# 무선인터넷 플랫폼을 위한 메모리 관리 모듈 설계 및 구현

[0]유용덕, 김연수, 임형택, 강민구, 최훈
충남대학교 컴퓨터공학과
e-mail : {[0]ydyu, kimys, htlim, mkkang, hchoi}@ce.cnu.ac.kr

# Design and Implementation of Memory Management Module for a Wireless Internet Platform

[0]Yong-Duck Yoo, Youn-Soo Kim, Hyoung-Taek Lim, Min-Koo Kang, Hoon Choi
Dept. of Computer Engineering , Chungnam National University

**요 약**

현재 각 이동통신사별 무선인터넷 플랫폼 환경을 통일하는 WIPI 가 상용화되었다. 컨텐츠 개발자에게 있어서는 시장에 대한 경쟁력을 가지기 위하여 개발 비용 감소 및 개발 편의성을 제공하는 WIPI 에뮬레이터가 필요하다. 그러나 현재 개발된 WIPI 에뮬레이터들의 메모리 관리 모듈은 특정 운영체제에 의존적이다. 본 논문에서는 다른 운영체제로의 높은 이식성과 효율적인 메모리 관리를 제공하는 WIPI Clet 에뮬레이터 2.0 메모리 관리모듈의 설계 및 구현 사항을 기술한다. WIPI Clet 에뮬레이터 실행엔진에 구현된 메모리 관리 모듈은 동적 메모리 할당, 해제, 압축 등의 기능을 제공하는데 휴대폰과 같은 적은 메모리 장치에 탑재될 수 있도록 경량 알고리즘으로 구현하였다.

## 1. 서론

현재 국내 무선인터넷 시장은 유선인터넷의 발달과 더불어 최근 5 년 동안 급속도로 성장하고 있다. 하지만 각 이동통신사들이 자사에서 정한 무선인터넷 환경과 무선인터넷을 이용한 응용프로그램 실행환경을 사용함으로써 중소 규모의 컨텐츠 제공업자와 소비자들은 이동통신사간 컨텐츠의 호환이 불가능한 상황에 직면하게 되었다. 그 결과로 컨텐츠 제공업자는 같은 컨텐츠를 여러 개의 Target 플랫폼에 맞게 개발해야 하는 중복투자를 해야 하고, 소비자들은 이동통신사별로 컨텐츠가 한정되어 이용 가능한 서비스의 종류가 제한되는 불합리한 형태가 되었다. 이를 해결하기 위해 국내에서는 2001 년 10 월 초 SK 텔레콤, KTF, LG 텔레콤 등 3 개 이동통신사업자가 표준화를 위한 구체적인 요구사항을 제시하고, 한국전자통신연구원(ETRI)와 한국무선인터넷표준화포럼의 노력으로 각 이동통신사별 무선인터넷 플랫폼 환경을 통일하는 WIPI(Wireless Internet Platform for Interoperability)가 제정되었다[1]. 현재 2004 년 2 월에 버전 2.0 이 표준화된 상태이며[2], 이동통신사별로 단말기에 WIPI 의 이식 및 컨텐츠 개발이 진행 중이다.

WIPI 컨텐츠 개발자에게 있어서 시장에 대한 경쟁력을 가지기 위하여 새로운 형태의 응용프로그램 개발에 대한 개발 기간 및 비용을 보다 감소시킬 필요가 있다. 이러한 요구에 부응하기 위하여 개발 비용의 감소 및 개발의 편의성을 제공하는 WIPI 에뮬레이터와 같은 응용프로그램 개발환경이 필요하다[3,4].

