

정보가전기기에 적용되는 실시간 운용체제용 API 제안

신창민* · 김도형* · 박승민*

*한국전자통신연구원 컴퓨터소프트웨어기술연구소

API Implementation for Internet Appliances

Changmin Shin* · Dohyung Kim** · Seungmin Park***

*ETRI/CSTL

E-mail : cmshin@etri.re..kr

요 약

기존의 정보가전기기들은 기능이 제한적이어서 간단한 제어 프로그램으로 운영이 가능하였으나, 현재의 정보가전기기들은 IEEE 1394, USB, Bluetooth, PLC 등의 새로운 기술들이 출현하고 복잡한 멀티미디어 데이터와 같은 기능들이 확대되어 다양한 기능을 만족시키는 실시간 운영체제의 필요성이 증대하였다. 본 논문에서 이러한 실시간 운영체제들에서 독립적으로 응용프로그램 개발 인터페이스를 지원하는 API를 제안한다.

본 논문에서 제안하는 API는 REDHAT사의 EL/IX 모델을 기반으로 하고, POSIX.1과 ISO C99의 표준 인터페이스를 지원하여 리눅스 및 다양한 실시간 운영체제에 공통적으로 탑재 가능하게 하였다. 또한 계층적인 구조를 가지게 하여 디지털 TV, 셋탑박스, 휴대용 정보단말기 등의 다양한 정보가전기기에 적용될 수 있다. 실험 대상 RTOS로는 한국전자통신연구원에서 개발한 실시간 운영체제인 Qplus-P를 사용하였다.

ABSTRACT

As the new technologies, such as IEEE 1394, USB, Bluetooth, PLC are appeared, the embedded real time operating system which is satisfied the various technologies is needed. This paper proposed and implemented C standard library which is independently used in real time operating system and which supports an application development interface.

The proposed C standard library is based on EL/IX, an Embedded Linux API based on POSIX, which is suitable for other embedded operating systems as well. And because the proposed C standard library has four level, it supports various Internet appliances, such as digital TV, settop box. The developed C library is tested at Qplus-P real time operating system which is developed at ETRI.

키워드

EL/IX, POSIX, Linux, Qplus-P

1. 서 론

사회전반적으로 초고속 정보통신망이 구축이 되어 인터넷을 이용하여 시스템 간의 멀티미디어 데이터 교환이 활발이 이루어지고, 이러한 복합 정보를 이용하는 정보가전 기기의 대중화가 이루어지고 있다. 정보가전 기기로서 디지털 TV, 셋탑박스, 대화형 비디오 시스템, 홈서버, PDA 등을 들 수 있으며, 이러한 시스템들의 기반이 되는 RTOS의 시장은 점점 확대되고 중요성은 커지고 있다[1].

현재 개발되는 RTOS의 기술 형태를 보면, 작은 단말기에서 대형 시스템에까지 적용이 되어 사용될 수 있도록 마이크로 커널과 서로 독립적인 모듈들로 구성이 되는 조립형의 모양의 가진다. 이러한 독립 모듈들로 구성이 된 조립형 실시간 OS들로는 LynxOS, VRTX 등이 있다. VRTX는 Microtec에서 개발이 되어 MentorGraphics에 합병되어 개발도구인 SPECTRA 4.0을 발표하였지만 RTOS의 지속적인 버전업이 이루어지고 있지 않다. 현재 활발한 개발과

시장 점유율을 높이고 있는 Windows CE는 RTOS의 footprint가 크고 실시간 지원에 약한 모습을 보인다. 또한 Windows CE와 함께 부각되고 있는 Embedded Linux에는 RTLinux, eCos 등등 많은 RTOS가 나타나고 있다. 이 외에도 Microware System에서 개발하여 오디오 및 비디오 스트림에 적합한 OS9, QNX사에서 제작한 작고 빠른 RTOS인 QNX 4.25, WindRiver사의 VxWorks 등등이 점점 확대되고 있는 RTOS 시장을 선점하려고 하고 있다 [2-4].

위에서 기술한 바와 같이 정보가전 기기들은 제품의 종류가 아주 다양하게 나타나고 있고, 각 기기들의 기능들도 상이하여 기존의 RTOS로써는 그 기능들을 효율적으로 하기 어렵다. 따라서 새로운 표준의 독립 모듈로 이루어진 RTOS의 개발이 필요하다. 이에 한국전자통신연구원(ETRI)에서는 삼성전자, LG전자, (주)다산인터넷, (주)코스모, (주)휘엔시스와 함께 정보가전기기용 표준형 RTOS인 Q+T[9]와 Q+P를 개발하였다[1]. 본 논문에서는 실시간 운영체제인 Q+P의 일부 모듈로써 개발된 단계적 C 라이브러리를 기술한다.

