

임베디드 소프트웨어를 위한 원격 단위 테스트 도구의

설계 및 구현

김재현^o 조용윤 곽동규 유재우

송실대학교 컴퓨터학과

{jaehyun^o, yycho, coolman}@ss.ssu.ac.kr, cwyo0@computing.ssu.ac.kr

A Design and Implementation of the Remote Unit Testing Tool for Embedded Software

Jae-Hyun Kim^o Yong-Yoon Cho Dong-Gyu Kwak Chae-Woo Yoo
Dept. of Computing, Soongsil University

요 약

소프트웨어 개발의 생산성을 높이고 신뢰성 있는 프로그램을 개발하기 위하여 테스트 자동화에 대한 연구가 활발히 진행되고 있다. 임베디드 소프트웨어는 일반 소프트웨어 개발 환경과는 달리 호스트-타겟(host-target) 구조의 교차 개발환경에서 개발이 이루어지고 있다. 본 논문은 임베디드 소프트웨어의 개발 환경에 맞추어 제한된 자원의 특성을 고려한 원격 단위 테스트 도구를 제안한다.

제안하는 원격 단위 테스트 도구는 XML 테스트 스크립트를 기반으로 대상 소스 언어에 맞는 테스트 드라이버를 생성하여 테스트 대상과 함께 테스트 작업 용구(Test harness)를 생성한다. 이렇게 생성된 테스트 작업 용구는 이더넷(ethernet)을 통해 타겟(target)으로 전송되어 테스트를 수행한 후, 테스트 결과 로 그를 호스트로 전송한다. 호스트에서는 로그를 수집하여 XML형태의 파일로 제공하여 다양한 형태의 리포트 뷰(View)를 가능하게 한다. 본 논문의 원격 단위 테스트 도구는 임베디드 소프트웨어 개발 단계에서 반복적인 단위 테스트의 절차를 간소화 하여 보다 효율적이고 신뢰성 있는 임베디드 소프트웨어 개발의 기회를 제공할 것으로 기대된다.

1. 서 론

컴퓨터 소프트웨어가 개발되기 시작한 이래로, 시간이 흐르면 흐를수록 소프트웨어는 점점 그 규모가 커지고 복잡해지고 있다. 이러한 복잡해져 가는 소프트웨어의 신뢰성 테스트를 위하여 점진적이고 반복적인 소프트웨어 개발 방법론이 각광을 받으면서 분석, 설계, 구현, 테스트 단계 사이클의 짧은 주기의 반복적인 정책이 강조되고 있다.[2] 소프트웨어 개발 방법론 중의 하나인 XP(eXtreme Programming) 방법론에서는 무엇보다도 테스트의 중요성이 강조되고 있으며 개발해야 할 소프트웨어의 테스트 케이스를 먼저 작성하고 프로그램을 작성하기를 권고한다.[3][4] 최근에는 다양한 단말 장치에 임베디드 소프트웨어가 포함되고 있고, 임베디드 소프트웨어의 요구사항이 점점 커지고 복잡해짐에 따라 신뢰성 있는 임베디드 소프트웨어의 개발은 점점 더 어려워지고 있다. 신뢰성 있는 임베디드 소프트웨어를 개발하기 위하여 보다 자동화된 임베디드 소프트웨어의 단위 테스트 도구를 요구하게 되었다.

본 논문에서는 임베디드 소프트웨어 개발단계에서 소프트웨어의 자동화된 원격 단위 테스트를 통해 소프트웨

어의 신뢰성을 향상시킬 수 있는 단위 테스트 도구를 제안한다. 제안하는 테스트 도구는 임베디드 소프트웨어 개발자들에게 테스트 데이터의 입력에 편리한 GUI를 제공하며, 교차 개발 환경에 의해 컴파일 된 실행파일은 호스트 에이전트를 거쳐 이더넷(ethernet)을 통해 타겟 에이전트로 전달된다. 실행 파일은 테스트 결과를 다시 호스트로 전송하여 개발자에게 테스트 결과를 제공한다. 또한, 문자 스트링 형태의 저수준 테스트 결과를 변환해주는 데이터 처리기를 포함하여 보다 효율적인 테스트 성능을 제공한다.

본 논문은 2장에서 관련연구를 소개하고 3장에서 원격 테스트 도구를 제안하고 4장에서 C언어로 작성된 임베디드 프로그램을 대상으로 하는 원격 테스트 도구를 구현한다. 그리고 5장에서 결론 및 향후 연구 과제를 기술한다.

2. 관련연구

2.1 XML 테스트 스크립트 기반 단위 테스트 도구[1]

테스트 스크립트를 XML로 기술함으로써 기존의 단위 테스트 도구와 비교하여 가독성이 높고 익히기 용이하다는 장점이 있다. 테스트 대상 소스로부터 골격을 생성하

는 테스트 스크립트 생성기와 테스트 스크립트를 대상언어로 작성된 테스트 드라이버로 변환해 주는 테스트 드라이버 변환기를 제공함으로써 보다 간편한 테스트 환경을 제공한다. 본 논문에서는 XML 테스트 스크립트[1]를 이용한 단위 테스트 도구를 호스트-타겟 구조를 이용하여 원격 실행되는 형태로서 임베디드 소프트웨어에 적용한다.

