내장형 시스템에서의 향상된 가상 메모리 관리 방법에 관한 연구

신성룡○ 허 신
한양대학교 대학원 컴퓨터공학 운영체제 및 네트워크 응용연구실, 삼성전자 디지털 프린팅 사업부
sr.shin@samsung.com, shinheu@cse.hanyang.ac.kr

A Research on the enhanced virtual memory management for Embedded System

Sungryong Shin (Supervised by Prof. Shin Heu)
Dept. of Computer Science and Engineering Graduate School of Hanyang University

요 약

Windows CE .NET 내장형 시스템 환경에서 응용 프로그램을 작성하거나 실행하고자 할 때 가장 문제가 되는 것은 메모리 관리이다. Windows CE .NET 시스템은 메모리가 작기 때문에 때로는 전체 성능을 위해 메모리가 절약되는 방식으로 관리되어야 한다. 본 논문은 이와 같이 응용 프로그램이 갖는 가상 메모리의 사용 한계를 운영체제의 커널 수준에서 극복하기 위한 것이다. 현재의 Windows CE .NET 커널이 갖는 가상 메모리 관리의 단점을 극복하고 보다 많은 가상 메모리가 사용될 수 있도록 페이지 할당 및 관리를 담당하는 커널의 메모리 관리 루틴에 있어 새로운 알고리즘을 적용하여 효율을 높였으며 응용 프로그램 작성을 통한 실험을 통하여 가상 메모리의 할당 횟수를 늘일 수 있었다. 그리고 전체적인 메모리 관리 시스템의 성능 향상과 시스템의 안정성을 높일 수 있었다.

1. 서 론

내장형 시스템의 모바일 디바이스 시장이 꾸준한 성장세를 유지하면서 개발자들의 관심을 끌고 있다. 모바일 컴퓨팅 장비와 솔루션들이 기업의 생산성 향상을 위한 주요 도구로 자리잡아 가면서 21세기 주요 화두로 등장한 Embedded 프로그래밍의 주가 또한 동반 상승하고 있는 추세이다.

이러한 내장형 시스템의 운영체제로서 최근 각광을 받고 있는 것이 Microsoft에서 제공한 Windows CE .NET 운영체제이다. 이 운영체제는 기존의 32bit Windows OS와 호환성이 있기 때문에 사용자가 사용하기는 쉬우나 하드웨어의 사양이 높아져서 다른 운영체제에 비해 단점이었으나 최근 하드웨어의 눈부신 발달로 많은 환경에서 사용이 되고 있는 추세이다.

Windows CE .NET 운영체제는 기존 32bit 운영체제를 따로고 있기 때문에 작은 크기를 필요로 하는 내장형 시스템에 적용 시 여러 문제가 발생할 수 있으나 그 중에 대표적인 것이 메모리를 관리하는 것이다.

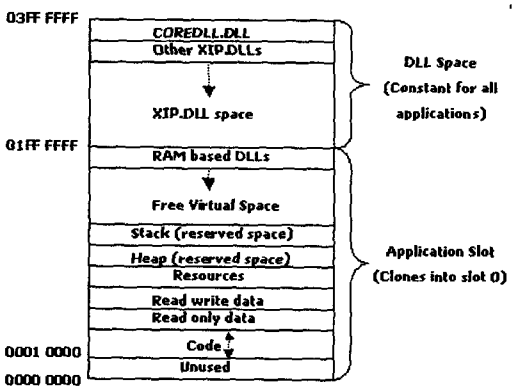
Windows CE .NET은 페이지 기반의 가상 메모리 관리 시스템을 구현하고 있으며 이 가상 메모리는 분리된 가상의 주소 공간으로 하드웨어가 제공하는 실제 메모리 주소 공간과는 다른 것이다. 운영체제는 메모리 관리 유닛(MMU: Memory Management Unit)을 통하여 가상 주소 공간을 실제 주소 공간으로 변환한다. Windows CE .NET에서의 페이지 크기는 마이크로 프로세서의 종류에 따라 1024부터 4096 바이트 까지 사용 된다. [1]

본 연구는 내장형 시스템의 페이지 기반의 가상 메모리 관리 시스템에서 Windows CE .NET이 가지는 장점들과 일부 취약한 가상 메모리의 할당 및 관리 부분의 성능 향상을 통하여 전체 시스템의 성능 향상에 초점을 맞추었다.

2. 본 론

2.1 Windows CE .NET 주소 공간 [2]

Windows CE .NET은 모든 응용 프로그램을 하나의 단일 2GB 가상 주소 공간에서 실행하며 한 응용 프로그램의 메모리 공간은 다른 응용 프로그램이 접근할 수 없도록 보호한다. 다음 그림은 Windows CE .NET 응용 프로그램의 가상 메모리 맵을 보여준다.



