

내장형 실시간 운영체제의 입출력 시스템 개발

김선자*, 정귀금*, 이형석*, 김홍남 *
*한국전자통신연구원 인터넷정보가전연구부
e-mail : sunjakim@etri.re.kr

Development of an I/O System for Embedded Real-time OS

Sunja Kim*, Jung, Gwi-Geum*, Lee, Hyung-Seok*, Heung-Nam Kim*
*Dept. of Internet Information Appliance, ETRI
**Dept. of Computer Engineering, Chung-Nam University

요 약

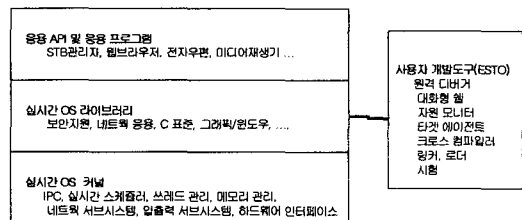
내장형 기기의 다양성 및 적용 범위의 확장으로 내장형 기기용 운영체제도 기기별 다양성을 지원할 것이 요구되고 있다. 이러한 특성을 갖는 운영체제를 조립형 운영체제라 한다. 본 논문에서는 조립형 실시간 운영체제인 Qplus 커널의 입출력 시스템 개발에 대하여 기술한다. Qplus의 입출력 시스템은 사용자의 요구에 따라 입출력 장치 구동기의 조립이 가능하며 일관된 사용자 인터페이스 및 간편한 장치 구동기 인터페이스, 빠른 입출력을 지원한다. 현재 SA110 보드를 사용한 ITSB에서 시험되었으며 앞으로 AIO, 가상 터미널 구동기의 지원 및 입출력 성능 시험을 수행할 예정이다.

1. 서론

초기의 내장형 시스템들은 단순히 제어 루프만으로 구성된 소프트웨어에 의해 제어되었으나, 내장형 시스템 자체가 기능이 확대되고 복잡해지며 네트워크로 연결되는 등 내장형 시스템에도 운영체제가 필요하게 되었다[1]. 이에 1980 년대에 상용 내장형 운영체제가 대거 출현하였으며 이중 VxWorks, pSOS, Neculeus 및 Windows CE 와 같은 운영체제는 괄목할 만한 성장을 보였다. 앞으로도 내장형 시스템은 정보 가전용 기기, 인터넷 수요의 증가에 따른 네트워크 장비, 디지털 TV 용 기기 등 그 수요가 증가할 것으로 보이며 이에 따라 내장형 운영체제도 수요가 폭발할 것으로 예측되고 있다.

한국전자통신연구원에서는 1998년부터 조립형 RTOS 개발 사업을 수행하면서 다양한 종류의 내장형 시스템에 사용될 수 있는 실시간 운영체제를 개발하고 있다[2]. 이 운영체제는 실지 탑재될 기기의 특징에 따라 운영체제에서 제공하는 기능을 선택적으로

채택하여 목표 시스템에 최적인 기능 및 성능으로 동작할 수 있도록 하는 조립형 구조를 주요 특징으로 하고 있다. 이 운영체제를 Qplus 라고 하며, 커널과 ESTO 라는 사용자 개발 도구로 구성된다. <그림 1>은 이 시스템의 구성을 나타낸 것이다.

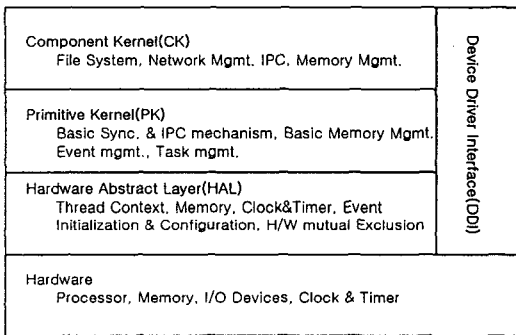


< 그림 1 > Qplus/ESTO 의 구성

본 논문에서는 Qplus 커널의 입출력 서브시스템 개발에 대하여 기술한다. 먼저 2 장에서는 Qplus 커널의 구조에 대해 간략히 기술하고, 3 장에서 Qplus 커널 입출력 서브 시스템 개발 내용을 기술한다. 4 장에서는 시험 환경 및 시험 내용에 대해 설명하고 5 장에서 결론으로 맺는다.

