임베디드 소프트웨어를 위한 실시간 성능 테스팅 도구의 설계 및 구현

신경호°, 조용윤, 유재우
숭실대학교 일반대학원 컴퓨터학과
delios@orgio.net, yycho@ss.ssu.ac.kr, cwyoo@comp.ssu.ac.kr

A Design and Implementation of the Performance Testing Tool for Embedded Softwares

Kyoung-Ho Shin, Yong-Yoon Cho, Chea-Woo Yoo
Dept of Computing, Soongsil University

요 약

본 논문은 임베디드 시스템의 제한된 자원을 효율적으로 사용할 수 있는 임베디드 소프트웨어의 개발을 위한 성능 측정 도구를 제안한다. 제안하는 성능 측정 도구는 편리한 GUI를 제공하는 호스트-타겟(host-target) 기반의 테스팅 환경이다. 제안하는 테스팅 도구는 타겟 머신에 맞는 테스팅 코드 생성을 위한 파싱 엔진과 호스트-타겟간 소스 코드와 결과의 전송 및 테스팅 수행을 위한 에이전트 모듈로 구성된다. 또한, 문자 스트링 형태의 저수준 테스팅 결과를 API 형태로 변경하는 데이터 처리기와 API 형태의 결과 정보를 그래픽 형태로 출력하는 레포트 생성기를 포함한다. 본 테스팅 도구는 개발자에게 레포트 뷰(view)를 제공하여 빠른 성능 분석과 코드 수정을 지원하며, 효율적이고 신뢰성 있는 임베디드 기반 소프트웨어 개발의 기회를 제공할 것으로 기대된다.

1. 서론

임베디드 소프트웨어에 대한 요구사항이 복잡해짐에 따라, 제한된 자원에 대해 효율적이고 신뢰성 있는 임베디드 소프트웨어의 개발은 더욱 어려워지게 되었다. 임베디드 소프트웨어 개발자들은 교차 개발 환경에서의 임베디드 소프트웨어 개발 시간 단축과 효율성 향상을 위해 임베디드 소프트웨어 성능 테스팅 도구를 요구하게 되었다.

본 논문은 효율적인 임베디드 소프트웨어 개발을 위해 임베디드 소프트웨어의 성능을 테스팅하고 GUI 기반의 레포트를 통한 결과 분석 기능을 제공하는 임베디드 소프트웨어 성능 테스팅 도구를 설계한다. 제안하는 성능 테스팅 도구는 시스템에 독립적인 자바(Java) 기반 GUI를 제공함으로써, 빠르고 편리한 테스팅 및 분석을 통해 신뢰성과 효율성 있는 임베디드 관련 소프트웨어 개발을 보장할 수 있다. 또한 순수 소프트웨어이므로 추가적인 하드웨어 장비를 요구하지 않는다.

2. 본론

2.1 관련 연구

Soft4Soft의 RESORT 는 Software Metrics기반으로 패키지 S/W를 분석, 테스팅, 품질관리를 지원하는 솔루션 도구이다. 그러나, 이것은 일반 패키지 소프트웨어의 개발지원도구로, 임베디드 소프트웨어개발 환경에서는 사용할 수 없다. AstonLinux의 CodeMaker 는 윈도우 환경에서 리눅스 기반 타겟 시스템을 개발하는 통합개발환경으로 원격 디버깅 기능과 소스 수준 디버깅 기능을 제공한다. 그러나, CodeMaker는 임베디드 소프트웨어 시험검증 도구로 사용할 수 없으며, 타겟시스템 기반의 테스팅이 불가능하고 실시간 분석기능을 제공하지 않는다.

2.2 요구 사항

임베디드 소프트웨어는 제한된 자원의 효율적인 사용을 위해 다음과 같은 사항에 대해 최적화된 개발이 보장되어야 한다.

가. 프로세서 자원을 적게 사용해야 한다. 제한된 임베디드 시스템의 프로세서 성능에 대해 빠른 실행을 위하여 불필요한 코드를 줄이고 적은 양의 코드가 생성되도록 작성해야 한다.

나. 메모리 자원을 적게 사용해야 한다. 임베디드 시스템의 제한된 메모리 자원에 대해 모든 프로그램은 작은 실행 코드를 가져야 하며, 데이터 저장을 위해 사용하는 메모리도 적어야 한다. 또한, 메모리 누수를 발생시키지 않아야 한다.

2.3 제안하는 성능 테스팅 도구

그림 1은 교차 환경 기반의 프로파일 도구의 전체적인 개념도 이다.

