실시간 운영체제를 위한 경량 네트워크 스택의 설계 및 구현

이정원°, 전상호, 이숭열, 이철훈
충남대학교 컴퓨터 공학과
{booster°, shjun, sylee, chlee}@ce.cnu.ac.kr

A Design and Implementation of Light Weight Network Stack for Real-time Operating System

Jungwon Lee° Sangho Jeon, Soongyul Lee, Cheolhoon Lee
Dept. of Computer Engineering, Chungnam National Univ.

요 약

정보가전기기의 발전에 따라 인터넷과 퍼스널 컴퓨터 중심의 정보화는 이제 어느 장소, 어느 기기에서나 네트워크와 연결되어 사용자에게 서비스를 제공하는 유비쿼터스 시대로 접어들고 있다. 이러한 정보기기들은 기기에 특성화된 실시간 운영체제를 탑재하고 통신하게 되며, 한정된 자원을 가진 내장형 정보기기들을 위해 경량의 네트워크 스택을 적용해야 할 필요성이 있다. 본 논문에서는 실시간 운영체제상에 내장형 시스템용 경량 네트워크 스택을 설계 및 구현하였다.

1. 서론

데이터 처리 속도와 전송속도의 눈부신 향상으로 많은 정보를 네트워크를 통해 제공받게 되었다. 한정된 자원을 가지고 있는 내장형 시스템의 경우 동일한 기능으로 보다 적은 메모리를 사용하는 경량 프로토콜의 필요성이 부각었다. 네트워크로 모든 기기들을 연결하는 유비쿼터스 환경에서 보다 적은 메모리의 사용은 매우 중요한 이슈가 되고 있다.

범용 운영체제에서 사용하는 TCP/IP 프로토콜 스택의 경우 운영체제에 비해 커다란 네트워크 스택의 크기로 인해 많은 메모리를 사용하게 된다. 이번에 적용하게 된 경량 TCP/IP 프로토콜 스택의 경우 기존의 스택에 비해 적은 메모리 사용량으로도 동일한 기능을 제공하는는 것으로 이미 상당수의 실시간 운영체제에서 이용되고 있다.

본 논문에서는 실시간 운영체제 UbiFOS™에 탑재된 기존의 TCP/IP 프로토콜 스택 대신 경량의 네트워크 스택을 설계 및 구현하였다[1][2][3]

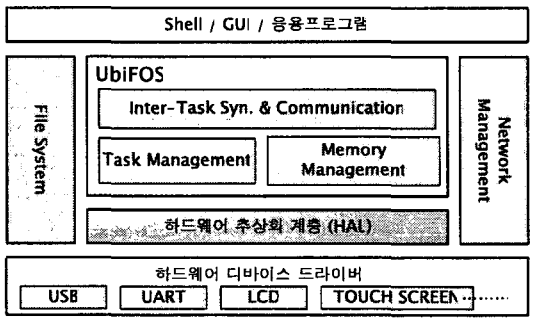
본 논문의 구성은 2 장에서 실시간 운영체제 UbiFOS™와 TCP/IP 를 소개하고, 3 장에서 경량 네트워크 스택과 Socket API 설계 및 구현을, 4 장에서 테스트 환경 및 결과에 대하여 기술한다. 마지막으로, 5 장에서는 결론 및 향후 연구 계획에 대해 기술한다.

2. 관련 연구

2.1 실시간 운영체제 UbiFOS™

UbiFOS™는 선점형 우선순위 기반의 실시간 운영체제 (Real-time Operating System)이다. 태스크는 중요도에 따라 우선순위가 부여되고 가장 높은 우선순위의 태스크가 CPU 를 선점하여 수행한다. 실시간 운영체제 UbiFOS™의 기본 기능은 다음과 같다.

