

For Better and Higher
Intelligence to the World

제17권 제1호

ISSN 2005-0011

제33회 한국정보처리학회 춘계학술발표대회 논문집

일 자 2010년 4월 23일(금)~24일(토)

장 소 강원대학교(춘천캠퍼스)

주 최 사단법인 한국정보처리학회

주 관 사단법인 한국정보처리학회, 강원대학교 IT대학,
강원대학교 정보통신연구소

후 원 정보통신산업진흥원, 한국과학기술단체총연합회

협 찬 (주)LG CNS, NHN(주), POSCO ICT, SK브로드밴드(주),
소프트하모니, SK C&C, 롯데정보통신(주),
(주)비트컴퓨터, 한국생산성본부, 한솔PNS(주),
현대정보기술(주), (주)씨유메디칼시스템,
(주)한국정보시스템공인감리단

The 33rd KIPS Spring Conference

<http://www.kips.or.kr>

목 차

임베디드시스템

- 001 RFID를 이용한 언어 학습기의 설계 및 구현 KIPS_C2010A_0012
최광원, 김남형, 박진용, 유영준(한림대학교), 안정호(강원도 청소년 S/W 기술교육장), 김진환, 박찬영(한림대학교) 3
- 002 DBRS:B2J를 이용한 BPEL 엔진 기반 로봇 시스템 설계 및 구현 KIPS_C2010A_0020
이종화, 곽동규, 최재영(숭실대학교) 6
- 003 굴절 차란을 위한 전자렌 조향 시스템 전자 제어 장치 진단 프로그램 개발 KIPS_C2010A_0032
이효걸, 정기현, 최경희, 박태원(아주대학교), 문경호(한국철도기술연구원), 김상중(계명문화대학) 10
- 004 반도체 생산 공정 디바이스 관리 및 모니터링에 관한 연구 KIPS_C2010A_0034
송경수, 이리나, 니 게오르기, 김수희(호서대학교) 14
- 005 OSEK/VDX 운영체제를 위한 Event 서비스 설계 및 구현 KIPS_C2010A_0102
권오용, 임성락(호서대학교), 유영창((주)에프에이리눅스), 송기석(충북대학교) 18
- 006 재구성형 프로세서 성능과 레지스터와의 상관 관계 탐구 KIPS_C2010A_0103
김용주, 허인구, 양승준, 이종원, 최영규, 백운홍(서울대학교) 22
- 007 임베디드 시스템에서 정적 파일을 활용한 YAFFS2 파일시스템의 최적화 KIPS_C2010A_0126
박병훈, 서대화(경북대학교) 26
- 008 Periodic Mapping을 통한 프로세서 레지스터 파일의 온도 관리 KIPS_C2010A_0145
허인구, 박상현, 김용주, 윤종희, 이진용, 백운홍(서울대학교) 29
- 009 SPARK Ada 기반 안전필수 내장형 시스템 개발 KIPS_C2010A_0157
오준석, 김진현, 최진영(고려대학교) 33
- 010 x86 assembly-to-C 변환기 KIPS_C2010A_0174
이호진, 이상희, 박보현, 김선욱(고려대학교) 37
- 011 선박 임베디드 장비 모니터링 도구 설계 KIPS_C2010A_0181
엄익정, 박운용, 윤남식(고려대학교) 41
- 012 데이터그램과 큐를 이용한 EIA-485 통신 KIPS_C2010A_0188
김두호, 이 진, 이혁준, 신운호, 김정선(한양대학교) 45
- 013 C++ 템플릿 기반의 Fixed-Point 연산 라이브러리 KIPS_C2010A_0198
황석중, 김선욱(고려대학교), 민병권(뮤텔테크놀러지(주)) 49
- 014 시각장애인을 위한 시각정보 촉각전달 장치 개발 KIPS_C2010A_0199
김경재, 신강철, 조진수(경원대학교) 53
- 015 데이터 흐름을 반영하는 임베디드 시스템의 코드 자동 생성기 설계 KIPS_C2010A_0232
이병용(LG전자), 류호동, 권진욱, 석미희, 이우진(경북대학교) 56
- 016 안드로이드 1.x 버전을 위한 블루투스 API 의 개발 KIPS_C2010A_0241
박병조, 양희재(경성대학교) 60
- 017 멀티미디어 플레이어 위한 RapidPLUS기반 가상 프로토타이핑 설계 및 구현 KIPS_C2010A_0245
임마누엘 아두, 안성순, 이정배, 최성희(선문대학교) 64