2.2 교차 개발 환경(Cross Tool Chain)

일반 소프트웨어의 개발과 같이 임베디드 소프트웨어 개발도 컴파일러, 링커, 인터프리터, 통합 개발환경 그리고 이와 관련된 개발 툴을 사용한다. 그러나 임베디드 소프트웨어의 개발 툴은 이들이 동작하는 플랫폼과는 다른 플랫폼에서 동작할 응용 프로그램을 개발하는데 사용된다는 점이 다르다. 이런 특성 때문에 일반적으로 임베디드 소프트웨어 개발 환경을 교차 개발 환경이라고 한다.[5]

2.3 테스트 작업 용구(Test harness)

단위테스트에 사용되는 테스트 작업 용구는 그림1과 같이 테스트 대상함수, 스텝(stub), 테스트 드라이버로 구성된다.[6]

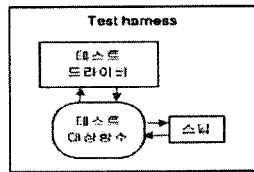


그림1. 테스트 작업용구(Test harness)

테스트 드라이버는 테스트를 수행하는 코드를 수행하여 테스트 대상 함수를 호출한다. 테스트 대상 함수는 개발 중인 함수를 대신하는 스텝을 필요에 따라 호출하며 실행되고 테스트 수행 결과는 테스트 드라이버에 의해 수집된다.

3. 원격 테스트 도구 설계

본 논문에서 제안하는 원격 단위 테스트 도구의 구성은 그림2와 같이 호스트-타겟 구조로 되어있다.

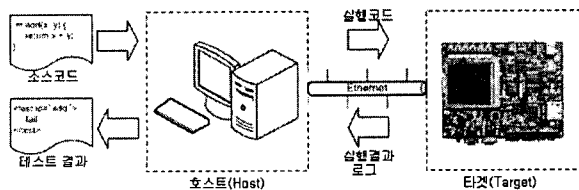


그림2. 원격 테스트 도구 구성도

호스트는 테스트 대상 소스코드를 입력받아 테스트 작업용구를 구성하여 교차 컴파일러를 이용하여 컴파일 한 후, 실행코드를 이더넷(ethernet)을 통해 타겟으로 전송하여 실행시키고, 실행 과정에 생성된 로그 자료를 호스트로 전송받아 테스트 결과를 출력한다. 호스트 에이전트와 타겟 에이전트 사이에는 TCP/IP 프로토콜을 통한 통신 세션을 지속적으로 유지되며, 네트워크의 전송 오버헤드를 줄이기 위해 버퍼링하여 패킷을 전송하게 된다.

3.1 테스트 작업 용구의 생성

단위 테스트를 위한 테스트 입력 데이터를 GUI를 통하여 사용자로부터 입력받아 XML 테스트 스크립트를 생성하고, 테스트 대상 언어로 작성된 테스트 드라이버로 변환된다.[1] 생성된 테스트 드라이버는 테스트 결과를 수집하는 타겟 라이브러리(target library)와 테스트 대상 함수, 스텝과 함께 교차 컴파일러에 의해 컴파일 되어 타겟 시스템에서 실행될 수 있는 ARM용 ELF 형태의 실행파일을 산출한다. 이렇게 생성된 실행 코드는 호스트 에이전트를 통해 타겟으로 전송되어 실행된다.

3.2 호스트 에이전트(Host agent)

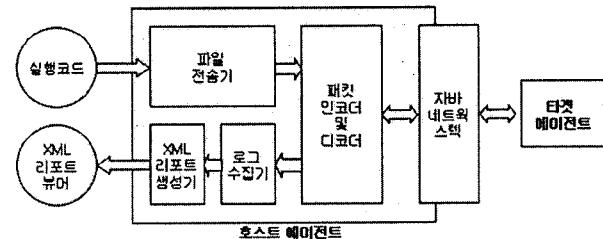


그림3. 호스트 에이전트 구조도

호스트 에이전트는 타겟 에이전트로 ELF형태의 실행 파일인 테스트 작업 용구를 전송하고 실행명령을 내리며, 패킷 형태의 로그를 전달받아 테스트 결과를 XML 리포트로 변환하는 역할을 한다. 또한, 패킷 인코더와 디코더를 포함하여 가능한 작은 크기의 실행코드를 타겟으로 전송하고, 테스트 수행 결과를 디코딩하여 로그 수집기로 전달한다.

3.3 타겟 에이전트(Target agent)

타겟 에이전트는 호스트 에이전트로부터 테스트 실행 코드를 전달 받아 동적 로더(dynamic loader)를 통해 이름 테이블을 재배치(relocate)하고 타겟 운영체제에서 실행시키며, 테스트 드라이버에 포함된 로그 라이브러리로

부터 테스트 결과 정보를 수집하여 호스트 에이전트로 전달한다.