- 우선순위 기반 멀티 태스킹
- 동적 메모리 관리
- 태스크간 통신



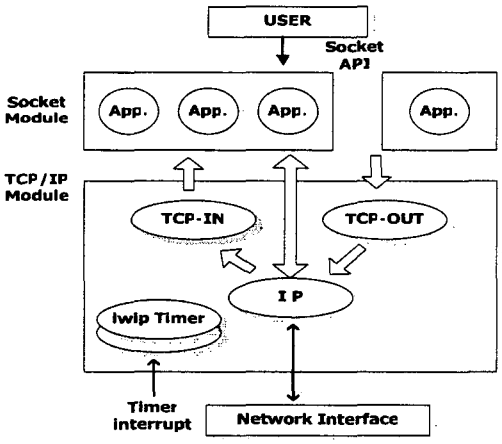
[그림 1] UbiFOS™ 의 구조

* 본 논문은 국방과학연구소의 실시간 운영체제 인터페이스용 미들웨어 연구과제의 수행결과임

2.2 LWIP (A Light Weight TCP/IP stack)

LWIP 는 스위스의 SICS(Swedish Institude of Computer Science)의 CNA(Computer and Network Architectures)연구실에서 개발한 독립적으로 구현된 경량 TCP/IP 프로토콜 스택이다. LWIP 는 TCP 의 모든 기능을 제공하면서 40Kbyte 정도의 코드 크기와 수십 Kbyte 정도의 데이터 크기만 필요로 하는 내장형 시스템을 위한 네트워크 스택이다. LWIP 가 제공하는 기능들은 다음과 같다.

- **IP** (Internet Protocol)
- **ICMP** (Internet Control Message Protocol)
- **UDP** (User Datagram Protocol)
- **TCP** (Transmission Control Protocol)
- **DHCP** (Dynamic Host Configuration Protocol)
- **ARP** (Address Resolution Protocol)
- Berkeley-alike **Socket API**
- **raw API** for enhanced performance



[그림 2] lwip in UbiFOS™

3. 경량 TCP/IP Protocol Stack 의 설계 및 구현

3.1 운영체제 시스템 함수

lwip 를 UbiFOS™에 적용하기 위해 운영체제 의존적인 부분들에 대한 설계 및 구현이 필요하다.

```
void sys_init(void *pAddr);
sys_sem_t sys_sem_new(u8_t count);
void sys_sem_free(sys_sem_t sem)
void sys_sem_signal(sys_sem_t sem)
u32_t sys_arch_sem_wait(sys_sem_t sem, u32_t
    timeout)
sys_mbox_t sys_mbox_new(void)
void sys_mbox_free(sys_mbox_t mbox)
void sys_mbox_post(sys_mbox_t mbox, void
    *msg)
u32_t sys_arch_mbox_fetch(sys_mbox_t mbox,
    void **msg, u32_t timeout)
struct sys_timeouts * sys_arch_timeouts(void)
sys_thread_t sys_thread_new(void (* thread)(void
    *arg), void *arg, int prio)
```

[그림 3] 운영체제 의존적 함수

sys_init()는 메모리와 타이머 관련 부분을 초기화 하여 lwip 네트워크 스택을 초기화 해주는 함수이며, **sys_sem_new() / sys_sem_free() / sys_sem_signal() / sys_arch_sem_wait()** 함수는 세마포어를 생성 / 삭제 / 세마포어 자원의 반납 / 세마포어 자원의 요청을 수행하는 함수이다. **sys_mbox_new() / sys_mbox_free() / sys_mbox_post() / sys_arch_mbox_fetch()** 함수는 메시지 메일박스를 생성 / 삭제 / 메시지 송신 / 메시지 수신을 수행하는 함수이다. **sys_arch_timeouts()** 함수는 타이머를 설정해주는 함수이며, **sys_thread_new()** 함수는 새로운 쓰레드를 생성하는 함수이다.

3.2 네트워크 드라이버

운영체제 의존적인 부분이외에도 하드웨어 의존적인 부분에 대한 설계도 필요하다. 본 연구를 위해 네트워크 인터페이스에 대한 초기화 및 송 / 수신 루틴을 설계 및 구현하였다.

```
MK_Status_t S3C2440_HWinit(MK_Void_t);
MK_Void_t MK_NetworkInitialize( MK_Void_t
    *pAddr);
MK_Status_t cs8900if_init(struct netif *netif)
```

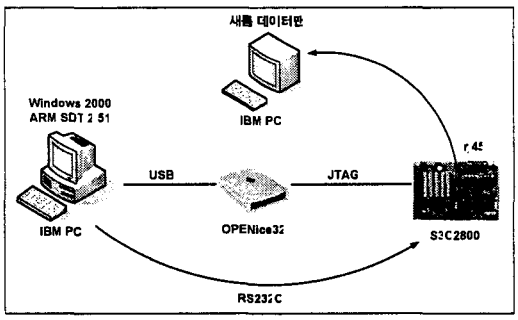
[그림 4] 네트워크 관련 API

MK_NetworkInitialize()함수는 네트워크 디바이스를 초기화 하고 lwip 네트워크 스택의 각 모듈들을 초기화 하는 함수이며 **S3C2440_HWinit()**함수는 개발 보드를 초기화 하는 함수 이다. **Cs8900if_init()**함수는 네트워크 디바이스를 초기화 하는 함수이다.