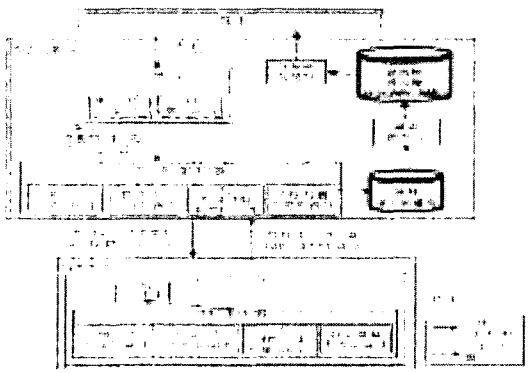

그림 1. 임베디드 소프트웨어 테스팅 도구의 전체적인 개념도

제안하는 테스팅 도구는 원시 테스팅 코드에 대한 타겟 실행 파일 생성을 위한 크로스 컴파일러 및 추가 코드 삽입기와 실행 파일을 타겟 보드에 로드(load)하고 실행 및 결과 전송을 위한 에이젼트 모듈 그리고, 사용자에게 결과를 출력하기 위한 뷰어(viewer)로 구성된다.

2.3.1 코드 분석기

제안하는 시스템은 교차 개발된 임베디드 소프트웨어에 대해 타겟에서 실행될 수 있는 실행 파일을 생성하기 위해 코드 분석기를 포함한다. 다음 그림 2는 코드 분석기에 대한 개념도 이다.

소스 코드 삽입기에 의해 생성된 중간 코드는 어휘, 구문, 의미 분석 과정을 거친 후 소스 코드 삽입기 모듈에 의하여 필요한 위치에 프로파일링 코드가 삽입된 새로운 소스 코드로 변환된다. 이렇게 만들어진 테스트를 위한 최종 소스 코드를 교차 컴파일러 모듈의 입력으로 전달하여 테스트 대상 프로그램이 수행될 환경에 적합한 목적 코드로 컴파일 하게 된다.

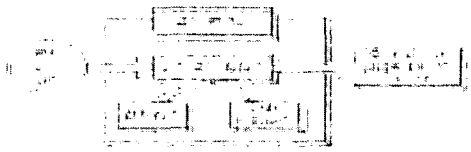

그림 2. 코드 분석기 구성도

2.3.2 에이전트 모듈

제안하는 성능 테스팅 도구는 호스트와 타겟에 클라이언트-서버 형태의 프로파일 에이전트를 가진다. 에이전트 모듈은 호스트에서 실행되는 호스트 프로파일러 모듈과 타겟에서 실행되는 타겟 프로파일러 모듈사이에서 테스팅 코드와 결과의 전송과 연결을 위한 통신 세션을 유지한다. 또한, 각 테스팅 프로파일에 따라 해당 프로파일러들을 실행하고 제어하는 제어 역할을 수행 한다.

2.3.3 테스팅 모듈

제안하는 성능 테스팅 모듈은 다음과 같은 구별된 요구사항에 대해 테스팅을 수행한다.

테스팅 범위	내용
성능 테스팅	- 응용 프로그램 전체 또는 원하는 일부분의 실행에 걸리는 시간을 측정 - 특정 함수별 실행 시간 측정 - CPU 할당 및 처리 시간 측정
코드 범위 테스팅	- 응용 프로그램을 실행 시 실제로 사용되는 부분과 사용되지 않는 부분, 자주 사용되는 부분, 거의 사용되지 않는 부분을 측정
메모리 테스팅	- 메모리의 할당과 해제 정보를 출력 - 전체 메모리 사용량 측정. 또한, - 할당 되었는데 해제되지 않은 메모리를 검사함으로써 메모리 누수를 측정 - 중복 해제되어 버그를 유발시킬 수 있는 코드를 측정
트레이스 테스팅	- 응용 프로그램의 실행과 함께 어떤 순서로 함수 호출이 일어나는지를 출력 - 응용 프로그램이 정상적으로 진행되고 있는지 여부 측정 - 불필요하게 함수 호출이 많이 일어나지는 않는지를 측정

2.3.4 데이터 분석기

타겟에서 실행된 소스 코드에 대한 테스팅 결과를 분석하고 조작하기 쉬운 형태로 결과를 변경하기 위한 모듈이다. 데이터 분석기는 레포트 생성기가 사용자의 요구에 맞는 GUI 형태의 결과를 출력할 수 있도록 API 와 XML 형태로 테스팅 결과를 변경

저장한다. 그림 3은 데이터 분석기 모듈을 구성하는 내부 모듈을 나타낸다.