2. Qplus 커널

Qplus 커널은 Neucleus 사의 nano kernel 을 기반으로 하여 설계되었다. 즉 타스크 관리, 스케줄러 및 잠금 기능 등의 기본 기능은 nano kernel 에서 제공하는 기능을 사용하고 그 외 메모리 관리, 네트워크, 입출력, 하드웨어 추상 계층 등은 독자적으로 개발되었다. <그림 2>는 Qplus 커널의 구조를 나타낸다. 현재 SA110(StrongArm) 보드를 사용한 ITSB 및 DTSB 에 탑재되어 동작한다.

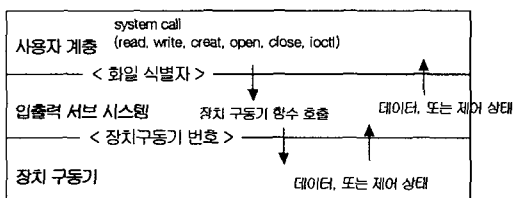


< 그림 2 > Qplus 커널 구조

3. 입출력 시스템 설계

3.1. 개요

Qplus 커널 입출력 시스템은 사용자와 입출력 장치 구동기 사이에서 사용자에게는 손쉽고 일관된 입출력 인터페이스를 제공하고, 또한 손쉬운 장치 구동기 제어 인터페이스를 구축한다. <그림 3>에서와 같이 입출력 서브시스템은 사용자와는 화일 식별자를, 장치 구동기와는 장치구동기 번호를 통해서 인터페이스한다. 사용자 인터페이스는 기본적으로 POSIX 인터페이스를 준수한다.



<그림 3> 입출력 서브 시스템의 위치

시스템 내의 입출력 장치와 장치 구동기는 별도로 관리되며 따라서 논리적으로 임의의 장치는 어떤 장치 구동기로도 구동(drive)할 수 있다. 예를 들어 하나의 시리얼 장치 구동기로 여러개의 문자 포트를 제어할 수 있는 구조를 가진다.

3.2. 화일

Qplus 입출력 서브시스템에서는 화일의 종류에 관계 없이 모든 화일에 동일한 방식의 입출력을 지원한다. 모든 화일은 시스템에서 유일한 식별자를 가지며 이 화일 식별자는 사용자가 화일 생성(creat) 또는 열기(open)시 사용자에게 제공된다. 화일 식별자는 입출력 시스템 내부에서는 화일 테이블로의 인덱스로 사용되며, 이를 통해서 해당 화일에 대한 정보를 얻을 수 있다.

3.3. 장치 구동기 관리

시스템에서 동작하는 장치 구동기는 입출력 시스템 내 장치 스위치 테이블에 등록된다. 입출력 시스템은 특정 장치 구동기를 입출력 시스템에 등록할 수 있는 방법을 제공한다. 장치 구동기는 이 루틴을 호출하여 자신을 입출력 시스템에 등록 또는 제거할 수 있다. 즉 모든 장치 구동기는 자신의 초기화 루틴에서 입출력 시스템의 장치 스위치 테이블에 자신을 등록시키는 과정을 수행하며, 입출력 시스템은 장치 구동기의 등록 후 장치 스위치 테이블에서의 인덱스값을 장치 구동기 번호로써 반환한다. 이때 장치 구동기의 다른 루틴들, 열기(open), 생성(creat), 닫기(close), 삭제(delete), 읽기(read), 쓰기(write), 제어(ioctl) 의 7 가지 기본 루틴의 이름을 입출력 시스템에 전달한다. 입출력 시스템에서는 장치 스위치 테이블에 이 주소를 기록하였다가 해당 장치 구동기 번호를 가지는 입출력 요구시 적절한 장치 구동기 루틴을 호출한다. 위 7 가지 기본 루틴 이외의 기능은 제어(ioctl) 루틴의 함수로 포함시켜서 구현되었다. 예를 들면 select 함수도 ioctl 시스템 호출을 통하여 구동된다. UNIX 시스템에서와 달리 장치 스위치 테이블은 블록 장치와 문자 장치별로 별도로 존재하지 않고 하나의 장치 스위치 테이블을 사용하도록 하였다.