4. 테스트 환경 및 결과

4.1 테스트 환경

본 논문은 ARM920T 기반의 S3C2440 32-Bit RISC MicroProcessor 를 탑재한 MBA2440 Evaluation Board 에 UbiFOS™를 탑재하여 테스트 하였다. 컴파일러는 ARM ADS 2.51 을 사용하였고 디버거는 OPENice32 A900 을 사용하였다. TCP/UDP 송수신 테스트를 위해 리눅스 서버를 사용하였다.



[그림 5] 테스트 환경

4.2 결과

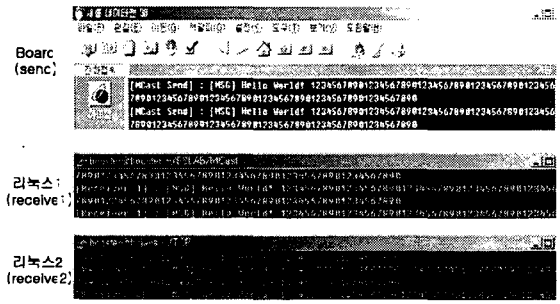
4.2.1 TCP/IP 프로토콜 스택 코드 크기

[표 1] 운영체제 와 TCP/IP 스택의 크기

	모듈	Code	Data	Code + Data
기존의 네트워크 스택	BSP	7KB	1KB	8KB
	Kernel	28KB	5KB	33KB
	Net	**47KB**	**4KB**	**51KB**
	Total	**82KB**	**10KB**	**92KB**
lwip 네트워크 스택	BSP	7KB	1KB	8KB
	Kernel	28KB	5KB	33KB
	Net	**30KB**	**2KB**	**32KB**
	Total	**65KB**	**8KB**	**73KB**

4.2.2 UDP 송수신

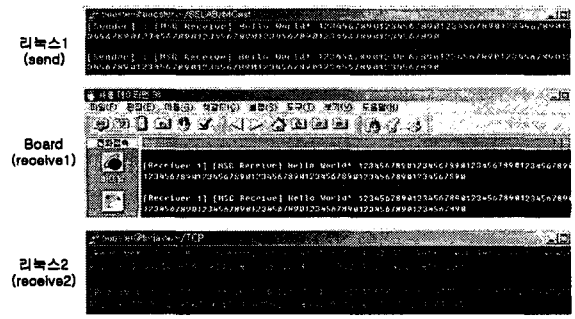
개발보드와 리눅스 서버간의 UDP 송수신 테스트를 하였다.



[그림 6] UDP 송수신 테스트

4.2.3 TCP 송수신

개발보드와 리눅스 서버간 TCP 송수신 테스트를 하였다.



[그림 7] TCP 송수신 테스트

5. 결론 및 향후 연구 과제

본 논문에서는 내장형 시스템을 위한 경량의 네트워크 스택을 설계 및 구현하였다.
향후 연구 과제로는 차세대 인터넷 프로토콜인 IPv6 지원과 HTTP / FTP / SNMP 등 사용자 어플리케이션 프로토콜을 추가적으로 구현하여 사용자 응용성을 향상시키는 것이다.

6. 참고 문헌

[1] http://www.aijisystem.com
[2] David Stepner 외 2 명, *"Embedded Application Design Using a Real-Time OS"*, IEEE, 1999
[3] C.M.Krishna, Kang G.Shin, *"Real-Time Systems"*,
The McGraw-Hill Companies, Inc.1997
[4] S.Deering, "Host Extensions for IP Multicasting", RFC 1112, 1989
[5] W.Fenner, "Internet Group Management Protocol, Version 2 , RFC 2236, 1997