018	Readahead 기능을 활용한 NAND Flash 읽기 성능 향상에 대한 연구	KIPS_C2010A_0246	박호준, 임채덕(한국전자통신연구원)	68
019	임베디드 시스템의 코드 최적화를 위한 프로파일러 설계 및 구현	KIPS_C2010A_0285	장우성, 손현승, 김우열, 김영철(홍익대학교)	72
021	다양한 환경에서 최적의 동적 전압 조절을 위한 적응형 예측 필터	KIPS_C2010A_0311	서범준(고려대학교), 방관후, 정의영(연세대학교), 윤성모(고려대학교)	75
021	무인 비행체를 위한 경량 실시간 프로토콜 기반의 ARINC653 지원	KIPS_C2010A_0314	이상현, 진현욱(건국대학교)	78
022	이종 MPSoC 설계에 관한 연구	KIPS_C2010A_0316	이상철, 이성재, 차영호, 김관영(에이디칩스)	82
023	Polaris-1 보드 상에서 EEMBC 벤치마크 동작 구현	KIPS_C2010A_0317	박기성, 이호균, 김선욱(고려대학교)	86

컴퓨터시스템(유비쿼터스, 분산·병렬, 그리드 등)

024	전자메일 서비스를 이용한 이벤트 알림 시스템의 설계 및 구현	KIPS_C2010A_0021	이승민, 김명일, 경희석, 김재성, 이상민(한국과학기술정보연구원)	91
025	OpenFlow 기반 네트워킹 실험을 위한 플로우 제어 인터페이스 구현	KIPS_C2010A_0024	신성호, 김남근, 김종원(광주과학기술원)	95
026	C# 이용 웹 서비스 구현 사례	KIPS_C2010A_0080	임재걸, 김민혜, 김민수(동국대학교)	99
027	Wi-Fi 전자책 단말기를 위한 전자책 콘텐츠 다운로드 서버 설계 및 구현	KIPS_C2010A_0083	곽경민, 김효근(고려대학교)	103
028	증강현실을 위한 적외선 센서 인터페이스	KIPS_C2010A_0086	최한용, 장재혁, 송창근, 고영웅(한림대학교)	107
029	스타 그래프, 버블정렬 그래프와 팬케익 그래프 사이의 임베딩(embedding) 알고리즘	KIPS_C2010A_0091	김동완(순천대학교), 민준식(경동대학교), 이형욱(순천대학교)	111
030	가상화 기술을 적용한 서버 통합 사례 연구	KIPS_C2010A_0101	고범석(고려대학교)	115
031	연성 실시간 Xen 하이퍼바이저 설계	KIPS_C2010A_0114	허경우, 김진, 고영웅(한림대학교)	119
032	WPAN을 이용한 전기 휠체어용 실내 위치 기반 서비스 플랫폼	KIPS_C2010A_0125	남영진, 박영균, 강신재, 정주석(대구대학교), 김범준(계명대학교), 최병준((주)퀵솔루션)	123
033	산업 안전 서비스를 위한 IP-USN 기반 멀티미디어 공동 작업의 동적 오류 관리기	KIPS_C2010A_0131	고용남(백석대학교)	127
034	그리드 환경에서 서버 가상화를 이용한 작업스케줄링 기법	KIPS_C2010A_0144	김오범(한국방송통신대학교), 박지수(고려대학교), 손진곤(한국방송통신대학교)	131
035	EECCH 환경에 적합한 키 관리 기법에 관한 연구	KIPS_C2010A_0165	장성수, 강동민, 박선호, 정태명(성균관대학교)	135
036	무선 센서 네트워크에서의 클러스터 헤드 스케줄링 기법에 관한 연구	KIPS_C2010A_0167	이준호, 강동민, 김승환, 박선호, 정태명(성균관대학교)	139
037	유비쿼터스 서비스 개발을 위한 서비스의 효과적 융합기법 및 도구	KIPS_C2010A_0168	안춘기, 최재현, 박제원, 이남용(숭실대학교)	143
038	융합 서비스 환경에서 제공되는 서비스를 위한 과금 관리 모델 연구	KIPS_C2010A_0183	김수득, 박선호, 정태명(성균관대학교)	146