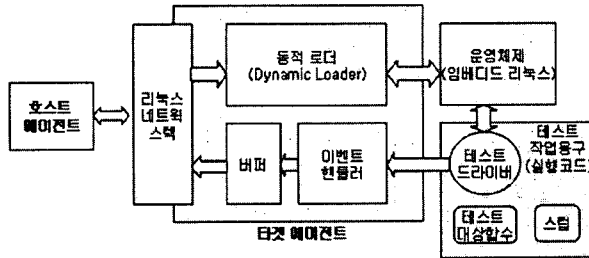


그림4. 타겟 에이전트 구조도

4. 구현 및 실험

제안하는 원격 단위 테스트 도구는 자바 언어를 사용하여 이클립스 플러그인으로 형태로 구현하였다. 자바 언어로 구현함으로써 호스트 개발환경이 시스템에 독립적이라는 장점이 있다. 또한, 이클립스 플러그인으로 구현함으로써 개발 환경에 필요한 많은 기본적인 부분들을 재사용 할 수 있다는 장점이 있다.

실험에서 사용된 타겟 시스템은 ARM 코어 기반의 프로세서를 사용하는 임베디드 개발보드이며, 운영체제는 리눅스환경에서 실험하였다. 테스트 대상 소스의 구문 분석에 사용되는 파서로는 자바 기반의 파서 제너레이터인 SableCC[7]에 C문법을 입력하여 생성된 파서를 사용하였으며, XML 테스트 스크립트 처리 및 XML 기반 레포트 생성을 위해 JAXP[8]를 이용하였다. 또한, XML 테스트 스크립트는 재사용이 가능하도록 프로젝트 별로 관리하여 테스트 데이터의 값만 수정함으로써 반복적인 테스트가 가능하다.

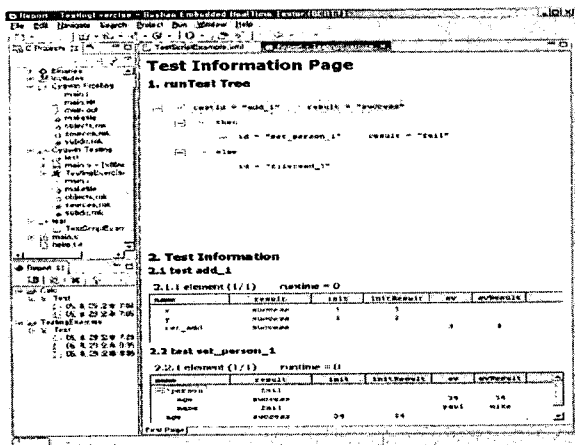


그림5. 테스트 실행 리포트 결과

그림5는 단위 테스트 3개를 설정하고 테스트의 성공, 실패의 흐름에 따라 분기하여 테스트를 진행한 후 호스트 개발환경에 수집된 레포트 결과이다.

5. 결론 및 향후 연구 과제

본 논문은 임베디드 소프트웨어를 위한 GUI기반 단위 테스트 도구를 설계하였다. GUI를 통하여 편리한 사용자 인터페이스를 통하여 테스트 데이터의 입력을 받아드리고, 호스트 에이전트와 타겟 에이전트를 이용하여 테스트 작업 용구를 자동으로 타겟으로 전송하여 원격으로 실행시킴으로서 기존의 방법처럼 프로그램을 컴파일 하여 실행할 때마다 타겟 이미지를 모두 새로 만들어 적재할 필요가 없다. 또한, XML 형태로 테스트 결과를 산출함으로써 사용자에게 다양한 뷰를 제공할 수 있으며, 재사용 방법에 따라 다양한 응용이 가능하다.

임베디드 소프트웨어의 개발 환경에 맞추어 호스트-타겟 구조에서 보다 간편한 단위 테스트 환경을 제공함으로써 임베디드 소프트웨어의 개발 기간을 단축시키고 보다 신뢰성 있는 소프트웨어를 개발에 기여할 것으로 기대된다. 향후, 본 논문에서 제안한 원격 테스트 도구는 보다 다양한 임베디드 타겟 환경에서 실행 될 수 있는 범용 타겟 형태로 구현 되어야 할 것이다.

참고문헌

- [1] 김재현, 조용운, 유재우, XML 테스트 스크립트 기반 단위 테스트 도구의 설계 및 구현, 한국컴퓨터종합학술대회, 2005.07
- [2] J.D. McGregor & D. A. Sykes, "A Practical Guide to Testing Object-Oriented Software", Addison-Wesley, 2001
- [3] Lisa Crispin, Tip House, Testing Extreme Programming, Addison-Wesley, October. 2002
- [4] Marc J. Balcer, William M. Hasling, Thomas J. Ostrand, Automatic Generation of Test Scripts from Formal Test Specifications, ACM, 1989
- [5] Karim Yaghmour, "Building Embedded Linux Systems", O'Reilly, 2004
- [6] Rational Test Realtime, <http://www-306.ibm.com/software/awdtools/test/realtime>
- [7] SableCC, <http://sablecc.org>
- [8] JAXP, <http://java.sun.com/xml/jaxp>