Q+P는 fast boot, 실시간 스케줄링 등을 지원하는 마이크로 커널과 라이브러리, 개발자용 개발도구인 ESTO, 응용 API로 구성되어 있다. Q+P 라이브러리는 EL/IX[10,11]를 기반으로 한 C 라이브러리, 멀티미디어 파일시스템, 그래픽 가속모듈, GUI 빌더, 네트워크 모듈로 이루어져 있으며, 본 논문에서는 Redhat에서 제안한 EL/IX에 기반으로 하여 만든 단계적 C 라이브러리에 대하여 기술한다.

본 논문에서 제안하는 C 라이브러리는 한국전자통신연구원과 Redhat사와 협력하여 개발되었다. 개발된 C 라이브러리는 Redhat에서 제안하는 EL/IX를 기반으로 하여, 새로이 나타나고 있는 정보가전 기기들에 공통적으로 사용되는 확장성을 가지도록 설계되었다. 또한 POSIX.1의 표준 API 인터페이스와 ISO C99표준을 지원하도록 하여 내장형 리눅스와 여타 다른 실시간 운영체제에서도 개발된 C 라이브러리가 이식될 수 있도록 하였다.

본 논문에서는 2장에서 제안하는 C 라이브러리가 확장성과 이식성을 가지도록 하는 단계모형 기준을 기술하고, 3장에서는 기준에 의하여 API를 분류하며, 4장에서 구현환경과 실험을 다루고, 5장에서는 결론 및 향후 연구를 기술한다.

II. C 라이브러리의 단계모형 기준

기존의 정보단말들은 간단한 제어 프로그램으로 충분히 기기를 운영할 수 있고 기능이 한정적이므로, 각각의 정보단말별로 간단한 RTOS를 개발하여 사용하였다. 그러나 초고속 정보통신망이 확대되고, USB,

IEEE 1394, Bluetooth, PLC 등의 새로운 기술들의 출현과 복잡한 멀티미디어 데이터와 같은 기능들이 확대됨에 따라서 요즘에 나타나는 정보단말들은 다양한 기능을 충족시키는 RTOS가 필요하게 되었다.

이러한 정보단말들의 다양한 기능들을 충족시키는 응용프로그램의 필요성이 발생하였고 응용프로그램 개발의 기본이 되는 C 라이브러리는 다양한 RTOS에 이식이 잘 되어야 한다. 또한 C 라이브러리는 PDA와 같은 정보단말에서 디지털 TV와 같은 복잡한 시스템에 적용이 잘 되는 확장성을 가지게 하여야 한다. 이러한 확장성은 C 라이브러리 API에 레벨을 주어서 구현이 되며, RTOS의 footprint에 효율성을 준다.

따라서 본 논문에서 제안하는 C 라이브러리는 이식성과 확장성이라는 두가지 개발 방향을 가지고 내장형 C 라이브러리를 개발하였다.

위와 같은 정보단말에 적합한 C 라이브러리를 개발하기 위하여 한국전자통신연구원은 Redhat사와 협력하여 C 라이브러리를 개발하였는데, Linux에서의 glibc-2.2.3에서의 API를 기반으로 하였다.

본 논문에서는 제안하는 내장형 C 라이브러리가 이식성을 가지게 하기 위하여 기존의 POSIX.1 표준, ISO C99 표준, Linux/GNU, BSD, SYSV을 따르도록 하였다.

그리고 확장성을 위해서는 RedHat에서 제안하는 EL/IX 표준을 따랐다.

2.1 이식성

IEEE가 주관하여 추진한 프로젝트의 결과로써 나온 POSIX (Portable Operating Systems Interface for Computer Environments)는 UNIX를 기반으로 하는 개방시스템의 규격을 정의한다. 그리고 POSIX는 사실상 AT&T의 시스템 V와 버클리리 BSD판 UNIX의 조합으로 이루어진 시스템 호출 함수를 정의하고 있다. 따라서 POSIX는 응용프로그램들간의 상호 운용성, 응용프로그램의 이식성(Portability), 사용자 이식성 등을 보장하도록 지원한다[5-8].

ISO/IEC POSIX 1003.1 표준은 다음과 같이 구분되어 있다.