3.4. 장치 관리

시스템에 장착된 입출력 장치들은 이 장치에 대한 정보를 나타내는 자료 구조로 표시된다. 장치의 특성에 따라 자료 구조도 다른 구조를 가지나 모든 입출력 장치 자료 구조들은 동일한 헤더를 포함한다. 그리고 이 헤더를 통해 서로 연결되어 있다. 입출력 시스템에서 장치 정보는 이 연결 리스트에 의해 관리된다. 장치 자료 구조는 크게 문자 장치에 대한 것과 블록 장치에 대한 것으로 나눌 수 있다.

입출력 시스템은 특정 장치를 입출력 시스템에 등록시키기 위한 기능을 제공한다. 사용자는 특정 장치

에 사용할 장치 구동기를 지정하여 장치를 입출력 시스템에 등록한다. 이후에 이 장치에서 화일 생성, 열기, 읽기, 쓰기 등의 입출력이 가능하게 된다.

3.5. 입출력 시스템 초기화

입출력 시스템 초기화 루틴은 사용자로부터 동시에 열기 가능한 화일 수 및 장치 구동기 수를 입력으로 받아 입출력 시스템을 초기화한다. 이에는 화일 식별자 테이블 구축, 장치 스위치 테이블 구축 등이 포함된다. 이 루틴의 실행 이후에 장치 구동기는 장치 구동기를 입출력 시스템에 등록할 수 있다.

3.6. 화일 시스템 인터페이스

블록 장치 구동기는 문자 장치 구동기와는 달리 화일 시스템을 거쳐 관리된다. 즉 블록 장치를 화일 시스템으로 구성하고 그 시스템 하에서 화일을 관리한다. 따라서 블록 장치 구동기를 설치할 때에는 직접 장치 구동기가 입출력 시스템에 등록되지 않고 장치 구동기에서 지정한 화일 시스템이 입출력 시스템에 등록된다. 장치 구동기는 다시 화일 시스템을 통해 호출되어야 한다. 블록 장치 구동기를 사용하기 전에 먼저 블록 장치의 특정 파티션에 화일 시스템을 구축하여야 한다. 입출력 시스템에서는 화일 시스템 구축 및 파티션 설정 등의 기능을 제공한다.

Qplus 커널의 화일 시스템에서는 기존의 UNIX 시스템이 갖는 가상 화일 시스템(Virtual File System, 이하 VFS) 계층을 지원하지 않는다. VFS는 하나의 운영체제가 여러 종류의 화일 시스템을 지원할 때 개발자의 입장에서 새로운 화일 시스템의 추가 또는 삭제보다 용이하게 하기 위한 것이다. 그러나 내장형 시스템의 운영체제에서는 소규모, 최적의 성능이 요구되므로 한 종류의 화일 시스템만 지원하더라도 포함되는 VFS 기능은 부담이 될 수 있다. 또한 Qplus 커널의 입출력 시스템에서는 시스템 화일 식별자 테이블을 통하여 장치 구동기 계층에서의 다양화를 제공할 수 있으므로 VFS 계층을 지원하지 않는다.

3.7. 인터럽트 처리

장치 구동기의 인터럽트는 장치 구동기 설치시 커널 인터럽트 테이블에 등록되어 장치 인터럽트가 처리되도록 경로가 설정되어야 한다. Qplus 커널에서 제공하는 인터럽트 경로 설정을 위한 루틴을 사용하여 장치 구동기 초기화 루틴에서 자신의 인터럽트 처리 경로를 구축한다.

3.8. select 인터페이스

select 시스템 콜은 하나의 task가 여러개의 장치로부터 입출력을 기다리다가 준비된 장치로부터 입출력을 수행할 수 있는 기능을 제공한다. 이 시스템 콜은 장치 구동기가 첫째, 태스크가 장치의 입출력을 기다리는 시간을 지정할 필요가 있을 때, 둘째, 장치 구동기가 여러개의 장치를 지원하고 태스크는 이들 장치

입출력을 동시에 기다릴 필요가 있을 때, 그리고 마지막으로 태스크가 동시에 여러 장치로부터 입출력을 기다릴 필요가 있을 때 유용하게 사용할 수 있다[3].

이 select 시스템 콜을 사용하기 위해서는 먼저 장치 구동기가 select 기능을 제공해야 하며, 입출력 시스템에서는 여러개의 장치 구동기 위에서 일관된 인터페이스로 태스크가 select를 수행할 수 있게 해주어야 한다.

Qplus 입출력 시스템에서 select는 다음과 같이 구현하였다.