(그림 1) Windows CE .NET Application Virtual Memory Space [3]

응용 프로그램의 코드는 가상 주소 0x10000(64KB)에서 시작하며 응용 프로그램이 실행될 때 모든 코드를 위한 충분한 공간이 주소 공간에 예약된다. 실제 코드는 필요로 할 때마다 주소 공간으로 요구 페이징 된다. 코드를 위한 공간 위에는 읽기

전용과 읽기/쓰기 데이터를 위해 예약되며 그 위에는 로컬 힙과 응용 프로그램에서 실행되는 쓰레드를 위한 스택 영역이 할당된다. 분리된 힙을 만들거나 VirtualAlloc API를 호출하는 등의 추가적인 메모리를 필요로 하는 경우 영역을 만족시키는 첫 번째 유휴 공간을 찾아 아래서부터(Bottom Up)할당을 하게 된다.

대표적인 Windows CE .NET 운영체제에 대한 진입점이 들어 있는 DLL은 Coredll.dll로서 함수들에 대한 진입점이 이 DLL에 결합되어 있다. 만약 시스템 DLL이 아닌 일반 DLL이 적재될 경우는 32MB 경계 안에 위치하게 된다.

2.2 가상 메모리 [4]

가장 기본적인 메모리의 유형으로서 본 논문에서 다루고자 하는 메모리이다. Windows CE .NET은 Memory Management Unit(MMU)을 전적으로 사용하는 가상 메모리 기반의 운영 체제이다. 운영체제는 힙, 스택 등에 메모리를 할당할 때 가상 메모리 API를 사용하여 메모리를 할당한다. 가상 메모리를 할당하는 동안 커널은 유휴 물리적 메모리를 검색하여 프로세스의 가상 주소 공간에 메모리를 매핑 한다. 가상 메모리는 64KB 경계로 할당되기 때문에 작은 공간의 메모리 할당 시 많은 조각(Fragmentation)들이 발생하게 된다. 그러므로 가상 메모리를 할당하는 데 있어 영향을 최소화하기 위해서 프로세스는 일반적인 처리를 하기 전에 모든 할당을 완료하는 것이 좋다.

2.3 Windows CE .NET 응용 프로그램 (Knowledge Pad)

2.3.1 개요

본 논문에서는 현재의 Windows CE .NET 운영 체제의 가상 메모리 관리의 문제점을 알아보기 위해 Windows CE .NET 환경에서 응용 프로그램을 작성하였다.

본 논문의 응용 프로그램은 저자가 직접 개발한 것으로 Windows CE .NET 가상 메모리를 이용하는 가칭 'Knowledge Pad'라 명명했으며 이 응용 프로그램의 기능은 PDA 환경에서 사용자가 언제든지 문서를 열어 편집 및 간단한 메모를 할 수 있으며 무선 네트워크 환경을 통하여 서버로 전송하여 실시간으로 다른 사용자와 파일을 주고 받을 수 있는 편리한 도구이다. 응용 프로그램의 주요 기능은 다음과 같다.

<표 1>응용 프로그램의 주요 기능

기능	설명
문서 열기/보여 주기	무선 PDA의환경에서 문서를 열어서 화면상에 보여 주는 기능
문서 편집	PDA의 터치 스크린 기능을 이용 메모를 할 수 있는 기능
문서 저장	편집한 문서를 버튼을 이용 저장 기능
문서 축소	화면의 크기에 맞게 축소를 하는 기능
문서의 서버 전송	편집한 문서를 TCP 연결을 이용 접속 가능한 서버로 전송할 수 있는 기능

가상 메모리를 사용하기 위하여 Windows CE .NET의 주요 메모리 관리 함수인 **VirtualAlloc** 함수를 사용한다.

2.3.2 응용 프로그램의 가상 메모리 할당의 문제점

VirtualAlloc 함수를 이용하여 가상 메모리를 할당하는 방법은 Windows XP 응용 프로그램이나 유사한 운영체제에서 흔히 사용되는 방법이나 이 방법은 Windows CE .NET 응용 프로그램에서는 좋은 방법이 되지 못한다. 응용 프로그램에서 쓰인 간단한 예제의 가상 메모리 할당 방법을 보자. [3]

```
PVOID pMem[512];
for (i = 0; i < 512; i++) {
  pMem[i] = VirtualAlloc (0, PAGE_SIZE, MEM_RESERVE |
MEM_COMMIT, PAGE_READWRITE);
}
```

이 코드는 한 개의 페이지 크기의 할당을 512회 호출한다. Windows CE .NET 운영체제에서의 이 코드는 수 메가 바이트의 RAM 영역이 유휴 상태여도 항상 실패하게 된다. 문제는 Windows CE. NET을 포함한 Win32 운영체제가 어떻게 메모리를 예약하는가 하는 데 있다. 가상 메모리 주소 공간을 예약할 때 항상 64KB 경계를 단위로 예약을 하게 되어 있다. 그러므로 512개의 메모리 블록을 예약 및 할당하기 위해서는 512개의 64KB 주소 공간이 필요하다. 여기서 문제는 Windows CE .NET 응용 프로그램은 32MB 가상 메모리 공간 내에서만 사용이 가능하다는 것이다. 이 공간 내에서 응용 프로그램 코드, 로컬 힙, 스택과 응용 프로그램에 의해 적재되는 DLL들이 존재하기 때문에 512개의 64KB 메모리 블록은 할당되지 못하고 약 471회의 호출 뒤에 실패를 하게 된다.