임베디드 시스템의 코드 최적화를 위한 프로파일러 설계 및 구현*

장우성*, 손현승*, 김우열*, 김영철*

*홍익대학교 컴퓨터정보통신공학과

e-mail:{jang, son, john, bob}@selab.hongik.ac.kr

Design and Implementation of Code Optimization Profiler for Embedded system

Woo-Sung Jang*, Hyun-Seung Son*, Woo-Yeol Kin*, R. Young-Chul Kim*

*Dept. of CIC, Hongik University, Jochiwon, Korea

요 약

임베디드 시스템은 하드웨어 리소스가 매우 작다. CPU속도가 느리고 메모리 크기도 작다. 이런 환경에서의 소프트웨어는 최적화된 크기를 가지고 수행속도가 빠르며 병목 현상이 없어야한다. 이렇게 코드를 최적화하기 위해서는 현재 코드의 문제를 찾아내야 한다. 이것은 정적 분석으로 만으로는 부족하고 프로그램을 수행시켜가면서 정보를 수집하는 프로파일러가 필요하다. 기존의 프로파일러는 윈도우, 리눅스 상에서 수행되는 응용프로그램을 위한 것이기 때문에 저급 임베디드 시스템에서 프로파일러를 수행할 수 없다. 본 논문에서는 이러한 문제를 해결하기 위해서 임베디드용 프로파일러를 설계 및 구현 한다.

1. 서론

임베디드 시스템의 범위가 넓어졌다고 해서 하드웨어의 성능도 똑같이 증가 한 것은 아니다[1]. 이것은 저비용으로 생산되는 제품일수록 더욱 심하다. 높은 성능의 하드웨어는 높은 비용을 요구하기 때문에 대량 생산되는 임베디드 시스템에게는 가격대비 최적화된 하드웨어를 탑재해야 한다. 이 말은 곧 하드웨어리소스가 제한적이라는 것이다. 그렇기 때문에 소프트웨어의 최적화가 매우 중요하다. 똑같은 제품을 생산하는 두 개의 업체가 있을 때 동일한 기능을 수행한다면 사용자는 비용이 저렴한 제품을 선택하기 때문이다. 결국 제품의 성패는 하드웨어보다는 소프트웨어의 최적화가 더 중요하다. 프로그램의 최적화하는 방법은 프로파일링하여 최적화된 코드를 선택하는 과정으로 수행된다[2,3,4].

기존의 대표적인 프로파일러를 살펴보면 Oprofile[5], GProf[6], VTune[7] 등이 있다. 이들은 코드를 수행하여 함수 호출횟수나 수행 시간 등을 프로파일링 해준다. 그러나 기존의 이러한 도구들은 수행되는 프로그램과 같이 프로그램이 설치되어 있어야 한다. 즉, 프로그램을 개발하는 환경과 수행하는 환경이 다른 크로스 개발환경에서는 적용할 수 없다. 본 논문에서는 이러한 문제를 해결하기 위해서 임베디드 시스템에서 동작할 수 있는 프로파일러를 설계 및 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련된 연구로서 기존의 프로파일러인 Oprofile, GProf, VTune에 대해서 기술한다. 3장에서는 제한한 프로파일러에 대해서 언급한다. 4장에서는 적용사례로 MP3플레이어 소프트웨어를 프로파일링 한다. 마지막으로 5장에서는 결론 및 향후 연구를 언급한다.

2. 관련연구

기존의 프로파일러에는 대표적으로 Oprofile, GProf, VTune이 있다. 이들을 각각 살펴보면 다음과 같다. OProfile은 퍼포먼스 카운터 레지스터와 사용자 공간 데몬에 접근할 수 있는 커널 모듈을 사용하여 이 레지스터에서 데이터들을 모음을 사용한다. 데몬이 시작하기 전에, OProfile은 이벤트의 유형과 각 이벤트의 샘플 카운트를 설정하여 수행한다. OProfile의 장점은 커널 모듈을 사용하기 때문에 낮은 오버헤드로 실행된다. 또한 백그라운드에서 실행되는 데몬으로 시스템 퍼포먼스에 영향을 주지 않고 프로파일링을 수행할 수 있다. 하지만 CPU의 레지스터 정보를 사용하기 때문에 OProfile이 지원하는 프로세서만 가능하고 리눅스에서만 사용 가능하다.

GProf은 GNU profiler으로 프로그램 수행시간동안 각 함수의 호출횟수와 함수 호출시 진입에서 종료할때까지의 시간을 기록해두고 이 정보에 대한 통계를 제공하는 방식이다. 코드를 삽입 방식이고 GCC컴파일러에 내장되어 있다. 컴파일시 -pg옵션만 주면 알아서 정보들이 생성

* 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(2009-0076485).

된다. 장점으로는 GCC와 함께 제공 별도의 설치 필요 없고 간단한 옵션으로 프로파일 수행할 수 있다. 하지만 함수의 호출 횟수와, 시간에 대한 정보만 제공하는 단점이 있다. 또한 GCC 컴파일러를 사용하는 환경에서만 수행 가능하다.

VTune은 Intel의 퍼포먼스 분석 도구이다. 제공되는 요소는 플 그래프, 소스 뷰어, 카운터 모니터, 스레드 프로파일러, 퍼포먼스 튜닝 유틸 등과 같은 다양한 기능을 제공하지만 x86기반의 CPU에서만 동작되고 윈도우와 리눅스 플랫폼에서만 동작한다.