- 장치 구동기는 자신에게 select를 요청한 task들의 목록을 관리한다.
- 장치 구동기는 읽기 또는 쓰기 수행시 select를 요청한 task가 있으면 해당 task에 읽기 또는 쓰기의 조건이 충족되었음을 알린다.
- select를 수행하는 task는 모든 요구 장치에 select요청을 알리기 전에 특정 장치의 select조건을 기다리면서 sleep하지 않도록 보장한다. 즉 select를 수행하는 task는 모든 장치 구동기에 select를 기다리고 있음을 알린 후 장치 구동기로부터 select조건이 충족되기를 기다려야 한다.
- select를 기다리는 task가 다른 이유에서 삭제(kill)되는 경우에는 장치 구동기내의 select리스트에서도 제거된다.

task의 select요구는 자료 구조 select_request로 표현된다. select_request에는 select가 요청된 화일 식별자들, 그리고 장치 구동기가 처리한 화일 식별자들의 정보가 포함되어 장치 구동기에 전달된다. 장치 구동기는 select_request들을 자신의 select_list에 연결하고, select의 요구사항이 만족되면 select_list에서 해당 task에게 알려준다. select_list는 mutex lock에 의해 보호된다.

3.9. 네트워크 인터페이스

Qplus 커널의 네트워크는 BSD socket을 기반으로 동작한다. 따라서 socket 시스템 콜을 수행할 때 socket fd가 할당되고 이를 기반으로 통신이 이루어진다. Qplus 입출력 서버 시스템에서는 socket도 하나의 장치 구동기로 처리된다. 즉 네트워크 초기화 루틴에서 먼저 입출력 서버시스템에 socket을 하나의 구동기로 등록하여 socket 장치 구동기 번호를 할당받는다. 이후의 socket 관련 시스템 콜에서는 등록된 socket 장치 구동기로 제어가 이동된다. 시스템 화일 식별자 테이블에는 socket에 대한 화일 식별자가 할당되며 socket의 내부 구조 안에 socket 식별자가 저장된다. 따라서 시스템 콜 socket은 시스템 화일 식별자를 반환하며 socket 식별자는 입출력 서버시스템과 socket 구동기 사이에서만 사용된다. 이 구조는 기존의 socket 구동기의 내용을 변경하지 않고 그대로 사용할 수 있는 장점이 있다.

4. 시험

시험 환경은 SA110 보드를 사용, 자체 제작한 인텔 리전트 TV 셋톱박스(Intelligent TV Settop Box, 이하 ITSB)에서 수행하였다. ITSB는 32MB의 RAM, 16MB의 플래쉬 메모리가 장착되며 두 개의 시리얼 포트가 제공된다. 입출력 서버 시스템을 시험하기 위해 시리얼 장치 구동기와 플래쉬 메모리 장치 구동기를 수정하여 이 장치 구동기가 제대로 동작함을 증명함으로써 입출력 시스템의 시험을 수행하였다.

5. 결론

본 논문에서는 조립형 실시간 운영체제인 Qplus 커널의 입출력 시스템을 설계, 구현, 시험한 내용을 기술하였다. Qplus 커널의 입출력 시스템은 단일 사용자 인터페이스, 효율적인 장치 구동기 인터페이스를 제공하며 시스템 구성에 따라 장치 및 구동기를 추가하거나 삭제할 수 있는 조립형 구조를 지원한다. 모든 종류의 화일에 동일한 입출력 인터페이스를 지원하며 빠른 입출력 처리를 위해 장치 구동기의 역할을 최대화하였다.

본 논문에서 구현한 입출력 서브시스템은 입출력 서브시스템의 구축과 각 장치 구동기의 수정이 필요하였다. 각 장치 구동기에는 입출력 서브시스템 인터페이스 부분 및 select 콜 기능이 추가되었다.

앞으로 AIO(Asynchronous I/O)의 지원, 가상 터미널 및 PIPE 구동기를 지원하도록 확장할 계획이다. 또한 새로운 입출력 시스템 구조 하에서 입출력 성능을 측정하여 다른 내장형 운영체제와 비교, 분석하는 작업이 수행될 것이다.

참고문헌

- [1] Joel R Williams, "Embedding Linux in a Commercial Product", *Linux Journal*, April 1994
- [2] 한국전자통신연구원, "실시간 OS 커널 설계서", 한국전자통신연구원, May 1995
- [3] Alessandro Rubini, "LINUX Device Drivers", O'Reilly, February 1998