본 연구팀은 2003 년 12 월 윈도우 실행환경에서 WIPI Clet 응용프로그램을 개발 및 실행시킬 수 있는 'WIPI Clet 에뮬레이터 1.0'을 개발하였다[4]. 그러나 WIPI Clet 에뮬레이터 1.0 은 응용프로그램의 수행을 위한 메모리 관리에 있어 윈도우 실행환경에 의존적인 Win32 API 를 이용하여 구현함에 따라 메모리를 효율적으로 관리할 수 없으며, 또한 리눅스(Linux)나 유

[0] 본 연구는 한국전자통신연구원 "윈도우용 Clet 에뮬레이터 개발과제"의 지원을 받아 수행되었음

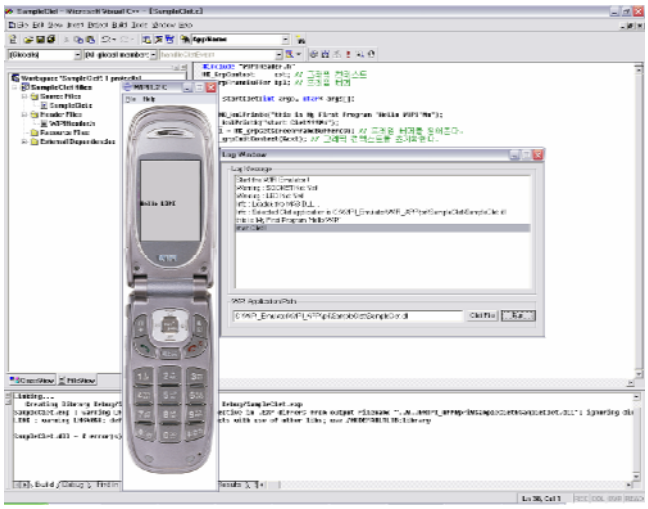
닉스(Unix) 같은 다른 운영체제에서는 실행될 수 없다. 따라서 본 연구팀은 WIPI 응용프로그램의 실행을 위한 메모리 관리에 있어 윈도우 실행환경에 의존적이었던 부분을 개선한 WIPI Clet 에뮬레이터 2.0 을 개발중에 있다[3].

WIPI Clet 에뮬레이터 2.0 은 버전 1.0 과는 달리 메모리 관리를 위해 Win32 API 를 사용하지 않으며, 또한 본 연구에서 새롭게 제안하는 운영체제에 독립적인 메모리 관리 기법을 사용함으로써 효율적으로 메모리를 관리할 수 있으며, 적은 비용으로 다른 운영체제로의 높은 이식성을 제공한다.

논문의 2 장에서는 본 연구팀에서 2003 년 12 월에 개발한 WIPI Clet 에뮬레이터 1.0 에 대해서 전반적으로 살펴보고[4], 3 장에서는 운영체제에 의존적이었던 WIPI Clet 에뮬레이터의 메모리 관리 모듈과 본 논문에서 새롭게 설계 및 구현한 메모리 관리 모듈에 대해서 기술한다[3]. 마지막으로 4 장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

## 2. WIPI Clet 에뮬레이터 1.0

WIPI Clet 에뮬레이터 1.0 은 윈도우 컴퓨터상에서 WIPI 의 C API 를 사용하여 만들어진 응용프로그램인 Clet 을 실행할 수 있는 기본적인 실행지원환경으로서, WIPI 응용프로그램을 일반 PC 환경에서 작성한 후 에뮬레이터에서 실행함으로써 응용프로그램을 실제 단말기에 적용하기 전에 동작을 확인하고 검토할 수 있어 응용프로그램의 개발 비용 감소 및 개발의 편의성을 제공한다. 본 연구팀에서 개발한 윈도우용 WIPI Clet 에뮬레이터 1.0 의 실행 모습은 다음과 같다.



**[그림 1] WIPI 에뮬레이터 1.0 실행화면**

개발한 WIPI Clet 에뮬레이터 1.0 의 검증은 한국전자통신연구원과 EXEmobile㈜에서 제공한 PCT(Platform Certification Toolkit)를 이용하였다.