본 논문은 Windows CE .NET 운영 체제의 커널 내에서 효과적인 가상 메모리 관리 기법을 도입하여 이러한 제약을 극복하고자 하는 것이다. 다음 장에서 본 논문에서 제시하는 가상 메모리 할당 기법에 대해 설명하고자 한다.

2.4 제안하는 가상 메모리 관리 기법

2.4.1 개요

본 논문에서는 2.3.2에서 제시했던 문제점을 해결하기 위하여 가상 메모리 관리를 응용 프로그램 수준에서 코드의 알고리즘을 통해 할 것이 아니라 Windows CE .NET 운영체제의 커널의 메모리 관리 루틴에 새로운 알고리즘을 적용하여 보다 운영체제의 효율을 높이며 사용할 수 있는 가상 메모리의 할당 횟수뿐만 아니라 실제로도 많은 가상 메모리가 할당될 수 있도록 해주는 것이다.

2.4.2 Windows CE .NET Shared Source Code [5]

본 논문에서 제시하는 알고리즘을 적용하기 위하여 기본 커널 소스는 Windows CE .NET evaluation edition 4.1에 제공된 추가 소스 코드를 이용하였다. 이 소스는 Windows CE .NET platform을 이해하기 위해 Microsoft에서 제공된 것으로 이 소스의 수정으로 커널의 일부 기능을 수정할 수 있으며 비상업적인 용도로만 사용이 가능하다.

2.4.3 제안하는 알고리즘

2.4.3.1 개요

하나의 응용 프로그램에서 사용할 수 있는 가상 메모리의 공간은 1개의 섹션(32MB)으로서 연속적인 페이지 할당 시 64KB 경계로 페이지가 할당되게 된다. 이 경우 한번의 VirtualAlloc 호출로 1개의 메모리 블록이 할당되게 되며 이와 같은 식으로 할당이 되면 총 512개의 메모리 블록 중 프로그램 코드 및 공통 DLL 영역을 제외한 약 471회 정도의 VirtualAlloc 호출만 가능하게 된다.

현재의 메모리 블록 검색 알고리즘은 16개의 페이지를 포함하고 있는 한 개의 블록에 대한 사용 유무를 단지 NULL_BLOCK(0)과 RESERVED_BLOCK((MEMBLOCK*)1)으로만 표기를 하고 있다. 16개의 페이지 내에 단 하나의 페이지만 할당이 되어도 전체 메모리 블록은 RESERVED_BLOCK 상태가 되어 해당 블록은 다른 VirtualAlloc

호출에 의해 할당이 되지 못한다. 이와 같은 페이지의 낭비를 막기 위하여 메모리 블록에 대한 정보 및 관련 변수를 다음과 같이 추가하였다.

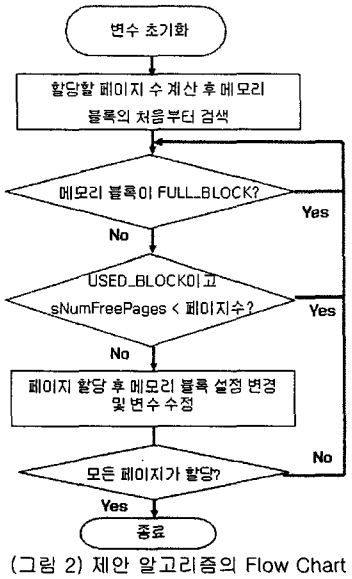
메모리 블록 정의 추가:
NULL_BLOCK (메모리 블록이 완전히 비어 있는 경우)
USED_BLOCK (일부 페이지가 할당되어 있는 경우)
FULL_BLOCK (메모리 블록이 완전히 꽉 찬 경우,
빠른 검색을 위해 추가함)
페이지 정의 추가:
#define FREE_PAGE (~0x0fUL) (BAD_PAGE 정의를 수정)
메모리 블록 Structure 추가
short sNumFreePages; (해당 메모리 블록에 몇 개의 유휴 페이지가 있는 지를 보여주는 변수)

위와 같이 정의된 구조를 통한 가상 메모리의 할당 과정은 다음 장에서 살펴 본다.

