2002년 경북대 전자공학과 석사.

2009년 경북대 전자공학과 박사.

현재, LIG 넥스원 선임연구원.

관심분야: 임베디드 소프트웨어, 실시간 수중 음향신호처리

Email: hslee1976@lignex1.com