검증 결과 PCT 자체의 테스트 항목 오류로 인한 에러항목(2 항목)과 실제 단말기 환경이 아닌 윈도우 에뮬레이터 환경요인으로 인한 NR(Not Run) 항목(6 항목)을 제외하고 모두 통과하였다(<표 1>).

따라서 본 연구팀에서 개발한 WIPI Clet 에뮬레이터 1.0 은 WIPI 표준 규격을 준수하며, 기존 Clet 응용프로그램을 PC 환경에서 성공적으로 실행하였다.

**<표 1> PCT 를 이용한 검증 결과**

| 결과 / 항목 | P | F | E | NR | RO | Total |
|---|---|---|---|---|---|---|
| 커널 API | 56 | 1 | 0 | 1 | 16 | 74 |
| 그래픽 API Batch | 41 | - | - | 1 | - | 42 |
| 그래픽 API Interactive | 28 | - | - | 1 | 18 | 47 |
| 데이터베이스 API | 50 | 1 | - | - | - | 51 |
| 파일시스템 API | 36 | - | - | - | - | 36 |
| 매체처리기 API | 8 | - | - | 2 | - | 10 |
| 시리얼 API | 7 | - | - | - | - | 7 |
| Phone API | 1 | - | - | - | - | 1 |
| Utility API Batch | 7 | - | - | - | - | 7 |
| Utility API Interactive | 1 | - | - | - | 2 | 3 |
| UIC API Batch | 40 | - | - | - | 1 | 41 |
| UIC API Interactive | 16 | - | - | 1 | - | 17 |

{P: Pass, F: Failed, E: Error, NR: Not Run, RO: Result Only}

## 3. 메모리 관리 모듈

### 3.1 WIPI Clet 에뮬레이터 메모리 관리 모듈

WIPI Clet 에뮬레이터 1.0 에서의 메모리 관리 모듈은 WIPI 응용프로그램 개발자로 하여금 응용프로그램 개발에 있어 응용프로그램이 사용하는 힙(heap) 메모리를 실시간적으로 파악할 수 있게 함에 따라 효율적인 개발환경을 제공한다. 이러한 기능을 제공하기 위해서 메모리 사용량의 모니터링을 위한 버추얼 맵(virtual map)을 고안하여 사용하였다. 다음 <표 2>와 같이 버추얼 맵을 위한 사용 구조체를 정의하였다.

**<표 2> 버추얼 맵 구조체**

| 필드 / 값 | Type | Value |
|---|---|---|
| M_MID | M_Uint32 | 파일 식별자 |
| mBufID | M_Uint32 | 공유 버퍼 식별자 |
| m_start | M_Uint32 | 힙 내 파일 및 공유 메모리 시작 주소 |
| m_end | M_Uint32 | 힙 내 파일 및 공유 메모리 종료 주소 |
| m_size | M_Uint32 | 파일 및 공유 메모리 크기 |
| address | M_Char* | 현재 액세스하는 주소 |
| Flag | M_Boolean | 구조체 사용 여부 |

구현한 버추얼 맵은 플랫폼에서 설정한 힙 영역의 사용 상태를 보여주기 위한 도구로서, 힙 메모리의 동적 할당 시, 버추얼 맵 테이블에 동적 할당 정보를 기록하고, 기록된 정보를 이용하여 실시간적으로 사용되는 힙 메모리의 양을 보여준다.

### 3.2 운영체제에 독립적인 메모리 관리 모듈

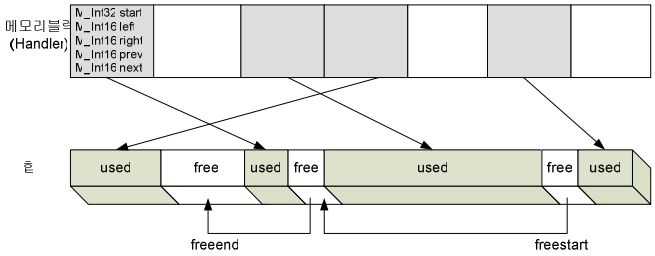
본 연구에서 새롭게 설계 및 구현한 '운영체제에 독립적인 메모리 관리 모듈'은 WIPI Clet 에뮬레이터 1.0 의 메모리 관리 모듈에서 제공하였던 메모리의 고정할당과 동적할당을 지원하며, 동적으로 할당된 메모리의 압축을 지원한다. 또한 메모리의 할당, 해제 및 압축에 있어 Win32 API 를 사용하지 않는 등 운영체제에 독립적인 메모리 관리 기능을 제공한다.

