실시간 운영체제 iRTOS상에서 시간결정성을 위한 메모리 할당 기법 설계 및 구현

The Design and Implementation of Memory Allocation Method for Time Determinism in iRTOS

박 세 영, 이 철 훈
충남대학교

Park se-young, Lee cheol-hoon
Chungnam National Univ.

요약

최근 임베디드 시스템이 발전함에 따라 시스템을 운영하는 방식이 단순한 펌웨어 수준에 그치지 않고 더 많은 서비스를 시스템에 제공하기 위해 운영체제의 사용이 증가하고 있다. 임베디드 시스템에는 제한적인 자원과 타깃시스템의 용도에 따라 실시간 운영체제(RTOS)가 주로 탑재된다. 실시간 운영체제 iRTOS는 가전, 무기체계 등에서 사용되며 현재 iRTOS가 채택하고 있는 메모리 할당 기법은 first fit 기법인데 대부분 시간결정성을 보장하지만 상황에 따라서 시간결정성을 보장하지 않을 수 있다. 따라서 시간결정성 보장을 향상시킬 수 있는 메모리 할당 기법이 필요하다. 본 논문에서는 실시간 운영체제 iRTOS에서 시간결정성을 보장할 수 있는 메모리 할당 기법을 설계하고 구현하는 것을 기술한다.

Ⅰ. 서론

최근 임베디드 시스템의 발전으로 이와 관련된 여러 산업들이 발전하고 있으며 특히 시스템의 핵심이라 할 수 있는 실시간 운영체제의 중요성이 부각되고 있다.

임베디드 시스템에 탑재되는 실시간 운영체제는 커널의 수행시간을 예측할 수 있기 때문에 시간결정성을 제공한다. 시간결정성은 시스템의 성능을 예측하게 할 뿐만아니라, 시스템의 전반적인 안정성에도 좋은 영향을 준다. 실시간 운영체제 iRTOS는 이러한 시간결정성을 보장하기 위해 메모리 할당 기법으로 First-Fit기법을 사용한다.

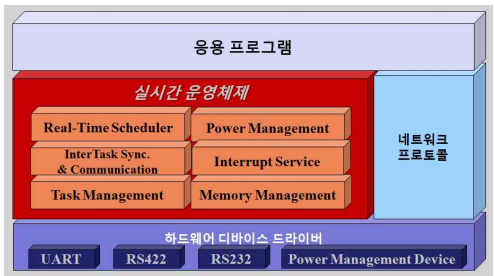
First-Fit기법은 메모리를 할당함에 있어 메모리 상에 존재하는 프리블록 중 할당할 메모리 크기를 만족하는 첫 번째 블록에서 메모리를 할당해 주는 기법으로 주소상에서 가장 앞에 위치한 프리블록을 항상 가리키고 있기 때문에 시간결정성을 보장한다. 하지만 할당할 메모리 크기를 만족하는 프리블록을 탐색함에 있어 그 시간이 상황에 따라서 길어질 수 있는 요지가 있으므로 시간결정성을 침해할 수 있다.

본 논문에서는 메모리 할당 시 first-fit기법의 사용으로 인한 오버헤드가 발생하는 것을 줄이는 방법이 될 수 있는 메모리 할당 기법에 대해 설계 및 구현하는 것을 기술한다.

Ⅱ. 관련 연구

1. iRTOS

실시간 운영체제 iRTOS는 멀티태스킹 환경을 지원하며, 태스크 생성에 필요한 메모리가 존재하면 생성 가능한 태스크의 수에는 제한이 없다. 모든 태스크에 0부터 255까지 256단계의 우선순위를 부여하여 우선순위 기반 선점형 스케줄러를 제공한다. 또한 동일한 우선순위에 대한 스케줄링 정책은 라운드-로빈 스케줄링을 사용한다. 임계 영역(Critical Region)이나 공유자원 사용을 위한 태스크간 동기화를 위해 세마포, 이벤트 플래그를 제공하며 메시지 큐, 메시지 포트, 메시지 메일박스 등의 태스크간 통신 기법을 지원한다. 또한 힙 메모리를 동적으로 할당하기 위해서 힙 스토리지 매니저와 메모리 풀을 지원한다[1].



▶▶그림 1. iRTOS 개념도

2. iRTOS 메모리 할당 기법

iRTOS는 힙 스토리지 매니저와 메모리 풀로써 메모리를 관리한다. 그 중 힙 스토리지 매니저는 가변길이의 메모리 영역을 할당하고 해제하는 기법으로써 메모리를 할당할 시 First-Fit으로써 프리블록 중 할당할 메모리 크기를 만족하는 첫 번째 블록을 할당한다. 첫 번째 프리블록을 가리키는 포인터를 가지고 있어 메모리 할당 시 시간결정성을 보장하지만 경우에 따라서 시간결정성을 저해하기도 한다는 단점이 있다[1].

Ⅲ. Worst-Fit의 설계 및 구현

1. Worst-Fit

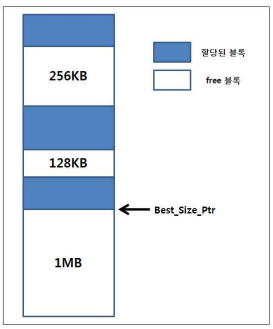
Worst-Fit기법은 메모리상의 프리 블록 중 크기가 가장 큰 블록에 메모리 공간을 할당해 주는 기법으로 가장 큰 프리 블록을 항상 가리키고 있으므로 메모리 할당 시의 오버헤드가 없다는 장점이 있다.