3. 제안한 프로파일러

기존의 프로파일 도구는 프로그램 개발과 실행이 모두 같은 환경에서만 수행이 가능하다. 기존의 프로파일 도구는 컴파일된 이미지를 실행하거나 데몬을 통해 백그라운드로 실행한다. 그러나 임베디드 시스템은 개발환경과 실행환경이 다른 크로스 플랫폼이다. 크로스 컴파일 환경에서는 컴파일된 이미지를 호스트에서 실행 불가능하고 OS가 없을 경우 데몬 형태의 백그라운드 실행방법을 수행할 수 없다. 그래서 본 논문에서는 소프트웨어에 프로파일 코드를 삽입하고 이를 수행한 결과를 UART를 통해 결과를 출력할 수 있는 프로파일러를 제안한다.

프로파일러는 크게 두 가지 부분으로 구성되어 있다. 코드에 삽입되어 프로그램의 수행을 알리는 부분과 프로그램의 수행을 계속적 일어나는 이벤트를 처리하는 부분이 있다. (그림 1)은 프로그램 삽입되는 EP_START와 EP_END의 정의 부분이다. EP_START는 StartProfile함수를 호출하고 EP_END는 StopProfile을 각각 호출한다. 전달하는 인자는 컴파일에 제공되는 변수로 현재의 함수를 나타낸다.

```
#define EP_START StartProfile(__FUNCTION__)
#define EP_END StopProfile(__FUNCTION__)
```

(그림 1) 소프트웨어 삽입되는 프로파일 코드

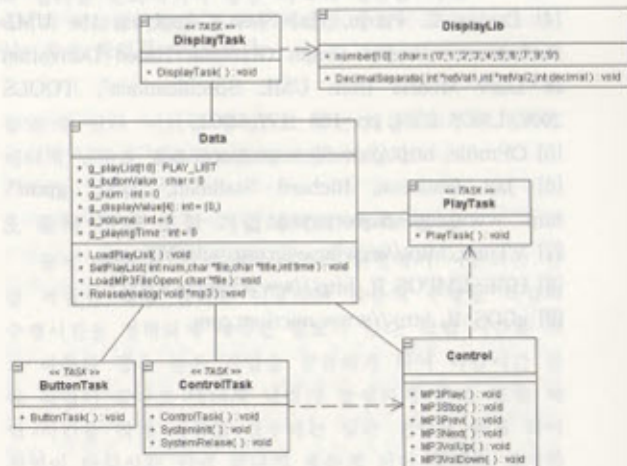
4. 적용사례

적용사례를 위해서 개발한 MP3 플레이어는 (그림 2)와 같이 한백전자 Empos-II[8]를 이용하였다. Empos-II는 PXA255를 사용하고 32MByte의 플래시 메모리와 128MByte의 SDRAM을 가지고 있다. 또한 Text LCD와 8세그먼트, 버튼을 가지고 있어 MP3를 구현하는데 적합한 하드웨어를 갖추고 있다. 또한 uCOS-II[9]를 설치하여 프로그램을 쉽게 구현할 수 있도록 하였다.



(그림 2) MP3를 적용한 Empos-II

구현한 MP3플레이어는 uCOS-II를 이용하여 멀티 스레드로 구성하였다. 화면을 제어하는 DisplayTask와 버튼을 제어하는 ButtonTask, 입력된 버튼을 체크하여 프로그램을 제어하는 ControlTask, 음악을 플레이하는 PlayTask로 각각 스레드로 구성되어 동시에 수행된다. MP3의 자세한 구조는 (그림 3)과 같다.



(그림 3) MP3의 클래스 다이어그램

5. 결론

기존의 프로파일러는 윈도우, 리눅스 상에서 수행되는 응용프로그램을 위한 것이기 때문에 저급 임베디드 시스템에서 프로파일러를 수행할 수 없다. 본 논문에서는 이러한 문제를 해결하기 위해서 임베디드용 프로파일러를 제안하였다. 제안한 프로파일러는 코드 내에 삽입되어 함수의 호출횟수 수행 시간 등을 체크하여 정보를 사용자에게 알려준다. 적용사례로 MP3 프로그램을 구현하여 프로파일러를 수행해보았다. 그 결과 프로그램을 수행하면서 호출되는 함수, 가장 많이 호출되는 함수, 가장 느리게 수행되는 함수를 쉽게 확인이 가능하였다. 임베디드 시스템 프로그램을 최적화할 때 이러한 정보를 이용한다면 쉽게 문