㈎ 메모리 자료구조

다음은 WIPI Clet 에뮬레이터 2.0 메모리 관리 모듈을 위해 정의한 자료구조<표 3> 및 구현방법([그림 2])이다.

**<표 3> 메모리 관리를 위한 구조체**

| | 비고 |
|---|---|
| memblock.start | 메모리 블록에 해당하는 힙의 시작주소 |
| memblock.left | 해당 블록의 왼쪽에 인접한 블록아이디 |
| memblock.right | 해당 블록의 오른쪽에 인접한 블록아이디 |
| memblock.prev | 이전 미사용 블록아이디 |
| memblock.next | 다음 미사용 블록아이디 |
| freesize | 총 미사용 메모리 사이즈 |
| freestart | 미사용 메모리 블록리스트의 시작 아이디 |
| freeend | 미사용 메모리 블록리스트의 끝 아이디 |
| startblock | 힙 상의 첫 블록아이디 |




**[그림 2] 메모리 관리 모듈 자료구조**

㈏ 동적 메모리할당

메모리는 플랫폼이 사용할 휘발성 메모리의 크기 및 시작번지를 얻어오는 MH_sysGetHeapBlock API 를 통해 얻어진 고정된 메모리로부터 할당되며, 응용프로그램의 실행을 위한 메모리를 할당하기 위해서는 WIPI 커널 API 에서 정의된 MC_knlAlloc(), MC_knlCalloc() API 를 이용한다. 이 두 API 를 실행하여 메모리를 할당하는 방법은 다음과 같다.

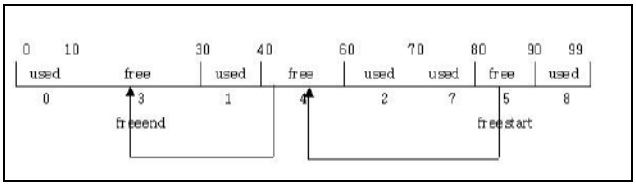
① 총 메모리 공간이 충분한지 확인한다. 충분하지 않다면 메모리를 할당할 수 없다는 오류 메시지를 리턴한다.

② 미사용 메모리 블록 리스트를 검색하여 할당 가능한 미사용 메모리 블록을 찾는다. 미사용 메모리 블록 리스트의 끝까지 검색한 후, 할당할 수 있는 미사용 메모리 블록을 찾을 수 없으면, 메모리 압축을 수행한 후 다시 할당을 시작한다.

③ 할당할 메모리 크기와 찾은 메모리 블록의 크기가 같다면 그 블록 전체에 모두 할당한다. 이 경우에는 메모리 블록의 개수는 늘어나지 않고, 단지 그 할당된 메모리 블록의 next 값을 '-1'로 바꾸어 할당된 영역이라고 표시를 해준 후 할당된 블록을 미사용 메모리 블록 리스트에서 제거한다.

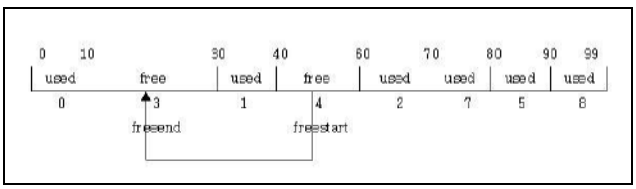
④ 할당할 메모리 크기와 찾은 메모리 블록의 크기가 다를 경우, 찾은 메모리 블록에서 할당을 수행한 후 나머지의 미 할당 영역은 그대로 미사용 메모리 블록 리스트의 영역으로 유지한다. 즉, 블록이 두 개로 나누어지므로 메모리 블록 하나가 추가된다.