2. Worst-Fit의 구현

*i*RTOS 힙 스토리지 매니저의 메모리 할당 기법인 First-Fit은 첫 번째 프리블록을 가리키는 포인터를 두어 메모리 할당 시에 포인터가 가리키는 곳부터 차례대로 프리블록을 검사하며 타당한 메모리 블록을 찾는다. Worst-Fit에서는 첫 번째 프리블록을 가리키는 포인터 대신 가장 큰 프리블록을 가리키는 포인터를 둔다.

2.1 메모리 할당

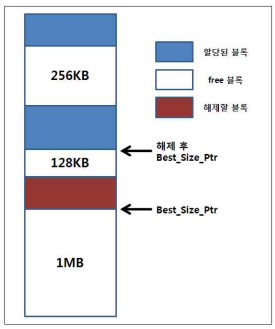
그림 2와 같이 메모리가 할당 되어 있다고 가정했을 때 128KB 크기의 메모리를 할당하는 상황에서는 비록 128KB의 프리블록이 있다하더라도 프리블록 중 가장 큰 블록인 1MB크기인 프리블록에 메모리를 할당한다. 메모리상에 가장 큰 프리블록을 Best_Size_Ptr이 항상 가리키고 있기 때문에 메모리 할당 시의 오버헤드를 없앨 수 있다.



▶▶ 그림 2. 메모리 할당 예시

2.2 메모리 반환

할당된 블록이 해제가 되면 해제 된 블록과 병합될 수 있는 프리블록이 존재하는지 검사한 뒤 해제 된 블록이 기존에 가장 큰 프리블록과의 크기를 비교한다. 기존의 프리블록보다 크기가 크면 Best_Size_Ptr을 새로 발생한 프리블록의 첫 번째 주소를 가리키게 하여 가장 큰 프리블록을 가리키는 포인터를 유지한다.

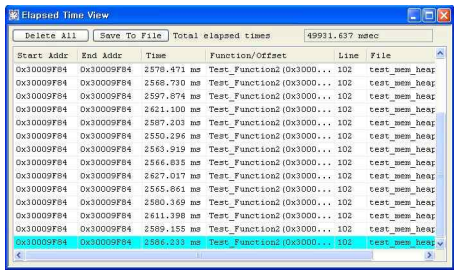


▶▶ 그림 3. 메모리 해제 예시

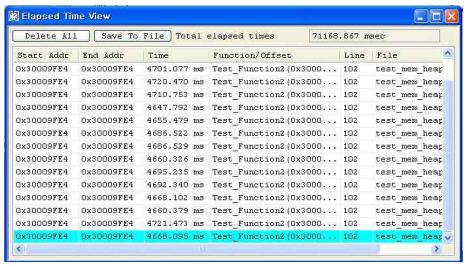
그림 3과 같이 붉은 블록이 해제될 때 위 128KB크기의 프리 블록과 아래 1MB크기의 프리블록과의 병합으로 인해 기존의 가장 큰 프리블록보다 큰 프리블록이 생성된다. 이 때 생성된 프리블록과 기존의 가장 큰 프리블록을 비교하여 Best_Size_Ptr을 교체한다.

Ⅳ. 테스트 환경 및 결과

본 논문은 MBA2440 Evaluation Board에 *i*RTOS를 탑재하여 구현하였으며, 컴파일러는 Metrowerks사의 Code Warrior를 사용하였다.



▶▶ 그림 4. First-Fit 사용 시 할당 해제 시간 측정



▶▶ 그림 5. Worst-Fit 사용 시 할당 해제 시간 측정

그림 4, 5는 각각 First-Fit과 Worst-Fit을 이용하여 메모리를 할당하고 해제한 후 수행시간을 측정한 것이다. Worst-Fit의 수행시간이 First-Fit에 비해 상대적으로 길게 측정되었는데, 메모리 해제 시에 가장 큰 프리블록을 지정해주는 시간이 First-Fit의 해제시간 보다 길기 때문이다.

Ⅴ. 결론 및 향후 연구 과제

본 논문에서는 임베디드 시스템에서 주로 사용되는 실시간 운영체제 *i*RTOS 힙 스토리지 매니저의 메모리 할당 기법 First-Fit의 단점인 프리블록 탐색 시 오버헤드를 줄이기 위해 Worst-Fit적용을 설계하고 구현한 내용을 기술하였다.

Worst-Fit기법에서 메모리 해제 시에 가장 큰 프리블록을 설정하는 과정에서 오버헤드가 발생하여 기존의 수행시간보다 더 길게 측정되었다.

향후 연구 과제로 Worst-Fit기법 사용 시 문제가 되었던 메모리 해제 시 오버헤드를 줄일 수 있는 연구가 진행되어야 한다.

■ 참 고 문 헌 ■

[1] *i*RTOS User's Guide
[2] Li, Qing and Yao, Caroline, Real-time concepts for embedded systems, 에이콘출판사, 의왕, 2004.
[3] 장용희, 권용진, 임베디드 시스템을 위한 RTOS, 홍릉과학출판사, 서울, 2006.
[4] 양희권, 박윤미, 류현수, 이철훈 "실시간 운영체제의 태스크 사용시간 측정 방법 구현", 한국정보과학회 학술발표논문집, 제31권, 제1호(A), pp. 166 ~ 168, 2004. 4
[5] MBA2440 board manual KOR v2.1, Aiji System, 2006