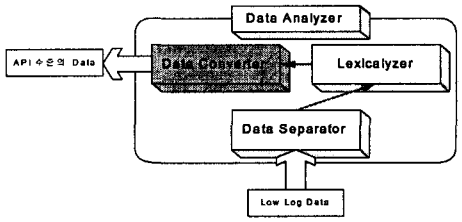

그림 3. 데이터 분석기 모듈 내부 구성도

Data Analyzer는 Data Separator와 Lexicalyzer, Data Converter로 구성되어 있다. Data Separator는 저수준 로그의 종류를 분석하여 분류하는 역할을 하고 Lexicalyzer는 분석된 저수준의 로그를 변환 가능한 형태로 변환하는 모듈이며 Data Converter는 사용 가능한 API형태로 변환하는 모듈이다

2.3.5 레포트 생성기

레포트 생성기는 결과에 대해 사용자에게 GUI 형태의 레포트를 생성하기 위한 모듈이다. GUI를 통해 사용자가 원하는 테스팅 레포트의 종류를 선택하면, 레포트 생성기는 데이터 분석기가 테스팅 결과에 대해 생성한 API와 XML 파일을 이용해 그래픽 형태의 결과를 출력한다.

3.구현및실험

제안하는 성능 테스팅 도구는 시스템에 독립적인 특성을 위해 자바로 구현되었다. 타겟은 ARM9 보드이며, 타겟 운영체제는 Embedded Linux이다. 그림 4는 제안된 임베디드 테스팅 도구의 초기 화면과 테스팅 소스 코드이다. 그림 5는 테스팅 소스 코드에 대해 실시한 메모리 테스팅에 대한 GUI 형태의 출력 화면이다.

4. 결론 및 향후 과제

본 논문은 임베디드 소프트웨어의 성능 테스팅을 위한 GUI 기반 성능 테스팅 도구를 설계하였다. 제안된 도구는 추가적인 하드웨어 장비를 요구하지 않는 순수 소프트웨어 성능 평가 도구이다. 또한, 제안한 도구는 개발자에게 편리한 GUI를 통해 임베디드 소프트웨어에 대한 간편하고 안정적인 테스팅 기회를 제공한다. 임베디드 개발자는 테스팅 결과에 대해 그래픽 형태의 다양한 레포트를 얻을 수 있다. 따라서, 개발자는 개발 프로그램의 성능 최적화를 위한 시간과 노력을 줄일 수 있어 보다 빠르고 안정된 임베디드 프로그램 개발 효율성을 얻을 수 있을 것으로 기대된다. 향후, 제안된 성능 테스팅 도구는 보다 다양한 임베디드 타겟 환경에서 실행 될 수 있는 범용 타겟 형태로 구현되어야 할 것이다.

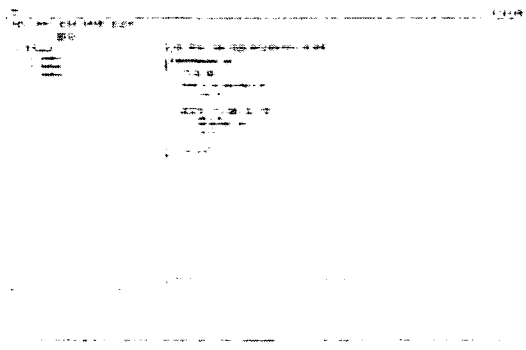

그림 4. 임베디드 소프트웨어 성능 테스팅 도구

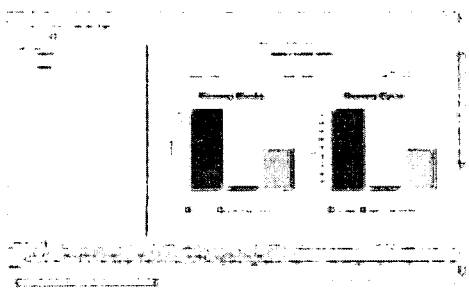

그림 5. 메모리 테스팅 결과

참고문헌

[1] 공기석, 손승우, 임채덕, 김홍남, "내장형 실시간 소프트웨어의 원격디버깅을 위한 디버그에이전트의 설계 및 구현", *한국정보과학회 가을 학술발표논문집 Vol. 26. No 2*, pp.125~127, 1999.

[2] In-Band Diagnostic, Debug and Reporting Tunneling Protocol-FatPipe Protocol, http://www.ineoquest.com/pub/docs/Papers/FatPipe_v2.pdf

[3] Dr. Neal Stollon, Rick Leatherman, Bruce Ableidinger, "Multi-Core Embedded Debug for Structured ASIC Systems", *proceedings of DesignCon 2004*, Feb, 2004.

[4] Bart Broekman, Testing Embedded Software, Addisson-wesley, Dec. 2002.

[5] L.Hatton, Embedded software testing, Software Testing Congress, 2000