***메모리할당 예**: [그림 3]의 메모리 사용 상황에서 크기 '10'의 메모리를 할당할 경우




**[그림 3] 메모리할당 전**

[그림 3]에서 새롭게 메모리를 할당할 경우, 미사용 메모리 블록 리스트의 시작에서부터 할당할 공간을 탐색하므로 미사용 메모리 블록 5 에 할당되며, 선택된 미사용 메모리 블록의 크기가 할당하려는 메모리 크기인 '10'과 같으므로 할당 후에 미사용 메모리 블록은 리스트에서 제거된다[그림 4].




**[그림 4] 메모리할당 후**

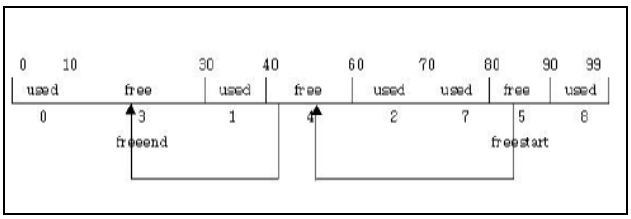
㈐ 동적 메모리해제

사용 중인 메모리를 해제하기 위해서는 WIPI 커널 API 중 MC_knlFree()를 사용하며, 해제시킬 블록의 인접한 블록의 next 값을 검사하여 통합 가능 여부에 따라 다음 4 가지 경우로 처리한다.

① **해제시킬 메모리 블록의 양 옆 메모리 블록이 미사용 블록인 경우**: 양 옆의 미사용 메모리 블록과 현재 해제시킬 메모리 블록이 합쳐져 하나의 미

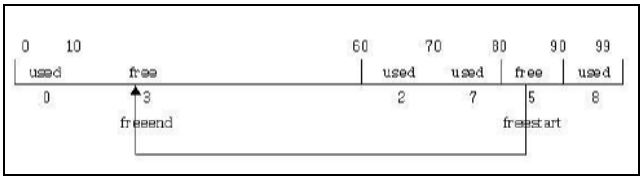
사용 블록이 된다. 하나의 미사용 블록이 된 후에 블록의 아이디는 합쳐진 3 개의 블록 중 가장 왼쪽에 있는 블록 아이디를 가진다.

***사용 메모리해제 예**: [그림 5]의 메모리 사용 상황에서 메모리 블록 '1'이 해제될 경우




**[그림 5] 사용 메모리 해제 전**

[그림 5]에서 미사용 메모리 블록 1, 4 는 없어지고, 메모리 블록 3 에 합쳐지며, 결과는 [그림 6]과 같다.




**[그림 6] 사용 메모리 해제 후**

② **왼쪽 메모리 블록만이 미사용 메모리인 경우**: 해제될 메모리 블록은 왼쪽의 미사용 메모리 블록과 합쳐져 하나가 되는 것이므로 메모리 사용 정보를 기록하고 있는 'memblock'에서 제거된다.

③ **오른쪽 메모리 블록만이 미사용 메모리인 경우**: 해제될 메모리 블록은 오른쪽의 미사용 메모리 블록과 합쳐져 하나가 되는 것이므로 오른쪽 미사용 메모리의 시작 주소값을 수정한 후, 해제될 메모리 블록은 메모리 사용 정보를 기록하고 있는 'memblock'에서 제거된다.

④ **양쪽 메모리 블록 모두 사용 중인 경우**: 해제될 메모리 블록을 미사용 메모리 블록이라고 표시만 해준다. 즉 해제될 메모리 블록을 미사용 메모리 블록 리스트의 끝에 추가를 시킨다.