- 1003.1a : 기본적인 unix API
- 1003.1b : Realtime Extension(Async I/O, semaphores등)
- 1003.1c : pthread_kill과 pthread_cancel를 제외한 모든 쓰레드 포함
- 1003.1d : Additional Realtime Extensions
- 1003.1j : Advanced Realtime Extensions

POSIX 1003.1x는 POSIX 1003.1(ISO/IEC 9945-1)에서 정의한 규격에 대해 일부분을 개정한 것이며 최종적으로는 POSIX 1003.1에 통합된다. 현재 9945-1:1996 버전에 1003.1b 및 1003.1c까지 통합되어

있다.

POSIX 1003.1b에서 정의하고 있는 실시간 OS의 표준은 LynxOS, OS9, VxWorks, QNX 등 많은 상용 제품에서 지원되고 있다. 따라서 본 논문에서 제안하는 내장형 C 라이브러리는 POSIX.1를 따르고 있으므로 다양한 RTOS에 이식성을 갖는다.

2.2 확장성

EL/IX[10,11]는 내장형 리눅스와 여타 다른 실시간 운영체제에 적합한 단일 표준 API 체계를 목적으로 1999년 Cygnus에 의해 만들어졌다. 그리고 POSIX를 기반으로 API레벨을 4단계로 나누어서 정의되었고 각 레벨에서도 사용자 선택에 의해 기능을 추가 또는 삭제할 수 있는 옵션 기능을 제공하고 있다. 또한 동일한 API 레벨 규격을 만족하는 운영체제 사이에서는 응용의 호환성을 제공한다. EL/IX의 단계별 API 규격은 다음과 같다.

- 레벨 1 : RTOS compatible layer
- 레벨 2 : Linux single process only
- 레벨 3 : Linux multiprocess
- 레벨 4 : Full POSIX or Linux compliance

레벨 1은 대표적인 내장형 실시간 운영체제에서 모두 사용되는 기능을 제공하며, 레벨 2는 단일 처리기 리눅스용 API가 레벨 1에 추가된 것이다. 레벨 3에서는 내장형 응용을 위한 다중 처리기 리눅스용 API 추가되었고, 레벨4는 POSIX API 호환을 뜻한다.

따라서 본 논문에서 제안하는 내장형 C 라이브러리는 EL/IX에 따라서 정의되었기 때문에 PDA와 같은 작은 정보단말에서부터 홈서버와 같은 복잡한 정보기기까지 적용될 수 있는 확장성을 가진다.

EL/IX는 2000년 9월에 기본 API 규격 Draft V1.2가 발표되었으며, 2000년 8월에 EL/IX 규격을 지원하는 Redhat의 내장형 운영체제인 eCOS의 개발 버전이 발표되었다. 현재 인텔, MIPS, 도시바 등이 참여하고 있다.

III. C 라이브러리의 단계모형

본 논문에서 제안하는 내장형태의 단계적 C 라이브러리는 기본적으로 정보단말의 특성상 제안된 환경에 맞게 크기 및 성능이 최적화되어야 한다. 또한 다른 RTOS들과의 이식성, 다양한 정보단말과의 확장성을 위하여 POSIX.1, ISO C99, EL/IX 등의 기존의 표준안들을 기반으로 개발되었다. 또한 다중 처리 및 단일 사용자 환경을 기반으로 하며, 실시간성을 위한 C 라이브러리 함수들을 추가하였다.

제안되는 단계형태의 C 라이브러리는 POSIX 1003.1 표준 API 관련 모듈들과 ISO C99 표준 함수 관련 모듈들로 분류하였는데, POSIX 1003.1 표준

API 관련모듈들은 아래와 같다.

- 프로세스 : `execl()`, `execv()`, `fork()`
- 프로세스 환경 : `getpid()`, `getenv()`
- 파일과 디렉토리 : `opendir()`, `chdir()`
- 입출력 : `pipe()`, `dup()`, `fcntl()`
- 디바이스와 클래스 : `cfgetispeed()`
- C 언어 함수 : `asctime_r()`, `fdopen()`
- 동기화 : `sem_getvalue()`
- 메모리관리 : `mlock()`, `shm_open()`
- 스케줄링 : `sched_get_priority_max()`
- 쓰레드관리 : `pthread_attr_init()`
- 쓰레드관련데이터 : `pthread_key_create()`
- 쓰레드 취소 : `pthread_cleanup_pop()`

ISO C99 표준 함수 관련 모듈들은 아래와 같다. API의 수가 많아서 모듈별로 API의 일부만을 기술한다.