2.4.3.2 자료/제어 흐름도(Data and Control Flow Chart)

(그림 2)에서 보여주는 것과 같이 VirtualAlloc 함수 호출 시 인자(Parameter)로 넘겨 받은 페이지 크기로부터 4KB의 페이지가 몇 개가 필요한지 계산한 후에 현재 활성화된 섹션의 첫 번째 메모리 블록부터 검색을 한다. 현재의 메모리 블록의 Flag(NULL_BLOCK, USED_BLOCK, FULL_BLOCK) 및 변수(sNumFreePages)값에 따라 다음과 같은 처리를 한다.

NULL_BLOCK: 필요한 페이지가 이 블록에 모두 할당되면 할당 후 USED_BLOCK 또는 FULL_BLOCK 설정 및 sNumFreePages 변수를 갱신한다. 아직 할당할 페이지가 있으면 다음 블록을 검색한다.



(그림 2) 제안 알고리즘의 Flow Chart

USED_BLOCK: 할당 요청된 페이지 수와 현재 블록이 할당 가능한 페이지 수를 비교하여 현재 블록에 모두 할당 가능하면 할당 후 Flag 및 sNumFreePages 변수를 갱신한다. 현재 블록에 요청된 페이지 전부가 할당 가능하지 않으면 다음 블록을 검색한다. 이 경우 현재 블록의 남은 페이지 수만큼 할당하지 않은 이유는 이 과정의 오버헤드가 더 커져서 할당 속도를 느리게 할 뿐 아니라 블록들 사이를 연결해주는 과정도 복잡해지기 때문에 이와 같이 처리하였다. 그리고 다음 가상 메모리 할당의 호출 시 현재 블록에 크기에 맞는 페이지가 할당될 수 있기 때문에 이와 같이 하는 것이 운영체제가 단순하면서 빠르게 동작하도록 하는 데 이득이 된다.

FULL_BLOCK: 이 경우 빠른 속도로 다음 블록을 검색

이와 같은 알고리즘으로 처리하면 기존의 64KB 경계로 가상 메모리가 할당되어 연속적인 VirtualAlloc 함수 호출 시 약 471회 정도만 할당이 되던 것이 메모리 블록 내의 남은 페이지에도 할당이 되게 되어 요구하는 페이지의 크기에 따라 최소 1배(64KB 할당 요구 시)에서 최대 16,000배(4KB이하 할당 요구 시)까지 횟수가 증가하게 된다.

2.5 실험 및 테스트 결과

개선된 알고리즘이 반영된 운영체제 이미지를 MS Platform Builder 4.0을 이용하여 USB 포트를 통해 Samsung S3C2410 보드에 다운로드하여 기존 운영체제와의 비교 실험을 하였다.

<표 2> 운영체제 메모리 할당 관련 테스트 결과

응용 프로그램 기능	기존 운영체제			제안하는 커널 적용 운영체제		
	시도 횟수	성공 횟수	성공 률	시도 횟수	성공 횟수	성공 률
문서 열기/보여주기	20	20	100%	20	20	100%
편집 문서 서버 보내기	20	20	100%	20	20	100%
연속된 512 회의 가상 메모리 할당	20	0	0%	20	20	100%

3. 결 론

실험을 통해 제안하는 방법이 기존의 방법 대비 향상된 가상 메모리 할당 성공률을 보였고 보다 많은 가상 메모리를 응용 프로그램이 쓸 수 있음을 확인하였다. Windows CE .NET 운영체제의 전체적인 메모리 관리 성능을 향상시키기 위해서는 가상 메모리 관리 모듈 뿐만 아니라 힙, 스택, 정적 데이터 등 관련 메모리 관리 모듈도 연관되어 성능이 향상되어져야 하나 본 논문에서는 Windows CE .NET에서 Shared Source Code를 이용한 가상 메모리 관리 모듈의 개선 과정을 보임으로써 다른 분야에 있어 성능을 향상하고자 하는 개발자에게 충분한 사례를 보이고 있다. 지속적인 연구를 통하여 다른 운영체제를 사용하는 내장형 시스템에도 긍정적인 영향을 끼칠 것이다.

참 고 문 헌

[1] Douglas Boling, " Programming Microsoft Windows CE" , Microsoft. Press., 2002
[2] MSDN, " How Windows CE .NET is Designed for Quality of Service" , Microsoft. Inc., February 2003
[3] MSDN, " Windows CE .NET Advanced Memory Managements" , Douglas Boling, Windows Embedded MVP, August 2002
[4] James Y. Wilson, Aspi Havewala, " Building Powerful Platforms with Windows CE" , ADDISON-WESLEY-, 2001
[5] Mike Mitchell, " Using Shared Source Code in Microsoft Windows CE .NET" , Microsoft. Inc., Updated January 2002