㈐ 동적 메모리 압축(compaction)

동적 메모리 할당 시, 현재 미사용 메모리의 총 크기는 충분하지만 새롭게 메모리를 할당할 수 없거나 할당할 미사용 메모리가 없는 경우, 또는 MC_knlGetFreeMemory() API 를 사용할 때만 일어난다. 메모리의 압축 방법은 다음과 같다.

① 메모리 사용정보를 유지하고 있는 'memblock'의 첫 블록에서부터 시작하여 해당 메모리 블록의 오른쪽 필드를 따라가며 한 블록씩 왼쪽으로 복사시킨다.

② 처리할 메모리 블록이 미사용 메모리 블록이라면 해당 블록의 start 값을 '-1'로 설정하고, 사용 중인 블록 이라면 이전에 복사시켰던 메모리 블록의 바로 다음에 복사해준다.

③ 모든 메모리 블록의 복사가 끝나면 마지막으로 미사용 메모리 블록을 하나로 모아 메모리상의 가장 마지막 부분에 추가시킨다.

## 4. 결론

현재 국내 무선인터넷 시장은 유선인터넷의 발달과 더불어 최근 5 년 동안 급속도로 성장하고 있다. 하지만 각 이동통신사들이 자신의 서비스 환경에 맞는 무선인터넷 환경과 무선인터넷을 이용한 응용프로그램 실행환경을 사용함으로써 컨텐츠 개발자는 같은 컨텐츠를 여러 개의 Target 플랫폼에 맞게 개발해야 하는 중복투자를 해야 하고, 소비자들은 이동통신사별로 컨텐츠가 한정되어 이용 가능한 서비스의 종류가 제한되는 불합리한 형태가 되었다. 이에 각 이동통신사별 무선인터넷 플랫폼 환경을 통일하는 WIPI 가 제정되었다. 그러나 최근 소비자들의 컨텐츠에 대한 요구사항이 다양해지고, 컨텐츠 개발자에게 있어서는 시장에 대한 경쟁력을 가지기 위하여 개발 비용 감소 및 개발 편의성을 제공하는 WIPI 에뮬레이터 개발이 필요하게 되었다.

본 논문에서는 본 연구팀에서 개발한 WIPI Clet 에뮬레이터 1.0 을 간단히 소개하고, 윈도우 실행환경에 의존적이었던 메모리 관리 모듈을 Win32 API 를 사용하지 않으며, 운영체제에 독립적으로 실행시킬 수 있는 WIPI Clet 에뮬레이터 2.0 의 설계 및 구현 사항을 기술하였다. WIPI Clet 에뮬레이터 2.0 은 버전 1.0 에 비하여 WIPI Clet 응용프로그램의 실행을 위한 메모리 관리에 있어 효율성을 제공하며, 적은 비용으로 다른 운영체제로의 높은 이식성을 제공한다.

향후 연구방향은 메모리의 잦은 할당 및 해제로 인하여 발생하는 메모리 관리 모듈의 과부하를 줄일 수 있는 메모리 관리 기법에 대한 연구를 계속 진행할 계획이다.

## 참고문헌

[1] 모바일 표준 플랫폼 WIPI 1.2.1, KWISFS.K-05-001R3

[2] 모바일 표준 플랫폼 WIPI 2.0, KWISFS.K-05-002

[3] "윈도우용 Clet 에뮬레이터 개발", ETRI 연구보고서, 충남대학교, 2004.8

[4] "휴대정보단말기용 웹 서비스 클라이언트 및 WIPI SDK", ETRI 연구보고서, 충남대학교, 2003.11

[5] 유용덕, 김연수, 최훈, "동적 Upgrade 기능을 구비한 WIPI 개발환경 설계", 한국통신학회 추계 종합학술발표회 논문초록집, vol. 28, p.343, 2003.12

[6] "한국무선인터넷표준화포럼" www.kwisforum.org, 2004.9.9