- 입출력함수 : `fgetc()`, `fprintf()`, `setbuf()`
- 터미널 제어 함수 : `cfmakeraw()`
- 오류처리 함수 : `Errno()`, `perror()`
- 파일시스템관련함수 : `fchdir()`, `fseek()`
- 문자열 처리 함수 : `strchr()`, `strlen()`
- 메모리관련함수 : `calloc()`, `free()`
- 프로세스관련함수 : `abort()`, `atexit()`
- 지역화 함수 : `localeconv()`
- 전역 점프 함수 : `longjmp()`
- 시스널 처리 함수 : `signal()`
- 시간처리 함수 : `ctime()`, `localtime()`
- 수학 함수 : `abs()`, `ceil()`, `cos()`
- 사용자,그룹관련함수 : `endgrent()`
- 네트워크관련함수 : `connect()`, `htonl()`
- 기타 함수 : `atof()`, `btowc()`, `qsort()`

본 논문의 내장형 C 라이브러리를 적용하는 RTOS는 Q+P이며, Q+P는 홈서버를 대상 타겟 시스템으로 한다. 홈서버는 PDA와 달리 복잡한 기능을 가지므로 내장형 C 라이브러리의 단계모형에서 EL/IX 레벨 3를 적용한다. 따라서 2장에서 기술한 바와 같이 확장성과, 이식성을 기준으로 glibc-2.2.3의 API들을 분류하고, 분류된 API들 중에서 EL/IX 레벨 3에 해당하는 API들을 타겟 시스템인 홈서버에 적용하였다.

IV. 구현 환경 및 실험

제안하는 확장성과 이식성을 가지는 단계적 C 라이브러리는 Q+P SDK를 이용하여 실험이 이루어졌고 타겟시스템은 Gigabyte GA-60 XE를 메인보드로 하는 홈서버이다. Q+P SDK은 cross tool chain, target configurator, Library reduction, 그리고 다양한 부팅 및 설치 메커니즘을 지원한다.

구현 환경은 다음과 같다.

- Gigabyte GA-60 XE를 메인보드로 하는 타겟인
 홈서버
- Redhat 7.1인 설치된 호스트 시스템
- 타겟 시스템과 호스트 시스템간의 이더넷 연결

실험한 과정으로는, glibc-2.2.3는 모두1945개의 API로 구성되어 있었고, 이중에서 EL/IX 레벨 3인 API는 모두 1409개의 API이었다. 그리고 다시 응용 프로그램 개발자를 위하여 실제 내부에서만 호출되어지는 함수는 제외하여 1016개의 API가 생성되었다.

향후에 타겟 시스템을 홈서버가 아닌 기능이 적은 정보단말로 하였을 경우, 내장형 C 라이브러리는 EL/IX 레벨 3가 아닌 레벨 1 또는 2를 적용하므로 API 개수는1016개의 훨씬 적어질 것이다.

V. 결 론

많은 정보가전 기기들의 출현에 따라서 다양한 기능들을 충족시키는 RTOS의 중요성이 대두되었으며, 그러한 RTOS에 필요한 C 라이브러리는 확장성과 이식성을 가져야 한다. 본 논문에서는 한국전자통신연구원에서 개발한 내장형이고 조립형이 Q+P에 탑재될 C 라이브러리가 확장성과 이식성을 가지도록 C 라이브러리의 단계모형을 제안, 구현하였다.

개발된 단계적 C 라이브러리는 현재 홈서버에서 실험되었다. 내년에는 홈서버와 같은 복잡한 시스템이 아닌 PDA 등의 정보기기에도 적용이 되도록 단계모형의 수정이 이루어져야 한다.

참고문헌

- [1] 인터넷정보가전산업협의회, 인터넷정보가정기술개발 워크샵, 2000.
- [2] VxWorks 5.3.1 Programmers Guide Edition 1, Wind River Systems, 1997.
- [3] pSOSSystem Programmers Reference, Integrated Systems, 1997.
- [4] VTRX Reference Guide, Mentor Graphics Corporation, 1997.
- [5] ISO/IEC 9945-1. C 언어를 위한 시스템 응용 프로그램밍 인터페이스(API) 표준, 1993.
- [6] The GNU C Library Reference Manual, 4 October, 1996.
- [7] Martin Leslie, C Programming Reference Release 1.09, <http://stat4.ph.ssu.ac.kr/cref/C/cref.html>.
- [8] Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, Prentice Hall, 1988.

- [9] Do-Hyung Kim, Sung-Min Park, 실시간 운용체제 Q+를 위한 C 표준 라이브러리 설계 및 구현, 정보처리학회논문지, Vol.8-A, No.1, March, 2001.
- [10] Nick Garnett, EL/IX Base API Specification Draft - V1.2, RedHat Inc. September, 2000.
- [11] Michael Tiemann, EL/IX:Unifying APIs for Linux and Post-PC Computing, EL/IX Whitepaper, September 1999.